JAERI-M
82-199

中性子輸送コードのベクトル計算処理

(DOT3.5, TWOTRAN, ANISN, PALLAS, BERMUDA)

1982年12月

石黒美佐子・筒井　恒夫

日本原子力研究所
Japan Atomic Energy Research Institute

JAERI-Mレポートは，日本原子力研究所が不定期に公刊している研究報告書です。
入手の問合わせは，日本原子力研究所技術情報部情報資料課（〒319-11茨城県那珂郡東海村）あて，お申しこしください。なお，このほかに財団法人原子力弘済会資料センター（〒319-11 茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.
Inquiries about availability of the reports should be addressed to Information Section, Division of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

編集兼発行　日本原子力研究所
印　　刷　いばらき印刷㈱

JAERI－M 82－199

中性子輸送コードのベクトル計算処理
(DOT 3.5, TWOTRAN, ANISN, PALLAS, BERMUDA)

日本原子力研究所東海研究所計算センター
石黒美佐子・筒井 恒夫[+]

(1982年12月8日受理)

中性子輸送計算は，原子炉の遮蔽問題や臨界問題などを取扱い，原研では，計算量の多い分野の1つである。そこで，近い将来におけるスーパコンピュータの利用を想定し，中性子輸送コードのベクトル計算への適応性について調査した。

差分近似法を用いたDOT 3.5，TWOTRAN，ANISNと，直接積分法を用いたPALLAS－2DCY，BERMUDA－2DNの5つのコードを取上げる。

ベクトル計算効果は，解法，形状，取扱われる問題に大きく依存することが解った。すなわち直接積分法は差分法より，ベクトル計算向である。しかし，輸送方程式の差分化により導かれる式は，(r, z)，(r，θ)－形状では漸化式となりベクトル化できない。しかし，反復解法を変更すれば，差分式はベクトル化され，臨界問題において効果があることが分かった。

ここでは，各コードに対し，ベクトル化の問題点，プログラムの再構成，F 230－75 APUによる計算速度の実測，反復解法の数値実験などが記述される。

＋ 原子炉工学部

Vector Processing of the Neutron Transport Codes

Misako ISHIGURO and Tsuneo TSUTSUI+

Computing Center
Tokai Research Establishment, JAERI

(Received December 8, 1982)

One of the large computations in JAERI is the neutron tansport ones used for reactor shielding and criticality analyses. The adaptability of vector processings has been investigated on the neutron transport codes under the assumption of future use of super-computer.

Five codes have been tested. They are DOT3.5, TWOTRAN and ANISN based on finite difference method, and PALLAS-2DCY and BERMUDA on the direct integration method.

It has been found that the gain from vectorization depends upon the numerical methods, geometries, and problems types to be solved. That is, the direct integration is rather suited for vector processing. But in the conventional finite difference method, the difference equation has an unvectorizable recurrence form in (r,z) and (r,)-geometries. But by altering the iterative process, the equation can be vectorized and some gains have been found to be achieved in a criticality problem.

For each code, described are some views on vectorization, program restructurings, speedup ratio on F75 APU, numerical studies on the iterative process, and so forth.

Keywords; Neutron Tansport, Nuclear Codes, Vectorizations, Super Computers, TWOTRAN, DOT3.5, ANISN, PALLAS, BERMUDA

+ Division of Reactor Engineering, Tokai Research Establishment, JAERI

目　　次

CONTENTS

1. はじめに

1.1 前置

ますます巨大，精巧化する原子力コードの計算需要に対応するために，近い将来においてスーパーコンピュータの導入が必要となる。スーパーコンピュータの持つ高速演算性能は，そのベクトル計算機能にある。ベクトル計算機は汎用計算機と異なり，計算コードの数値計算アルゴリズムやプログラム構造により計算性能に大きな巾が生じる。そこで，予めよく使用される大型原子力コードについて，ベクトル計算の適応性を調査しておくことが必要となる。この目的で，昭和55年10月から富士通との間の共同研究が発足し，その一環として，この仕事がなされた。

ここでは，中性子輸送コードのベクトル計算への適応性について報告する。中性子輸送コードは，最も計算需要の多いものの1つで，原子炉の遮蔽や臨界計算に使用される。原研でよく使用される，差分法を用いたNEA CPLのDOT 3.5[1] TWOTRAN[2]とANISN[3]及び，直接積分法を用いて原研で開発されたPALLAS－2DCY[4]とBERMUDA－2DN[5]の5つのコードを調査対象とする。

調査の結果，中性子輸送コードでは，解法の差異，形状，計算目的（遮蔽計算か臨界計算）によってベクトル計算効果が異なることが解った。

1.2 差分法のベクトル化の問題点

中性子輸送方程式は，従来，差分近似により扱われてきた。ボルツマンの輸送方程式を角度依存性を考慮して解く。中性子散乱時の角度とエネルギー方向を離散化し，各エネルギー群，角度分点，空間メッシュ点について方程式を連立させて解き，実効増倍係数（K－effective）と各点の中性子束の分布を求める。方程式は，Fig. 1.1で示す2重反復解法によって解かれる。

(1) 内部反復計算（inner iteration）

同一群内の中性子束の値を空間，角度分点に対して求めるために，差分近似で離散化した輸送方程式を（これを階差式と呼ぶ）反復計算によって解く。

(2) 外部反復計算（outer iteration）

臨界計算では，階差式の右辺は，全エネルギー群の中性子束から計算される。つまり，実行増倍係数を固有値，中性子束の値を固有ベクトルとした固有値問題を解くことになる。固有値と中性子束が収束するまで反復計算される。

計算コードの実行に際しては，内部反復計算時の中性子束の計算に時間を要し，この部分の高速化が要求される。Fig. 1.1に相対計算時間を％で示す。

ところが，中性子束計算に用いられる階差式は，既知の境界値から出発して，逐次，隣の値を計算していくいわゆる漸化式で，2次元問題では，次の4点階差式で与えられる。[1],[2]

$$\phi^{g,(m)}_{i,j,k} = a + b\,\phi^{g,(m)}_{i-1,j,k} + c\,\phi^{g,(m)}_{i,j-1,k} + d\,\phi^{g(m)}_{i,j,k-1} \qquad (1.1)$$

ここで，i，j は空間メッシュ方向，k は角度方向インデックス，g はエネルギー群数，mは内側反復回数を示す。

（1.1）式で$\phi^{g,(m)}_{i,j,k}$をi，j，またはk 或いはその組合せに対しベクトル化することは，このままの形では，データの依存性から不可能である。

しかしながら，（x，y）形状では，（1.1）式の係数dがゼロとなり，k；角度インデックス，に対しベクトル化が可能となる。

（r，θ）や（r，z）形状をベクトル化するためには，（1.1）式の反復解法で，前回の反復時のϕの値を用いるようにするなど反復解法を変更しなければならない。

すなわち，i 方向にベクトル化する場合には，（1.1）式において右辺$\phi^{g,(m)}_{i-1,j,k}$の代りに$\phi^{g,(m-1)}_{i-1,j,k}$が使用される。

$$\phi^{g,(m)}_{i,j,k} = a + b\,\phi^{g,(m-1)}_{i-1,j,k} + c\,\phi^{g,(m)}_{i,j-1,k} + d\,\phi^{g,(m)}_{i,j,k-1} \qquad (1.2)$$

反復解法のこのような変更により，反復回数の増加が結果として生じるのみならず，場合によっては解が収束しない場合も起こる。また，（1.2）の反復解法の妥当性については，十分検討を要するところであるが，本報告では未検討である。反復解法のケース・スタディは，M 200 で DOT 3.5 で行なわれた。

1.3　直接積分法のベクトル化

直接積分法を用いたコードでは，中性子の散乱計算を数値積分により計算する部分に大部分の計算時間が費される。この部分は，本質的には，3 方向のメッシュ・インデックスと群インデックスに関する数値加算で，ベクトル化が容易である。ただしPALLAS コードでは，プログラム構造上 2.5 倍程度の速度向上に留まり，これを 4 倍程度に高めるには大巾な書換とそのための考察が必要である。

1.4　輸送コードのベクトル化概要

各コードのベクトル化の詳細は，以下の章で示されるが，結果をまとめると Table 1.1 に示すようになる。ここで○は適応，×は不適応，数字を記入しているのは，F75　APU（アレイプロセッサ）等による実測値を示す。数値に巾があるのは，対象とする問題に依存するからである。数値は，通常の計算（スカラー計算と呼ぶ）に比べての速度比を表わす。Table 1.1 の結果から次のことが言える。

(1)　差分法より直接積分法の方が潜在的にベクトル計算に向いている。

(2)　差分法で反復計算法を変更しない場合，つまり，（1.1）式を利用するかぎり，（x，y）

方向しかベクトル化できない。

(3) 反復解法を変更する場合においても，外部反復計算の初回は（1.1）式によるスカラー計算を行い，予め妥当な解を得ておくことが必要である。2回目から（1.2）式によるベクトル計算を行う。

(4) 固定ソース問題を取扱う遮蔽計算では，ベクトル計算しても固有値の誤差が大きく(10%)，計算精度が得られない。

Table 1.2には，ここで取上げたコードのベクトル化概要の一覧を示す。

以下では，各コードのベクトル化のための分析と再構成，F75 APUによる計算時間の実測，反復解法の数値実験，ベクトル化の問題点などが述べられる。

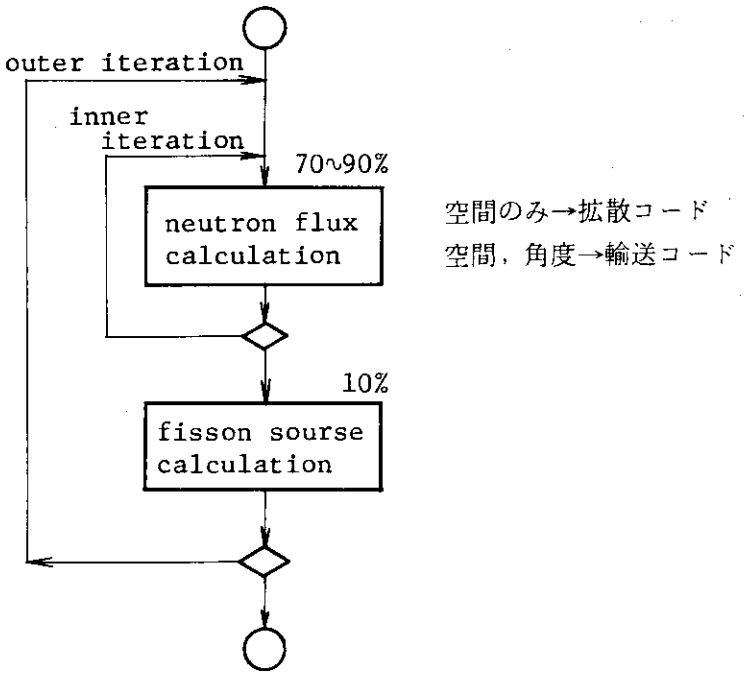


Fig. 1.1 Double loop iteration

Table 1.1 Vectorization effect of the neutron transport codes

コード名	解法	形状	反復法変更なし		反復法の変更	
			臨界	遮蔽	臨界	遮蔽
TWOTRAN	差分法	(x, y)	○ 1.9－4.5	○	○	×
TWOTRAN	差分法	(r, z) (r, θ)	×	×	○	×
DOT 3.5	差分法	(x, y)	○	○	○	×
DOT 3.5	差分法	(r, z) (r, θ)	×	×	○ 1.5－2.5	×
PALLAS	直接積分	(r, z)		○ 3.0－4.0		

Table 1.2 Summary of vectorization effect on the neutron transport codes

コード名	分野，問題	解法	ソース枚数	ベクトル化率	速度向上	ソース修正率	変換のレベル	ベクトル長	CPU M200	I/O回数	評価手段	年
TWOTRAN	2次元多群，(x, y), S_{16}	差分法	7,000	83 %	4.5倍	4 %	3	48	3.6 M (F75)		F75 APU	52
TWOTRAN	〃 (r, z), S_{18}			20 %	1.1倍		3	−	8.2		F75 APU	52
TWOTRAN	〃 (x, y), S_8			75 %	2.0倍		3	10	2.7	500 ~1000	F75 APU	57
ANISN	1次元多群	差分法	3,000	21 %	1.2倍	2.3 %	2	−	23.3		$\alpha = 5$ で推定	56
DOT 3.5 (遮蔽)	2次元多群，(x, y)	差分法	6,500	80 %	1.4倍	8 %	3	96	8.5	5671	M 200の数値実験からの推定	57
DOT 3.5 (遮蔽)	〃 (r, θ), or (r, z)			20 %	1.1倍	〃	3	−	−			57
DOT 3.5 (臨界)	〃 (x, y)			78 %	2.0倍	〃	3	16	−			57
DOT 3.5* (臨界)	〃 (r, θ), or (r, z)			78 %	2.0倍	〃	3	40	0.4	500		57
BERMUDA-2DN	〃 (r, z)	直接積分	5,000	90 %	3.0倍	2 %	2	40	19.3	500 ~1000	F75 APU	57
PALLAS-2DCY	〃 (r, z) 群メッシュ	直接積分	3,800	80 %	2.6倍	8 %	2	60	5.6	1000~3000	F75 APU	57

α ; ベクトル／スカラ計算速度比

変換のレベル { 3 数値計算モデルの変更
2 プログラム構造の変更

* 反復解法が変更による

2. DOT3.5[1]（2次元中性子輸送コード）

2.1 計算内容

2次元中性子輸送コードで，オークリッジ国立研究所で作成された。(x, y), (r, z), (r, θ) 形状が取扱える。原研では，遮蔽計算によく使用されている。コードはNEA CPL のバージョンに基づく。

2.2 計算コードとテストデータの概要

ソース行数	約6,500	
主 記 憶	1,432KB	（遮蔽計算の場合）
CPU時間（M200）	8分32秒	（ 同 上 ）
入出力回数	5671	（ 同 上 ）
使用ファイル数	11ファイル，いずれも作業用	

計算問題

(1) 遮蔽計算（固定ソース問題）

15群，(x, y) 形状20×40メッシュ，角度分点96

原工部遮蔽研究室のデータ

(2) 臨界計算（外部反復計算あり）

3群，(x, y) 形状，40×3メッシュ，角度分点16，NEA CPL のサンプル問題，M 200によるCPU時間25秒

2.3 計算時間のかかるサブルーチン

遮蔽計算におけるM 200での各サブルーチンの実行時間をTable 2.1で示す。時間を要するのはサブルーチンGRINDで，この場合93%が費される。GRINDは，内部反復計算（Fig. 1.1参照）の核となるサブルーチンで，そのDOループの構造はFig. 2.1で示される。GRIND はINNERから群(g)毎，内部反復(m)毎に呼ばれる。

上下，或いは，左右のいずれかの境界値の値が与えられ，いずれかの方向に逐次的に計算が進められる。この時，境界値は，前回の反復時の計算結果に基づいて決定される。

Table 2.1 Summary of computing time in DOT3.5 code

Subroutines	CPU time %
GRIND	93.04
INNER	4.11
WWESOL	1.64
OUTER	1.05
WANDER	0.16
	——

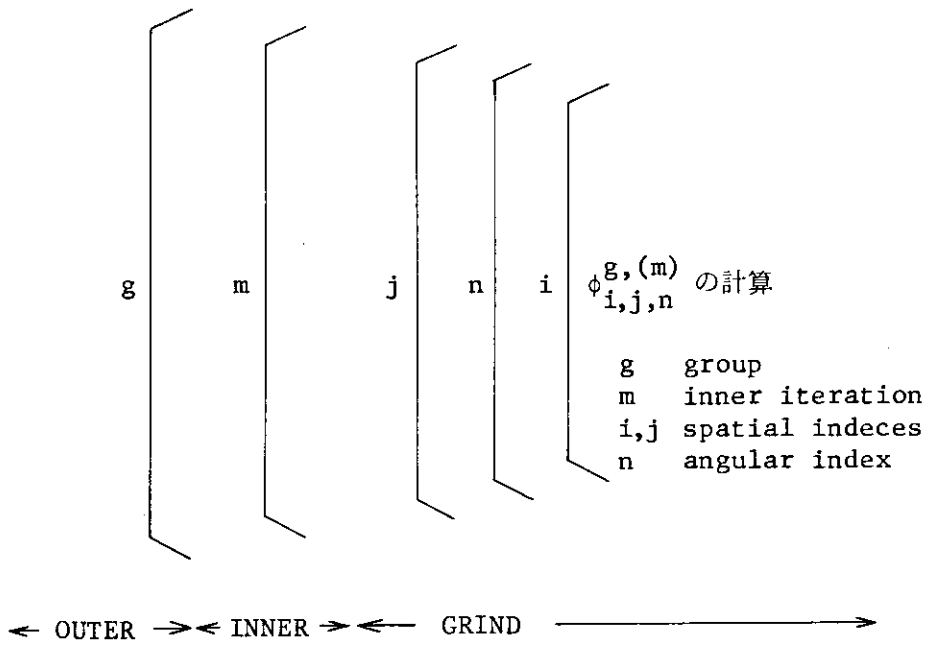


Fig. 2.1 DO loop structure of subroutine GRIND in DOT3.5 code

2.4 数値計算法の概要[1]

Fig. 2.2にシリンダ形状の座標型を示す。解くべき輸送方程式は，r，z，θ，η，ψ，Eに対し次のように与えられる。

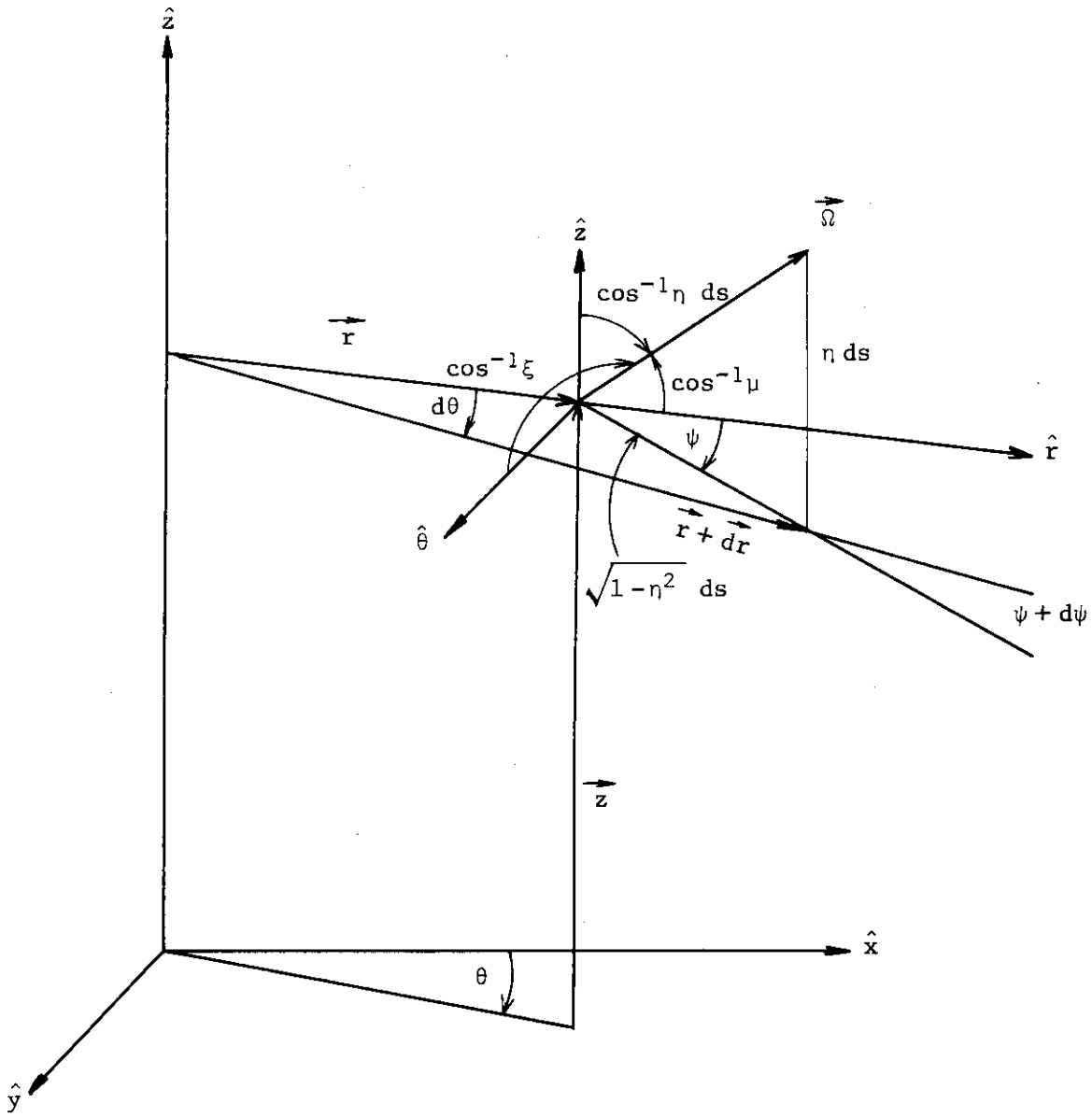


Fig. 2.2 The coordinate system for cylindrical geometry

$$\frac{d\phi(r,z,\theta,\eta,\psi,E)}{ds}+\Sigma^{T}(r,z,\theta,E)\phi(r,z,\theta,\eta,\psi,E)$$

$$=S(r,z,\theta,\eta,\psi,E)$$

$$+\int_{-1}\int_{0}^{2\pi}\int_{0}^{\infty}\Sigma^{s}(r,z,\theta:E'\overline{\Omega}'\rightarrow E,\overline{\Omega})\phi(r,z,\theta,\eta',\psi',E')dE'd\psi'd\eta' \tag{2.1}$$

ここでϕは中性子束，sは空間及び角度方向を表わしr, z, θ, ηで与えられる。Sは中性子源を表わす。

r, z, η, ψ, E に対するインデックスを(i, j, K, n, g)とおけば，差分近似により，以下の式が導出される。

$$\phi_{G,I,J,D} = \left(|\bar{\mu}_D| \, 2\pi \bar{r}_I \, \Delta z_J \left(\text{or} \begin{matrix} \phi_{G,i+1,J,D} \\ \phi_{G,i,J,D} \end{matrix} \right) + |\bar{\eta}_D| \, 2\pi \bar{r}_I \, \Delta \bar{r}_I \left(\begin{matrix} \phi_{G,I,j+1,D} \\ \phi_{G,I,J,D} \end{matrix} \right) \right.$$

$$\left. + \frac{\Delta z_J}{W_D} \bar{r}_N \, \phi_{G,I,J,n,K} + \frac{1}{2} V_{I,J} S'_{G,I,J,D} \right) \Big/ \left(|\bar{\mu}_D| \, \Delta z_J \, 2\pi \bar{r}_I + |\bar{\eta}_D| \, 2\pi \bar{r}_I \, \Delta \bar{r}_I \right.$$

$$\left. + \frac{\Delta z_J}{W_D} \bar{r}_N + \frac{1}{2} V_{I,J} \Sigma_G^T \right), \quad \bar{r}_N = \frac{1}{2} (r_{n+1} + r_n) \qquad (2.2)$$

ここで，I, J, Dは，各々$(i-\frac{1}{2}, i+\frac{1}{2})$，$(j-\frac{1}{2}, j+\frac{1}{2})$及び$(n-\frac{1}{2}, n+\frac{1}{2}; K)$の区間を示す。また，どちらの端から計算されるかによって上段または下段の計算式が使用される。

(2.2)式から$\phi_{G,I,J,D}$を求め，以下の(2.3)式からメッシュ(i, j, n, g)上のϕを補間する。

$$\begin{bmatrix} \phi_{G,i,J,D} \\ \phi_{G,i+1,J,D} \end{bmatrix} = 2\,\phi_{G,I,J,D} - \begin{bmatrix} \phi_{G,i+1,J,D} \\ \phi_{G,i,J,D} \end{bmatrix}$$

$$\begin{bmatrix} \phi_{G,I,j,D} \\ \phi_{G,I,j+1,D} \end{bmatrix} = 2\,\phi_{G,I,J,D} - \begin{bmatrix} \phi_{G,I,j+1,D} \\ \phi_{G,I,j,D} \end{bmatrix}$$

$$\phi_{G,I,J,n+1,K} = 2\,\phi_{G,I,J,D} - \phi_{G,I,J,n,K} \qquad (2.3)$$

(2.2)－(2.3)式の計算は，"power iteration"法により実行される。(2.2)－(2.3)式は，例えば(r, z)形状の場合にはr方向インデックス(i), z方向インデックス(j)および角度方向インデックス(n)に対する漸化式を成す。しかしながら，(x, y)形状では，(2.2)式の右辺第1項の係数が0となり，nに対して依存性がなくなる。このことは，同じ解法に基づくTWOTRANコードでも言える。

従って，(x, y)形状以外では，ベクトル計算する場合に，前回で計算された中性子束の値で代行するなど反復解法の変更が必要となる。

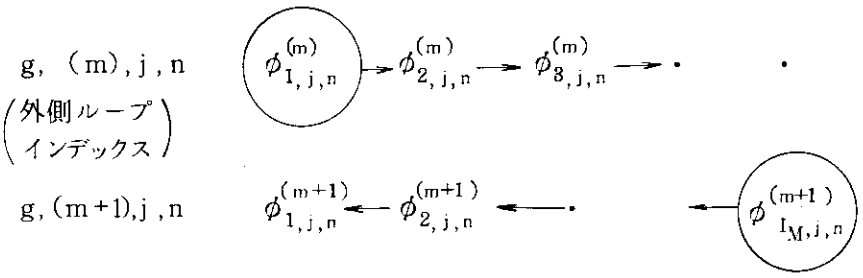
2.5 反復解法の変更と数値実験

m　　内部反復回数

f → g　　gはfを用いて計算される

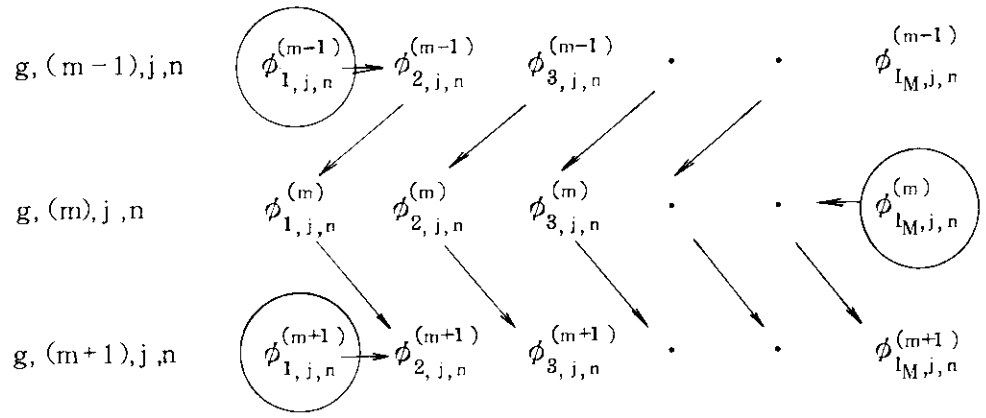
とおく。

(1) スカラ計算



の値が与えられ，右向き左向き交互に計算される。　　（2.4）

(2)　ベクトル計算（$\phi^{(m)}$ の代りに $\phi^{(m-1)}$ を利用）



（2.5）

（2.5）式を用いたベクトル計算化は，数値実験の結果から反復計算の初期の段階では誤差が大きく，大まかな解が得られるまで従来のスカラー計算（2.4）式により逐次的に計算することが必要であることが解った。臨界問題と遮蔽問題について反復解法についてM 200で数値実験を行った結果をTable 2.2に示す。

これらの結果から，次のことが解った。

(1)　2重反復計算（Fig. 1.1）の外部反復計算の初回をスカラー計算し，2回目からベクトル計算を行う方法では，反復回数は増加するが解が得られる。

(2)　内部反復計算の初めの数回をスカラー計算し，残りをベクトル計算する場合には，固有値の誤差が大きくなり妥当な解が得られない。また，たとえ求解できたとしても，内部反復回数は3～10程度の小さい値のため，反復計算の後半の部分をベクトル計算しても，反復回数の増加がベクトル化による速度アップを相殺し，コード全体としての速度向上は望めない。

2.6　ベクトル化効果

（2.5）式で示したベクトル化手法は，全形状に適応できる。臨界計算では，外部反復計算の2回目以降についてベクトル計算できる。この場合，反復回数の増加で1.5倍程度の計算量となり，この部分のベクトル計算により4倍の速度向上となる。初回の反復時のスカラー計算（約10％），コードでベクトル化できない部分（10％）を考慮すると，コード全体の計算速度は，次の

計算により，約2倍向上する。

反復計算（ベクトル化）	0.78（計算コスト比）×0.25（計算時間比）×1.5（反復回数増加比）
反復計算（初回スカラー）	0.08
その他の部分（スカラー）	0.14
合　計	0.51*

（*スカラー計算を1とした場合の相対計算時間）

反復計算についてのケース・スタディの結果はTable 2.2に示される。この結果によれば，2重反復計算（Fig.1.1）のうち，固有値等を求めるためのループである外側反復計算の初回（内部反復計算を含め）をスカラー計算し，それ以降をベクトル計算に引継ぐ方が，毎回の外部反復計算の初回の内部反復計算にスカラー計算を用いるより収束性において良好な結果を得ている。

したがって，遮蔽計算のような固定ソース問題では，外部反復計算にあたるものがないので，収束性も悪く，収束して得た固有値は，スカラー計算に較べ約10%の誤差を持つ。

ここで扱った問題では，臨界計算については，メッシュ数が少ないので，実用規模の問題でのケース・スタディが必要である。

Table 2.2　Case study for iterative method of the DOT 3.5 code on M200

問　題	反復計算法（注1）	INNER 最大値	OUTER 回　数	M200 CPU	予想倍率（注2） GRINDの部分
臨界計算 NEAサンプル 3群 x y　40×3 角度　16 $\varepsilon = 10^{-3}$	全部シリアル（オリジナル）	9	12	15.24S	1.0
	INNER 初回シリアル 2回目以降パラレル（ベクトル化）	9	15	30.61S	1.4
	INNER 1～3回シリアル 4回目以降パラレル（ベクトル化）	9	13	25.61	1.0
	OUTER 初回シリアル 2回目以降パラレル（ベクトル化）	9	14	22.07	2.3
遮蔽計算 15群 x y　20×46 角度 96 $\varepsilon = 2\times10^{-3}$	全部シリアル（オリジナル）	10 実際3～4	1	5M45S	1.0
	INNER 1～3回シリアル（注3） 4回目以降パラレル	10 実際7	1	9M39S	0.96

注1　"diamond－difference method" による漸化式の計算

注2　部分的にベクトル化することによる速度倍率は，計算時間比の逆数として得た。

ベクトル計算による計算時間（推定）
（一部シリアル）

＝（パラレル＋シリアル時間）（パラレル計算による反復数×1/4＋シリアル計算による反復数）／（全反復数）

注3　固有値の誤差が10%を大きく，計算値は不正確

2.7 プログラムの再構成

反復解法の変更のためにGRIND, INNER, OUTER, CNNP, FISCAL, 5ルーチン, 1322行のうち, GRINDの再作成約500行, その他の修正50行が行なわれた。

再構成では, x方向(i)40メッシュをベクトル化した。Fig. 2.3に, 中性子束$\phi^{(m)}_{i,j,n}$の計算(NBARの計算)のベクトル化について示す。NBARの計算で, B2MJとあるのが(2.4)式の$\phi^{m}_{i-1,j,n}$にあたる。これを(2.5)式のように反復解法を変更し, i = 1, 2,……, imax (40)についてベクトル計算できるようにする。この場合のB2MJ(I)には, $\phi^{(m-1)}_{i-1,j,n}$が入っている。

なお, GRINDルーチンから見たNBARの計算の占める割合は70%程度で, この他のベクトル計算できる部分を含めベクトル化できるのは約90%である。

```
C
C * * * * * * * * * *     INITIAL DIRECTION FLUX CALCULATION
C                         OR X-Y GEOMETRY FLUX CALCULATION
C * * * * * * * * * *     LINEAR MODEL
 8004 NBAR=(SOR+(XMUA1+XMUA2)*B2MJ+ETAB2*B4MI) /
     1      (SIGT + XMUA1 + XMUA2 + ETAB2)
```

⇩

```
8004 NNBAR(I)=(SOR(I)+(XMUA1(I)+XMUA2(I))*B2MJ(I)+ETAB2(I)*B4MI(I)) /
    1      (SIGT(I) + XMUA1(I) + XMUA2(I) + ETAB2(I))
```

Fig. 2.3 Restructuring for vector processings of neutron flux calculation in DOT3.5 Code

2.8 反復解法に対する考察

輸送方程式の差分近似では, $\phi^{(m)}_{i,j,n}$の計算式は, (2.2)および(2.3)で与えられ, それらを単純化して表現すれば, 既に(1.1)式で与えた階差式となる。

$$\phi^{(m)}_{i,j,n} = a + b\phi^{(m)}_{i-1,j,n} + c\phi^{(m)}_{i,j-1,n} + d\phi^{(m)}_{i,j,n-1}$$

一方, 3次元拡散方程式における階差式は, 次のように与えられる。

$$\phi^{(m)}_{i,j,n} = a + b\phi^{(m)}_{i-1,j,n} + c\phi^{(m)}_{i,j-1,n} + d\phi^{(m)}_{i,j,n-1}$$

$$+ e\phi^{(m-1)}_{i+1,j,n} + f\phi^{(m-1)}_{i,j+1,n} + g\phi^{(m-1)}_{i,j,n+1} \qquad (2.6)$$

拡散方程式では, 第m回目の反復時のϕの値は, 下方インデックスについては第m回目のもの

を，上方インデックスについては第（m－1）回目のものを用いて計算される。ところが，輸送方程式では，漸化式だから，下方インデックスのϕの値しか使われないという点でデータの再帰的参照の度合が大きい。

TWOTRANコードの場合も含め，中性子束計算には，拡散コードで見られるようなSORなどの過大緩和法は用いられない。TWOTRAN，DOT 3.5コード共に，メッシュの計算順序により解が偏向しないように計算順序を綿密に配慮した逐次計算が基本となっている。したがって，（2.5）式で与えたベクトル化手法は，解の収束性について大いに議論の余地がある。

差分近似法を使用する場合，拡散コードでは，ベクトル計算により生じる収束劣化問題は，緩和因子の変更によりある程度回避でき，また高々数10%程度の反復回数の増加があっても求解できる。[6] しかし，中性子輸送問題は，解法から来る逐次性を回避するのは，今のところなかなか難しい。

3. TWOTRAN[2]（2次元中性子輸送コード）

3.1 計算内容

2次元中性子輸送コードで，ロスアラモス研究所で作成された。（x, y），（r, z），（r, θ）形状が取扱える。原研では，遮蔽計算，臨界計算によく使用されている。コードはNEA CPLのバージョンに基づく。

3.2 計算コードとテストデータの概要

ソース行数	約7,000
主 記 憶	1～1.5Mバイト
CPU時間（M200）	1分～10分
入出力回数	500～1000
使用ファイル数	12
計 算 問 題	
Table 3.1を参照	

3.3 計算時間のかかるサブルーチン

内部反復計算（Fig. 1.1参照）のために，全計算時間の75～95％（問題による）が費される。内部反復計算には，疎メッシュ・リバランシング法が使用されている。

各サブルーチンの呼出回数，CPU時間はTable 3.2[7]で示される。

角度メッシュに対し，外側境界値を設定して内側に向って計算するINサブルーチンと，内側の境界値を決め外側に向って計算するOUTサブルーチンが交互に呼ばれる。最内ループを成す計算式は，漸化式である。

Table 3.1 Vectorization effect of TWOTRAN Code on F75 APU

比較項目＼問題		xy ケース1	xy ケース2	xy ケース3	rz
空間メッシュ		20 × 20	20 × 20	20 × 20	68 × 26
角度分点（メッシュ）		S_{12}（21）	S_{16}（36）	S_8（10）	S_{18}（10）
エネルギー群		5	5	9	8
ベクトル長		21	36	10	68
演算結果	FORTH（F75）	134.7	210.5	444.4	1485.6
	AP-FORT	32.6	39.1	198.3	1309.5
	CPU	7.8	8.1	40.9	86.4
	合　計	40.4	47.2	239.2	1395.9
	倍　率	3.33	4.46	1.86	1.06

(1) （xy）ケース1，ケース2および（r, z）については富士通による。

(2) S_N 計算の場合の角度分点数は N（N＋2）／8 で，（x, y）形状の場合には，角度分点方向をベクトル計算する。

(3) （r, z），（r, θ）では速度向上は期待できない。

Table 3.2 Summary of computing time in TWOTRAN Code

サブルーチン名	r z		x y	
	CPU 時間比率	CALL される回数	CPU 時間比率	CALL される回数
IN	37.1％	13,520	48.2％	1,840
OUT	36.8	13,520	44.9	1,840
FIXUP	21.4	1,785,197	0.6	2,848
その他	4.7	－	6.3	－

（x, y）形状については，Table 3.2 の S_{12} と S_{16} の2ケース分を対して計測した。

3.4 数値計算法の概要[2]

ボルツマンの輸送方程式は次のように与えられる。

$$\nabla(\Omega\varphi)+\sigma_t\varphi(r,E,\Omega)=\iiint dE'd\Omega\varphi(r,E',\Omega')\,\sigma_s(E'\to E,\Omega'\cdot\Omega)$$

$$+\chi(E)\iiint dE'\,d\Omega'\varphi(r,E',\Omega')\,\nu\sigma_f(E')/4\pi+Q(r,E,\Omega) \qquad (3.1)$$

ここで，φ は中性子束，r は空間座標（2次元），Ω は角度方向，E はエネルギー群，σ_s，σ_t，$\nu\sigma_f$，χ は群定数，Q は中性子源を表わす。

上式は，差分法による離散化の結果，空間メッシュをｉ，ｊ，角度をメッシュｍ，群をｇとおくと，メッシュインデックス（ｉ，ｊ，ｍ，ｇ）上の中性子束$N_{i,j,m,g}$は次のように表わされる。

サブルーチンＩＮの場合

$$N=\frac{|\mu|(A_{i+1/2}+A_{i-1/2})N_{i+1/2}+2|\eta|BN_{j+1/2}+(A_{i+1/2}-A_{i-1/2})(\alpha_{m+1/2}+\alpha_{m-1/2})N_{m-1/2}W+SV}{|\mu|(A_{i+1/2}+A_{i-1/2})+2|\eta|B+(A_{i+1/2}-A_{i-1/2})(\alpha_{m+1/2}+\alpha_{m-1/2})W+\sigma_t V}$$

$$i=imax, imax-1, \cdots, 1,\ j=jmax, jmax-1, \cdots, 1,\ m=1, 2, \cdots, mmax \qquad (3.2)$$

ここでＮ以外の記号は定数として前もって計算される。

ここで解かるように，インデックス（ｉ，ｊ，ｍ，ｇ）上の中性子束の計算は，直前に計算された中性子束の値$N_{i+1/2}$，$N_{j+1/2}$，$N_{m-1/2}$を使用して漸次実行され，ベクトル計算する場合には，このデータの依存性が問題となる。

上式のデータ間依存は，（ｘ，ｙ）形状の場合には（$A_{i+1/2}-A_{i-1/2}$）＝０となり，ｍ（角度分点）の方向に依存性がなくなり，角度分点方向にベクトル計算が可能となる。但し角度分点数はS_n計算の場合ｎ（ｎ＋２）／８個となる。通常S_4，S_8，S_{12}，S_{16}で計算され，ベクトル長は３～40と短かい。

S_8の場合の角度メッシュ順序をFig. 3.1で示す。

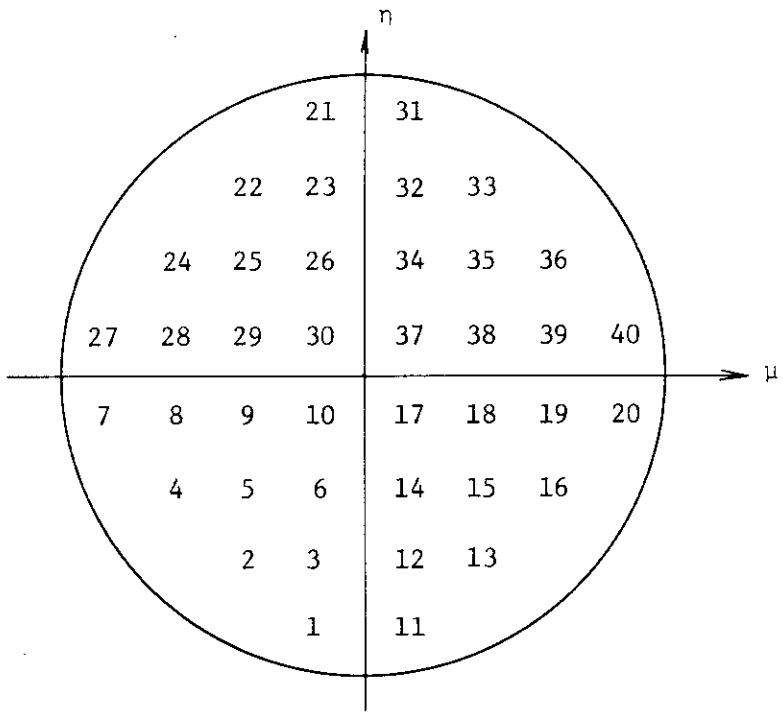


Fig. 3.1 Ordering of S_8 directions

3.5 プログラムの再構成とベクトル化率

52～53年に，富士通の南氏らがＦ75 APU用に変換したもの[7]に対し，COMMON文での配列長を可変にするなど変更した。

富士通で作成されたものは，（x, y）形状とそれ以外の形状とでプログラムが別になっていた。Fig. 3.2にサブルーチンINのオリジナル版，Fig. 3.3にINの（x, y）形状ベクトル版,Fig.3.4に同じく（r, z）または（r, θ）形状のベクトル版のプログラムを示す。

（x, y）形状の場合には，（3.2）式の計算を角度分点方向（M）にベクトル化できるので，係数を前もって配列化しておき，M＝1, 2,……MMDに対し同時に計算させる。

（r, z）または（r, θ）形状では，（3.2）式はベクトル計算できないので係数計算のみベクトル化する。

ベクトル化率は，（x, y）形状で70～90%（問題による）。その他の形状では約20%である。

3.6 F75 APUによる計算速度

南氏らによりテストされた問題に，新たに，S_nのnの数の小さい場合の（x, y）形状を追加実測した。それらを合わせてTable 3.2に示す。

3.7 反復解法の変更について

（x, y）形状のみしかベクトル化効果が上らないのは，DOT 3.5の場合と同じである。その他の形状については，DOT 3.5で用いた（2.5）式による反復計算法を採用すれば，外部反復計算を多く行うケースではある程度のベクトル化効果が期待できる。

```
ISN  ST-NO                SOURCE PROGRAM                                          SEQUENCE
 1          SUBROUTINE IN ( S,F,BH,BV,AL,P,W,U,E,WU,WE,AL1,AL2,A1,A2,A3,A4,CX,00500100
           1FH,FV,ID,NMD,ITD,JTD,MMD,NND,AF,ITPD )                               00500200
      C                                                                          00500300
 2          COMMON IA(250)                                                      00500400
      C                                                                          00500500
 3          COMMON /SWEEP/ BA,BC,J,J1,J2                                        00500600
 4          DIMENSION S(NMD,ITD),F(NMD,ITD),BH(JTD,MMD),BV(ITD,MMD),            00500700
           1AL(NND,ITD),P(NMD,MMD),AF(ITPD,MMD)                                 00500800
 5          DIMENSION W(1),U(1),E(1),WU(1),WE(1),AL1(1),AL2(1),A1(1),A2(1),     00500900
           1A3(1),A4(1),CX(1),FH(1),FV(1),ID(1)                                 00501000
      C                                                                          00501100
 6          EQUIVALENCE (IA(57),MM),(IA(58),NM),(IA(64),IT),(IA(77),ITP),       00501200
           1(IA(175),JCONV),(IA(201),ZZ),(IA(202),BB),(IA(203),CC),            00501300
           2(IA(204),DD),(IA(205),T),(IA(206),QT),(IA(207),CT),(IA(210),NN),   00501400
           3(IA(211),AA),(IA(212),TI),(IA(213),TJ),(IA(214),TM)                 00501500
      C                                                                          00501600
 7          DO 210 IB=1,IT                                                      00501700
 8          I=ITP-IB                                                            00501800
 9          I1=ID(I)                                                            00501900
10          I2=ID(I-1)                                                          00502000
11          CT=CX(I)                                                            00502100
12          M=1                                                                 00502200
13          DO 170 K=1,NN                                                       00502300
14          DO 160 L=1,K                                                        00502400
15          QT=0.0                                                              00502500
16          DO 100 N=1,NM                                                       00502600
17      100 QT=QT+P(N,M)*S(N,I)                                                 00502700
18          AA=U(M)*A1(I)                                                       00502800
19          BB=BA*E(M)*A3(I)                                                    00502900
20          IF (L.EQ.1) GO TO 110                                               00503000
21          CC=A2(I)*(AL1(M)+AL2(M))                                            00503100
22          T=(AA*BH(J,M)+BB*BV(I,M)+CC*AL(K,I)+QT)/(AA+BB+CC+CT)               00503200
23          TM=T+T-AL(K,I)                                                      00503300
24          GO TO 120                                                           00503400
25      110 T=(AA*BH(J,M)+BB*BV(I,M)+QT)/(AA+BB+CT)                             00503500
26          TM=T                                                                00503600
27      120 TI=T+T-BH(J,M)                                                      00503700
28          TJ=T+T-BV(I,M)                                                      00503800
29          IF (TI.LT.0.0) GO TO 130                                            00503900
30          IF (TJ.LT.0.0) GO TO 130                                            00504000
31          IF (TM.LT.0.0) GO TO 130                                            00504100
32          BH(J,M)=TI                                                          00504200
33          BV(I,M)=TJ                                                          00504300
34          AL(K,I)=TM                                                          00504400
35          GO TO 140                                                           00504500
36      130 ZZ=U(M)*A4(I+1)                                                     00504600
37          AA=AA-ZZ                                                            00504700
38          BB=0.5*BB                                                           00504800
39          CC=A2(I)*AL1(M)                                                     00504900
40          DD=A2(I)*AL2(M)                                                     00505000
41          CALL FIXUP (BH(J,M),BV(I,M),AL(K,I))                                00505100
42      140 T=W(M)*T                                                            00505200
43          DO 150 N=1,NM                                                       00505300
44      150 F(N,I)=F(N,I)+P(N,M)*T                                              00505400
45      160 M=M+1                                                               00505500
46      170 CONTINUE                                                            00505600
47          IF (I2.EQ.I1) GO TO 190                                             00505700
```

Fig. 3.2-1

```
FACOM 230-75  M7   AP-FORTRAN COMPILER   (OPT2)   SOURCE PROGRAM LIST  -781005-(V01,L11)  DATE  82.05.31/17:28    PAGE  70

ISN   ST-NO                SOURCE PROGRAM      ( IN      )                    SEQUENCE

      C                                                                       00505800
      C     COMPUTE HORIZONTAL FLOW                                           00505900
      C                                                                       00506000
 48         TA=BC*A4(I)                                                      00506100
 49         DO 180 M=1,MM                                                    00506200
 50     180 FH(I1)=FH(I1)+WU(M)*BH(J,M)*TA                                   00506300
 51     190 CONTINUE                                                         00506400
 52         IF (JCONV.NE.2) GO TO 210                                        00506500
      C                                                                       00506600
      C     SAVE HORIZONTAL ANGULAR FLUX                                      00506700
      C                                                                       00506800
 53         DO 200 M=1,MM                                                    00506900
 54     200 AF(I,M)=BH(J,M)                                                  00507000
 55     210 CONTINUE                                                         00507100
 56         IF (J2.EQ.J1) RETURN                                             00507200
      C                                                                       00507300
      C     COMPUTE VERTICAL FLOW                                             00507400
      C                                                                       00507500
 57         DO 220 I=1,IT                                                    00507600
 58         I1=ID(I)                                                         00507700
 59         DO 220 M=1,MM                                                    00507800
 60     220 FV(I1)=FV(I1)+WE(M)*BV(I,M)*A3(I)                                00507900
 61         RETURN                                                           00508000
 62         END                                                              00508100
```

Fig. 3.2-2 Subroutine IN in original TWOTRAN code

```
FACOM 230-75  M7   AP-FORTRAN COMPILER   (OPT2)   SOURCE PROGRAM LIST  -781005-(V01,L11)  DATE  82.06.01/14:33   PAGE  5

ISN  ST-NO                SOURCE PROGRAM                                          SEQUENCE

 1          SUBROUTINE IN ( S,F,BH,BV,AL,P,W,U,E,WU,WE,AL1,AL2,A1,A2,A3,A4,CX,00001970
           1FH,FV,ID,NMD,ITD,JTD,MMD,NND,AF,ITPD,                                00001980
           2QQT,AAA,BBB,ABC,TTT,TTI,TTJ,IDD,WRK1,MMM)                            00001990
     C                                                                           00002000
 2          COMMON IA(250)                                                      00002010
     C                                                                           00002020
 3          COMMON /SWEEP/ BA,BC,J,J1,J2                                        00002030
 4          DIMENSION S(NMD,ITD),F(NMD,ITD),BH(JTD,MMD),BV(ITD,MMD),            00002040
           1AL(NND,ITD),P(NMD,MMD),AF(ITPD,MMD)                                 00002050
 5          DIMENSION W(MMD),U(MMD),E(MMD),WU(MMD),WE(MMD),                     00002060
           * AL1(1),AL2(1),A1(1),A2(1),                                         00002070
           1A3(1),A4(1),CX(1),FH(1),FV(1),ID(1)                                 00002080
     C                                                                           00002090
 6          EQUIVALENCE (IA(57),MM),(IA(58),NM),(IA(64),IT),(IA(77),ITP),       00002100
           1(IA(175),JCONV),(IA(201),ZZ),(IA(202),BB),(IA(203),CC),            00002110
           2(IA(204),DD),(IA(205),T),(IA(206),QT),(IA(207),CT),(IA(210),NN),   00002120
           3(IA(211),AA),(IA(212),TI),(IA(213),TJ),(IA(214),TM)                00002130
 7          DIMENSION  QQT(MMD),AAA(MMD),BBB(MMD),ABC(MMD),                    00002140
           1          TTT(MMD), TTI(MMD),TTJ(MMD),IDD(MMD),WRK1(NMD)           00002150
 8          INDEX IDX/1,NEG/                                                    00002160
 9          LOGICAL MMM(MMD)                                                    00002170
     C                                                                           00002180
10          DO 210 IB=1,IT                                                      00002190
11          I=ITP-IB                                                            00002200
12          I1=ID(I)                                                            00002210
13          I2=ID(I-1)                                                          00002220
14          CT=CX(I)                                                            00002230
15          QQT(*)=PMM(S(*,I),P(*,*))                                           00002240
16          AAA(*)=A1(I)*U(*)                                                   00002250
17          BBB(*)=(BA*A3(I))*E(*)                                              00002260
18          ABC(*)=1.0/(AAA(*)+BBB(*)+CT)                                       00002270
19          TTT(*)=(AAA(*)*BH(J,*)+BBB(*)*BV(I,*)+QQT(*))*ABC(*)                00002280
     C                                                                           00002290
20          TTI(*)=TTT(*)+TTT(*)                                                00002300
21          TTJ(*)=TTI(*)-BV(I,*)                                               00002310
22          TTI(*)=TTI(*)-BH(J,*)                                               00002320
23          MMM(*)=TTI(*).LT.0.0 .OR. TTJ(*).LT.0.0                             00002330
24          NEG=IONC(MMM(*))                                                    00002340
25          IF(NEG.EQ.0) GO TO 140                                              00002350
26          IDD(IDX)=IDXL(MMM(*))                                               00002360
27          DO 130 NNN=1,NEG                                                    00002370
28          M=IDD(NNN)                                                          00002380
29          ZZ=U(M)*A4(I+1)                                                     00002390
30          AA=AAA(M)-ZZ                                                        00002400
31          BB=0.5*BBB(M)                                                       00002410
32          CALL FIXUP(BH(J,M),BV(I,M),TTT(M),TTI(M),TTJ(M),QQT(M),ZZ,AA,BB,    00002420
           1     CT)                                                            00002430
33      130 CONTINUE                                                            00002440
34      140 CONTINUE                                                            00002450
35          TTT(*)=W(*)*TTT(*)                                                  00002460
36          BH(J,*)=TTI(*)                                                      00002470
37          BV(I,*)=TTJ(*)                                                      00002480
38          DO 150 N=1,NM                                                       00002490
39      150 WRK1(N)=AIPD(P(N,*),TTT(*))                                         00002500
40          F(*,I)=F(*,I)+WRK1(*)                                               00002510
41      170 CONTINUE                                                            00002520
42          IF (I2.EQ.I1) GO TO 190                                             00002530
```

Fig. 3.3-1

```
FACOM 230-75  M7   AP-FORTRAN COMPILER   (OPT2)   SOURCE PROGRAM LIST  -781005-(V01,L11)  DATE  82,06,02/14:33    PAGE   6

ISN  ST-NO                SOURCE PROGRAM      ( IN      )                    SEQUENCE

     C                                                                      00002540
     C      COMPUTE HORIZONTAL FLOW                                         00002550
     C                                                                      00002560
 43          TA=BC*A4(I)                                                    00002570
 44          FH(I1)=FH(I1) + TA*AIPD(WU(*),BH(J,*))                         00002580
 45      190 CONTINUE                                                       00002590
 46          IF (JCONV.NE.2) GO TO 210                                      00002600
     C                                                                      00002610
     C      SAVE HORIZONTAL ANGULAR FLUX                                    00002620
     C                                                                      00002630
 47          AF(I,*)=BH(J,*)                                                00002640
 48      210 CONTINUE                                                       00002650
 49          IF (J2.EQ.J1) RETURN                                           00002660
     C                                                                      00002670
     C      COMPUTE VERTICAL FLOW                                           00002680
     C                                                                      00002690
 50          DO 1220 I=1,IT                                                 00002700
 51          I1=ID(I)                                                       00002710
 52     1220 FV(I1)=FV(I1)+A3(I)*AIPD(WE(*),BV(I,*))                        00002720
 53          RETURN                                                         00002730
 54          END                                                            00002740
```

Fig. 3.3-2 Subroutine IN of (x,y)-geometry in vectorized TWOTRAN code

```
      SUBROUTINE  INOUT(KK,KM,XL,II,MX)                                 00000060
      COMMON IA(250)                                                    00000070
      EQUIVALENCE   (IA(64),IT),(IA(77),ITP),(IA(210),NN)               00000080
      DIMENSION   KK(MX),KM(MX),XL(MX),II(IT)                           00000090
C****     APU013    **   INITIAL VALUES SETTING INTEAD OF DO-VARIABLES  00000100
      M=1                                                               00000110
      DO 10 K=1,NN                                                      00000120
      DO 10 L=1,K                                                       00000130
      KK(M)=K                                                           00000140
      KM(M)=K*(K+1)/2+1-L                                               00000150
      XL(M)=1.0                                                         00000160
      IF(L.EQ.1)XL(M)=0.0                                               00000170
   10 M=M+1                                                             00000180
      DO 20 I=1,IT                                                      00000190
   20 II(I)=ITP-I                                                       00000200
      WRITE(6,1)                                                        00000210
    1 FORMAT(//'   ¥¥¥ DATA CHECK IN INOUT ¥¥¥',//)                     00000220
      DO 30 M=1,MX                                                      00000230
   30 WRITE(6,2) M,KK(M),KM(M),XL(M)                                    00000240
    2 FORMAT(3I10,F10.1)                                                00000250
      DO 40 I=1,IT                                                      00000260
   40 WRITE(6,3) I,II(I)                                                00000270
    3 FORMAT(2I10)                                                      00000280
      RETURN                                                            00000290
      END                                                               00000300
      SUBROUTINE IN ( S,F,BH,BV,AL,P,W,U,E,WU,WE,AL1,AL2,A1,A2,A3,A4,CX,00033100
C****     APU001    **   ARGUMENTS SENT   --  IN                        00033110
     1FH,FV,ID,NMD,ITD,JTD,MMD,NND,AF,ITPD,                             00033120
     2 WK,XQT,XAA,XBB,XCC,XT,KK,KM,XL,II,IID,MMAX                    )  00033130
C                                                                       00033140
      COMMON IA(250)                                                    00033150
C                                                                       00033160
      COMMON /SWEEP/ BA,BC,J,J1,J2                                      00033170
      DIMENSION S(NMD,ITD),F(NMD,ITD),BH(JTD,MMD),BV(ITD,MMD),          00033180
     1AL(NND,ITD),P(NMD,MMD),AF(ITPD,MMD)                               00033190
      DIMENSION  W(1),U(1),E(1),WU(10),WE(10),AL1(1),AL2(1),            00033200
     1A1(IT),A2(IT),A3(IT),CX(IT),ID(IT),FH(9),FV(1),A4(1)              00033210
C                                                                       00033220
      EQUIVALENCE (IA(57),MM),(IA(58),NM),(IA(64),IT),(IA(77),ITP),     00033230
     1(IA(175),JCONV),(IA(201),ZZ),(IA(202),BB),(IA(203),CC),           00033240
     2(IA(204),DD),(IA(205),T),(IA(206),QT),(IA(207),CT),(IA(210),NN),  00033250
     3(IA(211),AA),(IA(212),TI),(IA(213),TJ),(IA(214),TM)               00033260
C****     APU002    **   SETTING APU DIM. --  IN                       00033270
      DIMENSION  KK(MMAX),KM(MMAX),XL(MMAX),II(IT),IID(IT)              00033280
      DIMENSION  WK(IT),XQT(IT),XAA(IT),XBB(IT),XCC(IT),XT(     IT)     00033290
      DIMENSION WK1(10,68),XBH(10,9)                                    00033300
C/*                                                                     00033310
      COMMON /NEGID/ NEG,IDD(9),IDDD(9)                                 00033320
C                                                                       00033330
C     APU DECLARATION IN IN                                             00033340
C****     APU003    **   APU  SOURCE                                   00033350
      DO 1167 M=1,MMAX                                                  00033360
      K=KK(M)                                                           00033370
      XQT(*)=PMM(P(*,M),S(*,*))                                         00033380
      XAA(*)=U(M)*A1(*)                                                 00033390
```

Fig. 3.4-1

```
      XBB(*)=BA*E(M)*A3(*)                                              00033400
      IF(XL(M).LT.0.5) GO TO 1120                                       00033410
      XCC(*)=(AL1(M)+AL2(M))*A2(*)                                      00033420
      WK(*)=1.0/(XAA(*)+XBB(*)+XCC(*)+CX(*))                            00033430
      GO TO 1122                                                        00033440
 1120 WK(*)=1.0/(XAA(*)+XBB(*)        +CX(*))                           00033450
 1122 CONTINUE                                                          00033460
C/*                                                                     00033470
      DO 1210 IB=1,IT                                                   00033480
      I=II(IB)                                                          00033490
      IF(XL(M).LT.0.5) GO TO 1124                                       00033500
      T = (XAA(I)*BH(J,M)+XBB(I)*BV(I,M)+XCC(I)*AL(K,I)+XQT(I)) * WK(I) 00033510
      TM=2.0*T-AL(K,I)                                                  00033520
      GO TO 1126                                                        00033530
 1124 T = (XAA(I)*BH(J,M)+XBB(I)*BV(I,M)                +XQT(I)) * WK(I) 00033540
      TM=T                                                              00033550
 1126 CONTINUE                                                          00033560
      TI=2.0*T-BH(J,M)                                                  00033570
      TJ=2.0*T-BV(I,M)                                                  00033580
      IF(TI.LT.0.0) GO TO 1130                                          00033590
      IF(TJ.LT.0.0) GO TO 1130                                          00033600
      IF(TM.LT.0.0) GO TO 1130                                          00033610
      BH(J,M)=TI                                                        00033620
      BV(I,M)=TJ                                                        00033630
      AL(K,I)=TM                                                        00033640
      GO TO 1140                                                        00033650
 1130 ZZ=U(M)*A4(I+1)                                                   00033660
      AA=XAA(I)-ZZ                                                      00033670
      BB=XBB(I)/2.0                                                     00033680
      CC=A2(I)*AL1(M)                                                   00033690
      DD=A2(I)*AL2(M)                                                   00033700
      QT=XQT(I)                                                         00033710
      CT=CX(I)                                                          00033720
      CALL FIXUP(BH(J,M),BV(I,M),AL(K,I))                               00033730
 1140 XT(  I)=W(M)*T                                                    00033740
      WK1(M,I)=BH(J,M)                                                  00033750
 1210 CONTINUE                                                          00033760
      DO 1150 N=1,NM                                                    00033770
 1150 F(N,*)=F(N,*)+P(N,M)*XT(*)                                        00033780
C/*                                                                     00033790
 1167 CONTINUE                                                          00033800
      DO 1212 L=1,NEG                                                   00033810
 1212 XBH(*,L)=WK1(*,IDD(L))*A4(IDD(L))                                 00033820
      FH(*)=FH(*)+BC*PMM(WU(*),XBH(*,*))                                00033830
C/*                                                                     00033840
C/*                                                                     00033850
      IF(J2.EQ.J1)RETURN                                                00033860
      DO 1220 I=1,IT                                                    00033870
      I1=ID(I)                                                          00033880
 1220 FV(I1)=FV(I1)+A3(I)*AIPD(WE(*),BV(I,*))                           00033890
      RETURN                                                            00033900
      END                                                               00033910
```

Fig. 3.4-2 Subroutine IN of (r,z) and (r,θ)-geometry in vectorized TWOTRAN code

4. ANISN[3]

4.1 計算内容

1次元中性子輸送コードで，ロスアラモス科学研究所でもとのバージョンは作成された。原研では，遮蔽計算や核データのエネルギー群縮約計算で主に使用されている。NEA CPL のバージョンに基づく。

4.2 計算コードのテストデータの概要

ソース行数	約3,000
主　記　憶	1,064KB
CPU時間（M200）	2分48秒
入出力回数	2369
使用ファイル数	8ファイル，ブロック長19K

計算問題

遮蔽計算のためのもの

100群，平板形状106メッシュ，S_{32} の計算

収束判定10^{-6}

4.3 計算時間のかかるサブルーチン[8]

上記の計算問題の場合の実行時のサブルーチンの呼出回数と所要時間はFig. 4.1で示される。

核となるプログラム構造は，外部反復ループ，エネルギー群ループ，および内部反復ループの三重ループ構造をなす（Fig. 4.2参照）。FORTUNE分析が示すコストの高いサブルーチンのうち，S833（同一エネルギー群内の散乱方程式の求解ルーチン）は，その内部に内部反復ループを含んでいる。又，S824（下方エネルギー群への散乱源計算ルーチン）は，エネルギー群ループの内側にある。サブルーチンWATE（断面積の縮約ルーチン）は，反復ループの外側にある。以上，3サブルーチンで計算コストの98.7％を占める。

SUBROUTINE NAME	EXECUTION TIMES	COST (%)
S833	100	79.0
WATE	1	12.2
S824	100	7.6
SUMARY	1	1.1
WOT	139	0.1
CONVT	1	0.1
S821	2	0.0
GUTS	1	0.0
TP	1	0.0
CLEAR	4	0.0

Fig. 4.1 ANISN SAMPLE DATA-2

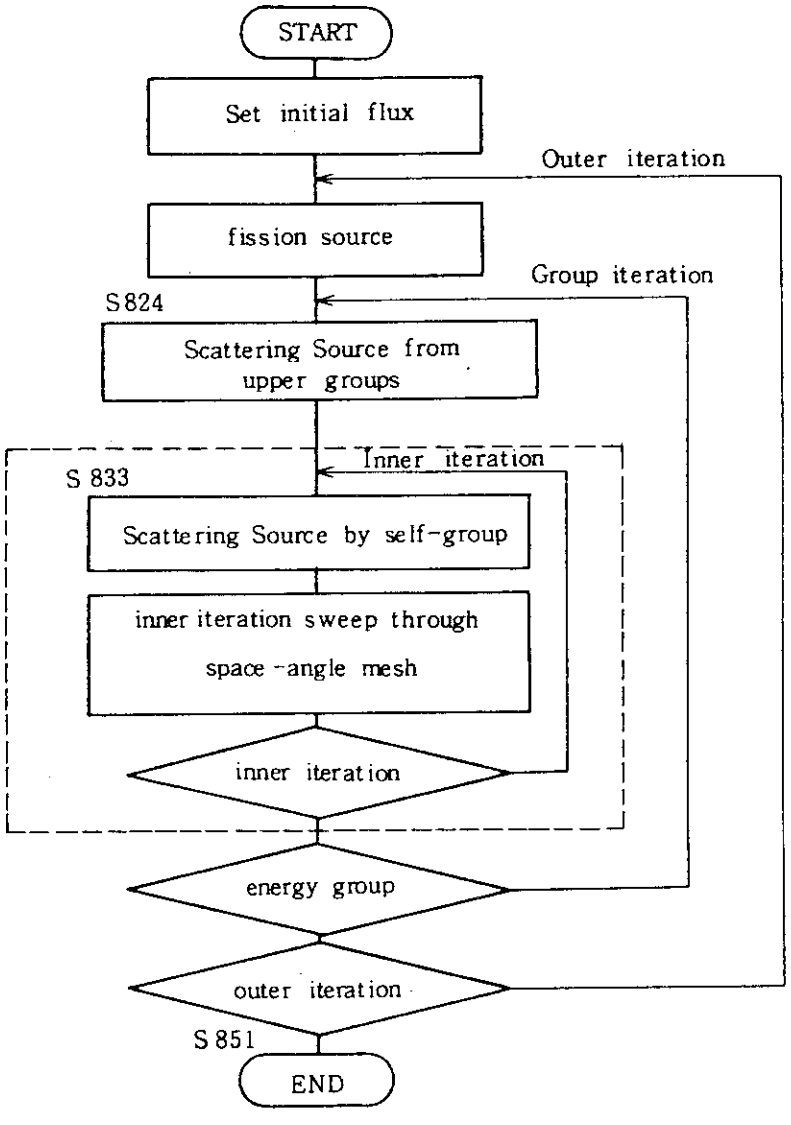


Fig. 4.2 Flow diagram of ANISN code

4.4 数値計算法の概要[3]

ボルツマンの輸送方程式を差分近似により離散化し，次式を得る。

$$\mu(A_{i+1}N_{i+1}-A_iN_i)+\frac{\alpha_{m+1/2}}{w}N_{m+1/2}-\frac{\alpha_{m-1/2}}{w}N_{m-1/2}+\Sigma_{tr}VN=SV \qquad (4.1)$$

ここでmは角度，iはメッシュ・インデックス，Nは中性子束を表わす。中性子束はFig. 4.3で示すようにメッシュ間隔，r_iからr_{i+1}にまたがるものとする。gはエネルギー群を示す。μは，散乱方向の余弦で，Aは形状に係わる定数（平板1，円筒$2\pi r_i$，球$4\pi r_i^2$）。wは角度μに対応するウェイトで$\sum_m w_m=1$，αは，平板の時は0で，曲線形状の場合には，次のように与えられる。

$$\alpha_{m+1/2}=\alpha_{m-1/2}-\tfrac{1}{2}w\mu(A_{i+1}-A_i)$$

Vは，ボリューム要素（平板Δx_i，円筒$\pi(r_{i+1}^2-r_i^2)$，球$4\pi(r_{i+1}^3-r_i^3)/3$），Σ_{tr}は群定数。

（4.1）をNについて解くと，

$\mu<0$のとき

$$N=\frac{2|\mu|AN_{i+1}+\frac{2\alpha}{w}N_{m-1/2}+S}{2|\mu|A+\frac{2\alpha}{w}+\sigma} \qquad (4.2)$$

ここで，

$$A=\frac{A_{i+1}+A_i}{2}$$

$$\alpha=\frac{\alpha_{m+1/2}+\alpha_{m-1/2}}{2} \qquad (4.3)$$

$\mu>0$のとき

$$N=\frac{2\mu AN_i+\frac{2\alpha}{w}N_{m-1/2}+S}{2\mu A+\frac{2\alpha}{w}+\sigma} \qquad (4.4)$$

（4.2）及び（4.4）式は，角度方向を最内ループとして計算する。$\mu<0$のときは，右から左へ，$\mu>0$のときは，左から右へ逐次的に計算される。

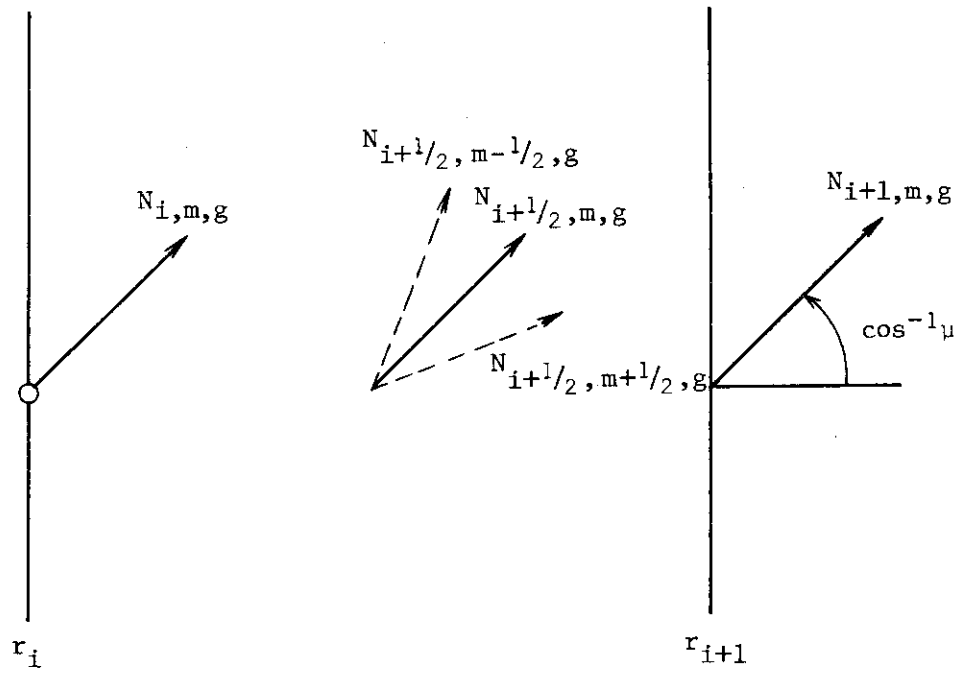


Fig. 4.3 Mesh interval for group, g in ANISN code

4.5 ベクトル化効果の推定[8]

サブルーチンS824は，コストの高いDOループ中に多数のIF文を含み，ベクトル化率は実質上，0％である。サブルーチンS833は，内部反復ループ（そのループ変数は空間座標の離散化分点である），及び角度分点に関するDOループなどを有する。空間座標に関する差分方程式の解法には，(3.2)－(4.4)式で示すような漸化式を用いており，再帰的データ参照関係のためにベクトル化できない。方程式の係数の計算はベクトル化可能である。従って，大巾なプログラムの書き換えなしでは，相対的に26％程度のベクトル化率と思われる。（この入力データの例では，空間分点のベクトル長＝106，角度分点のベクトル長＝33である）。又，サブルーチンWATEでは，小数のDOループにコストが集中しており，ループ制限変数の書き直しを行えば，相対的に41％のベクトル化が可能と思われる。プログラム全体では，21％程度のベクトル化率となる。これらの結果は，Table 4.1にまとめて示す。

このようにベクトル化率が低いと推定されるので，F75 APUによるベクトル計算の実測を行なわなかった。なお，ここでのベクトル化分析，推定結果は，56年度の調査時における富士通松浦氏によるもの[8]である。

(4.2)－(4.3)式をodd－even法[6]を用いてベクトル化するなど，計算アルゴリズムを変更してのベクトル化を今後試みてみたい。

Table 4.1 Dynamic Profile Analysis of ANISN-code

Subroutines	Execution times	Cost(%)	V_{rel}	V(%)	Comments
S824	100	7.5	0.	0.	Down scatter source
S833	100	79.0	0.26	20.5	Self-scatter equations
WATE	1	12.2	0.41	5.0	X-section reduction
Others	——	1.3	——	——	
		100.0	25.5		

5. PALLAS－2DCY[4]（2次元放射性輸送コード）

5.1 計算内容

船舶技研の竹内氏により作成されたコードで，（r，z）形状に対する中性子及び光量子の輸送問題を取扱う。遮蔽計算（つまり固定ソース）のみに適用できる。所内の各部門の遮蔽計算に使用されている。エネルギーについては，メッシュ毎に離散化され計算される。

5.2 計算コードとテスト問題の概要

ソース行数	6,500
主　記　憶	1,600 KB
CPU時間（M 200）	9分53秒
入出力回数	10,112
使用ファイル数	16，ブロック長 19KB

計算問題

中性子輸送問題（遮蔽研究室）

31群，（r，z）形状　52 × 26，領域数 7 × 2

角度分点は，Fig. 5.1で示す28点に固定されている。

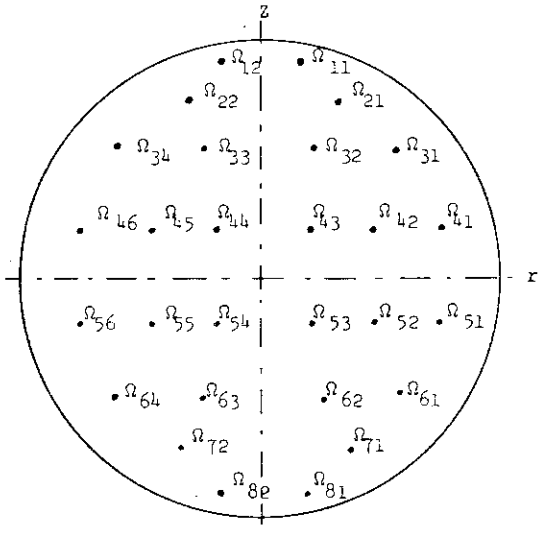


Fig. 5.1　Angular mesh points and their order in the current PALLAS code.

5.3 計算時間のかかるサブルーチン

計算時間のかかるサブルーチンの呼出回数と計算コストは，上記の問題に関してFORTUNEで測定した結果をTable 5.1で示す。

Table 5.1 Summary of computing time in PALLAS code

(calculation for neutron transport)

Subroutine	Cost ratio	CALL Times	Contents
NAGOYA	84.9 %	40	Calculation of neutron scattering source
KYOTO	10.5	15,680	Radiation transport calculation
OSAKA	2.8	40	Print angular and scalar fluxes
Others	1.8		

5.4 数値計算法の概要[4]

輸送方程式の直接積分法が使用され，方程式の積分は，各離散化された方向に対し，運動方向における粒子の飛躍パスに添って実行される。異方向散乱の取扱いや方程式を解くために，従来のルジャンドル展開近似や反復・収束法ではなく，中性子束計算には，微分散乱断面積データが直接使用され，また光量子に対しては，Klein－Nishina の公式が使用される。エネルギー・メッシュ法が使用されている。

ボルツマン輸送方程式

$$\bar{\Omega}\nabla\phi + \Sigma_t(\bar{r},E)\,\phi(\bar{r},\bar{\Omega},E)$$

$$=\iint dE'd\bar{\Omega}'\phi(\bar{r},\bar{\Omega}',E')\,\Sigma_s(\bar{r},E'\to E,\bar{\Omega}'\cdot\Omega)+S(\bar{r},\bar{\Omega},E) \qquad (5.1)$$

を直接積分することにより，次の積分方程式に変換する。

$$\phi(\bar{r},\bar{\Omega},E)=\phi(\bar{r}',\bar{\Omega},E)\,e^{-\Sigma_t R}$$

$$+\int_0^R Q(\bar{r}'',\bar{\Omega},E)\,e^{-\Sigma_t R'} \qquad (5.2)$$

ここでソース項は，次のように計算される。

$$Q(\bar{r},\bar{\Omega},E)=\iint\Sigma_s(\bar{r},E'\to E,\bar{\Omega}'\cdot\bar{\Omega})\,\phi(\bar{r},\bar{\Omega}',E')d\bar{\Omega}'dE'+S(\bar{r},\bar{\Omega},E) \qquad (5.3)$$

ここで，$\bar{r}$ は空間メッシュ，E はエネルギー，Ω は角度を表わす。Σ_s は微分散乱断面積を表わす。

計算時間を要するのは，（5.3）式の右辺第1項の散乱を計算する部分である。この項の計算では，微分散乱断面積を次のように与える。

a．中性子の弾性散乱

$$\Sigma_s(\bar{r}, E' \to E, \bar{\Omega}' \cdot \bar{\Omega}) = \Sigma_{el}(\bar{r}, E') f_{el}(E', \mu) \delta(\cos\theta - \alpha) \frac{(A+1)^2}{2AE'} \quad (5.4)$$

ここで θ は，極方向の角度，$\alpha = \cos\theta$，μ は重心系での極方向角度の余弦を表わす。A は原子核の質量を表わし，$E_{er}(\bar{r}, E')$，$f_{el}(E', \mu)$ は各々弾性散乱断面積と散乱分布関数を表わす（Fig. 5.2 参照）。

b．中性子の非弾性散乱

$$= \Sigma_{in}(\bar{r}, E') \frac{f_{in}(E' \to E)}{4\pi} \quad (5.5)$$

$E_{in}(\bar{r}, E)$ および $f_{in}(E' \to E)$ は非弾性散乱断面積を表わす。

c．光量子の散乱

$$= n(\bar{r}) \frac{K(E', E)}{2\pi} \left(\frac{E}{E'}\right) \delta(\cos\theta - \alpha) \quad (5.6)$$

$n(\bar{r})$ は電子密度，$K(E', E)$ は Klein－Nishina の公式から得られる値である。

（5.3）式右辺第一項を，エネルギーと角度方向に数値積分することにより，次式のように計算される。

a．中性子の弾性散乱

$$Q^N_{el}(\bar{r}, \bar{\Omega}_{pq}, E) = \sum_m \sum_n W_m \omega_n \Sigma_{el}(\bar{r}, E_m) f^m_{el}(E_m) I(\bar{r}, \bar{\Omega}_{ns}, E_m) \quad (5.7)$$

Ω_{pq} は角度メッシュ（Fig. 5.1 参照），W_m，ω_n は中性子散乱において，各々極方向および方位角方向に対応する重量を表わす。$I(\bar{r}, \bar{\Omega}, E)$ は，エネルギー中性子束で次のように与える。

$$I(\bar{r}, \bar{\Omega}, E) = E\Phi(\bar{r}, \bar{\Omega}, E) \quad (5.8)$$

b．中性子の非弾性散乱

$$Q^N_{in}(\bar{r}, \bar{\Omega}, E_j) = \sum_{k=1}^{j-1} \Sigma_{in}(r, E_k) \frac{f_{in}(\bar{r}, E_k \to E_j) E_j}{4\pi E_k} I_0(\bar{r}, E_k) \Delta E_k$$

$$= \sum_{k=1}^{j-1} \frac{\Sigma_{in}(\bar{r}, E_k \to E_j) E_j}{4\pi E_k} I_0(\bar{r}, E_k) \Delta E_k, \quad (5.9)$$

c．光量子の散乱

$$Q^G(\bar{r}, \bar{\Omega}_{pq}, \lambda_j) = \sum_m \sum_n \omega_n \, n(\bar{r}) \frac{K(\lambda_m, \lambda_j)}{2\pi} I(\bar{r}, \bar{\Omega}_{ns}, \lambda_m) \, \Delta\lambda_m \qquad (5.10)$$

ここでλは，Comptonの波長を表わす。

結局（5.3）式の右辺は次式のように書ける。

a．中性子の場合

$$Q(\bar{r}_i, \bar{\Omega}_{pq}, E_j) = Q^N_{el}(\bar{r}_i, \bar{\Omega}_{pq}, E_j) + Q^N_{in}(\bar{r}_i, \bar{\Omega}_{pq}, E_j) + S(\bar{r}_i, \bar{\Omega}_{pq}, E_j) \qquad (5.11)$$

b．光量子の場合

$$= Q^G(\bar{r}_i, \bar{\Omega}_{pq}, E_j) + S(\bar{r}_i, \bar{\Omega}_{pq}, E_j) \qquad (5.12)$$

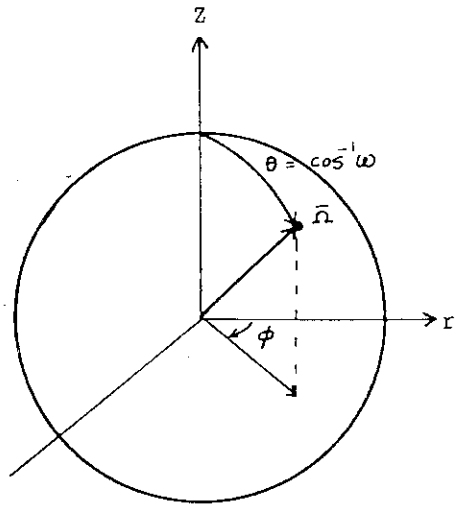


Fig. 5.2　Moving direction Ω of radiation in PALLAS.

5.5　ベクトル化効果

PALLASコードのベクトル化の問題点は，数値計算アルゴリズムにあるのではなく，そのプログラム構造にある。数値計算で時間を費すのは（5.9）－（5.11）の計算で，これは単なる乗積和に過ぎない。つまり，（5.9）－（5.11）式は，空間メッシュ，角度メッシュおよびエネルギーメッシュに対して独立に計算可能なので，潜在的には，ベクトル計算に向いている。

中性子の散乱計算は，サブルーチンNAGOYAで実行され，コード全体の計算時間の84.9％を占めている。この大部分は，ベクトル計算可能でその内訳はTable 5.2で示される。

しかしながら，サブルーチンNAGOYAのループ構造は，Fig. 5.3に示すとおり，領域とその構成要素（核種）のループが外側にあり，最内ループとして，領域内のメッシュ（15～315点，平均70点）に対し，中性子散乱の計算がなされ，ベクトル長がそれ程大きくない上に，間接アドレスの使用などで計算時間が増加する。この部分をこのままベクトル化すると，ベクトル化に要

するオーバヘッドで相殺される分を考慮しても約4倍の速度向上となる。

従って最内ループをベクトル化するだけで，以下の計算により約2.6倍の速度向上が得られる。

ベクトル計算の相対時間	0.83×0.25
スカラー計算の相対時間	0.17
合計の相対時間	0.377*

* スカラー計算を1とした相対時間

Table 5.2 Computing time of neutron scattering calculations in subroutine NAGOYA in PALLAS code

Contents	Computing time	vectorizability
Elastic scattering	61.5 %	Vectorizable
Inelastic scattering	18.2	Vectorizable
Total scattering	1.2	Vectorizable
Others	2.0	Vectorizable
	2.0	Not vectorizable
Total	84.9	Vectorizable 82.9

5.6 プログラムの再構成

現状におけるベクトル化は，Fig. 5.3の最内ループの変更によりなされた。まず領域に対するメッシュ対応表を先に作る。

N2　r方向の領域区分番号

N1　z方向の領域区分番号

し，次の表を作成する。

NM(N1, N2)　領域のメッシュ数（＝IN）

（MM(I, N1, N2), I＝1, 2,…, IN）（r, z）を1次元化した領域のメッシュ番号

サブルーチンNAGOYAの中性子弾性散乱の計算では，Fig. 5.4で示すようにプログラム構造を変更すれば良い。

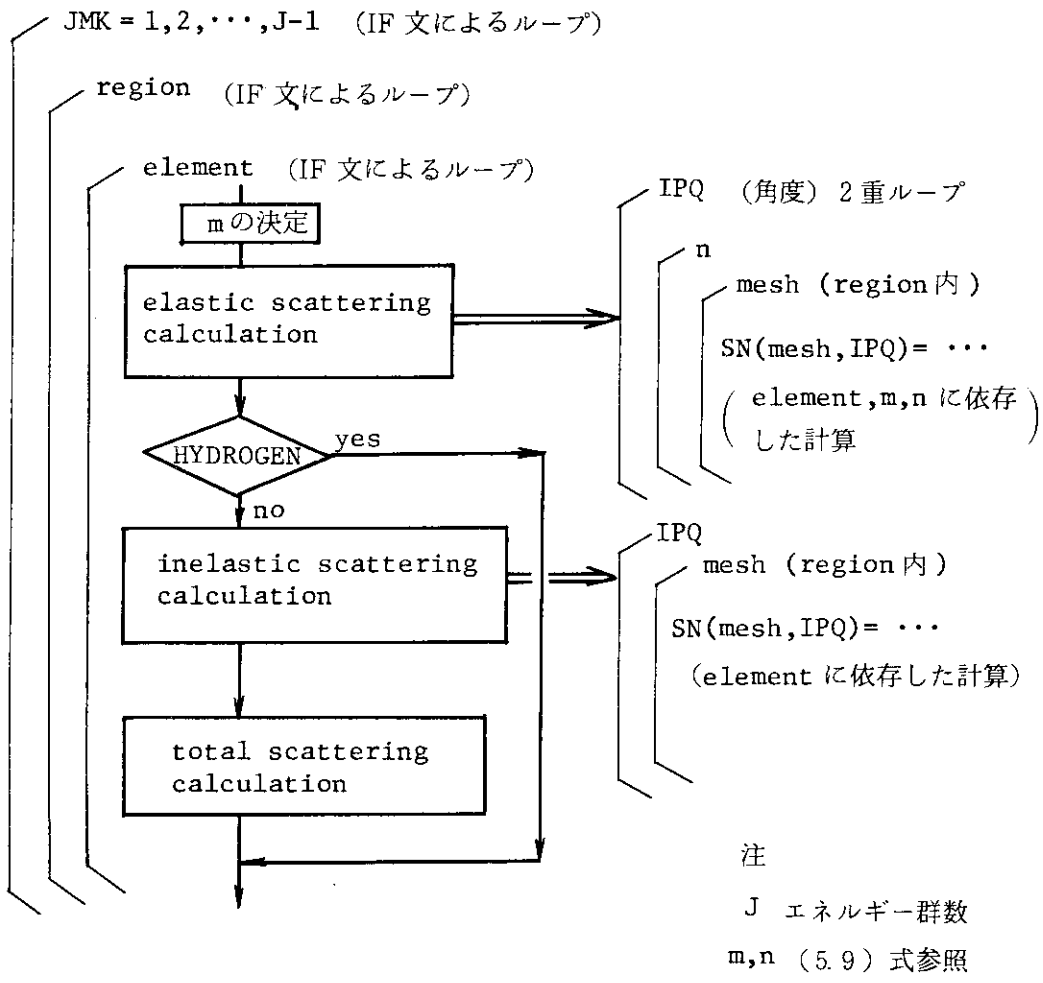


Fig. 5.3 Loop summary of subroutine NAGOYA in PALLAS code

```
      CALL PSU(N1,N2,MX1,MX2,MY1,MY2)
      DO 129  MY=MY1,MY2
      DO 129  MX=MX1,MX2
      SN(MY,MX,MPQ) = SN(MY,MX,MPQ) + AB * (FN(MY ,MX,IIN1) +
     1                FN(MY,MX,IIN2)) * X5
  129 CONTINUE
```

⇩ vectorization for region-mesh

```
      IN=NM(N1,N2)
      DO 129 I=1,IN
      IM(I)=MM(I,N1,N2)
      SN (IM(I),MPQ)=SN(IM(I),MPQ)+AB*(FN(IM(I),IIN1)+
                     FN(IM(I),IIN2)*X5
  129 CONTINUE
```

Fig. 5.4 Restructuring for vector processings of elastic scattering calculation in PALLAS code

NAGOYAの他の部分もこの方向でベクトル化し，PALLASコード全体の実行をF75 APUで行うとしたが，AP－FORTRANの障害により実行できなかった。そこで上記の弾性散乱の計算部分を取り出し，テストプログラムを作成し，F75 APUで実行した結果，2.7倍の速度向上を確認できた。再構成したプログラムは，Fig. 5.5に示される。

F75 APUでは，間接アドレス使用によるベクトル化を，1次元配列に制限しているので，余分な配列の置換えが必要となり，計算量が増加してしまう。今の場合，1つの文を実行するのに，3つの配列の置換が必要である。将来のベクトル計算機では，この点について解決されるはずであるから，計算量が3／7程度と成り，4倍の速度は出る。

5.7 計算速度をさらに上げるための考察

さらにベクトル長を大きく取り，速度向上を行うには，領域毎のメッシュではなく，全空間に対するメッシュを1回のベクトル計算で実行できれば，同様の計算がBERMUDAで5.3倍の速度向上となっていることから，コード全体としておそらく4倍程度の速度向上が得られるであろう。しかしながら，Fig. 5.3で示すように，SN計算に際し使用する係数（Fig. 5.4のプログラム・リストの変数AB，×5，IIN1，IIN2）が組成核種および（5.9）式のn，mに依存するので，NAGOYAサブルーチン（600行）の作り直しが必要となる。

PALLASコードの作成に当っては，物理モデル，数学モデル，プログラムの全てを，遮蔽分野の研究者である竹内氏（外来研究員として原研に出向中）が行った。このため，物理モデルの考え方がそのままプログラム化され，プログラム構造を第3者に対して複雑なものにしている。ベクトル計算により計算速度をさらに上げるためには作成者自らによる再構成しか方法がない。

```
ISN  ST-NO                SOURCE PROGRAM

 1          DIMENSION SN(55,55,28),FN(55,55,28)
 2          DIMENSION SNN(1),FNN(1)
 3          EQUIVALENCE (SN(1,1,1),SNN(1)),(FN(1,1,1,),FNN(1))
 4          DIMENSION IN1(8),IN2(8)
 5          DIMENSION ME2(10),MER(10),NM(10,10),MM(400,10,10)
 6          DIMENSION LIST(400),L1(400),L2(400),L3(400)
 7          INDEX  IX1/1,IIIN/                                  インデックスの設定
 8          DATA MER/15,6,5,3,9,9,5,3*0/
 9          DATA MEZ/21,5,8*0/
10          DATA IN1/3,6,9,12,15,18,21,24/
11          DATA IN2/3,6,9,12,15,18,21,24/
12          AB=123,45678
13          X5=987,65432
14          MPQ=21
15          DO 1 I=1,55
16          DO 1 J=1,55
17          DO 1 K=1,28
18          SN(I,J,K)=I+J+K
19          FN(I,J,K)=I*J*K
20        1 CONTINUE
21          DO 100 I=1,7
22          DO 100 J=1,2
23          NM(J,I)=MEZ(J)*MER(I)
24          CALL PSU(J,I,MX1,MX2,MY1,MY2)
25          ICNT=1
26          DO 200 II=MX1,MX2
27          DO 200 JJ=MY1,MY2
28          MM(ICNT,J,I)=55*(II-1)+JJ
29          ICNT=ICNT+1
30      200 CONTINUE
31      100 CONTINUE
32          CALL CLOCKM(ISTART)
33          DO 10 I+1,7
34          DO 10 J=1,2
35          DO 10 K=1,8
36          IIN1=IN1(K)
37          IIN2=IN2(K)
38          IIIN=NM(J,I)
39          LIST(IX1)=MM(IX1,J,I)
40          ITEMP1=(MPQ-1)*3025                                 I × 1 =
41          ITEMP2=(IIN1-1)*3025                                1, IIIN
42          ITEMP3=(IIN2-1)*3025                                までベク
43          L1(IX1)=LIST(IX1)+ITEMP1                            トル計算
44          L2(IX1)=LIST(IX1)+ITEMP2                            する。
45          L3(IX1)=LIST(IX1)+ITEMP3
46          SNN(L1(IX1))=SNN(L1(IX1))+AB*(FNN(L2(IX1))+FNN(L3(IX1)))*X5
47       10 CONTINUE
48          CALL CLOCKM(IEND)
49          ITIME=IEND-ISTART
50          WRITE(6,123) ITIME
51      123 FORMAT('   *****TIME *****',I10)
52          WRITE(6,1234) SN(1,1,1),FN(1,1,1)
53     1234 FORMAT(1X,2E15,5)
54          STOP
55          END
```

Fig. 5.5 Restructuring on F75-APU in PALLAS code (Test program)

6. BERMUDA－2DN[5]

6.1 計算内容

2次元（r，z）形状中性子輸送コード。P_∞モデル直接積分法，群モデルの結合により解く。遮蔽研鈴木氏作成中のコード。PALLASコードと異なるのは，エネルギーを離散化せず，通常のエネルギー群の取扱いがなされていることである。

ソース行数	約2500
主　記　憶	1,652KB
CPU時間	19.3分（Zメッシュ＝8のとき）
入出力回数	460（DWNSC2のみ）

6.2 テスト問題の計算時間

コードで時間を消費するのは，中性子束の計算と，下方散乱カーネルの計算である。

51群，角度分点40，組成3，rメッシュ35，判定$\varepsilon=10^{-3}$，Zメッシュ8，14，24の3通りの問題に対し，計算時間は，

下方散乱カーネルの計算		11.6分	合計19.3分
中性子束の計算	Z＝8	7.7分	
	Z＝14	16.5分	
	Z＝24	31.3分	

6.3 計算時間のかかるサブルーチンとその計算方法

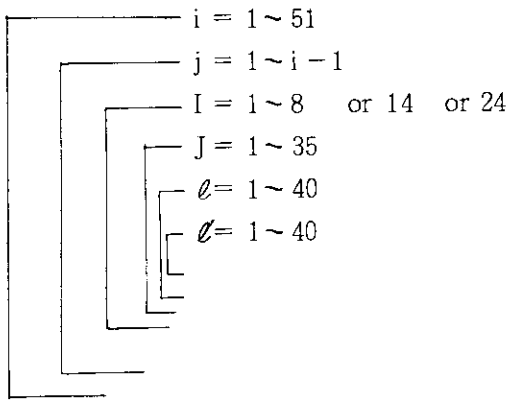
下方散乱を計算するDWNSC2	56％
中性子束の計算	34％
そ　の　他	10％

DWNSC2

$$SD(r_J, z_I, \Omega_\ell) = \sum_{j=1}^{i-1} \sum_{\ell'=1}^{40} \Sigma_s^{j\to i,\ \ell'\to\ell,\ M(r_J, z_I)}\ \phi^j(r_J, z_I, \Omega'_\ell)$$

i，jは群を示すインデックスで，j＝1からi－1までの$\Sigma_s^{j\to i,\ \ell'\to\ell,M}$と$\phi^j(r_J, z_I, \Omega_{\ell'})$をファイルから読み，掛け算して累加していくだけである。i＝51近くなるとCPUが上式の計算で大部分消費される。J，Iは空間インデックスを示す。

ループ構造



ベクトル計算は最内ループℓ'に対して行われその際DIMENSIONの変数順序をベクトル化する変数を連続して取るように変更した。これによって，ベクトル計算の時間が約20%短縮された。（Table 6.1）

6.4 F75 APUによる計算速度

DWNSC2のみF75 APUでベクトル計算し，その結果5.26倍の速度向上を得た（Table 6.1）中性子束の計算等のベクトル化の可能性もありコード全体としてベクトル化率は70%（DWNSC2のみ）～90%となり，2.2～3倍の速度向上が期待できる。

Table 6.1 Vectorization effect for BERMUDA－2DN Code (subroutine DWNSC2)

IG (群数)	M－200	75－CPU	① 75－APU	② 75－APU（注1）(DIMENSION変更)
1	5m秒	0	0	0
2	22	22	2	1
3	1045	2221	503	416
4	3073	6587	1504	1244
5	6108	13128	3004	2485
6		21901	5010	4140
7		32829	7509	6209
8		45862	10506	8696
9		61099	14002	11605
10		78463	18010	14914
51				
合計		2777.2秒	640.9秒（内CPU 3.5）	
速度倍率		1.0	4.33（注2）	5.26（注3）

（注1） A（L, *, M）⟹ A（*, L, M）と変更

（注2） $\frac{2777.2}{640.9} = 4.33$

（注3） $\frac{\text{CPU}}{\text{APU（DIMENSIONの順序変更）}} = \frac{7.8463}{14914} = 5.26$

7. 討　　議

この仕事は，著者ら2名が約6ケ月かかって行ってきた。

輸送コードのベクトル計算の適応性は，扱う問題，計算法，形状に大きく依存することは既に述べた。今後，できるだけ多くの問題について，ケース・スタディが必要である。

今回のベクトル化試行に際し，F75 APUで，意図通り実行できたのは，TWOTRAN，BERMUDAのみで，DOT 3.5，PALLASについては，AP－FORTRANのトラブルに合い，いたずらに日々を費やし，結局，カーネル部分のみしか計算できなかった。将来のスーパコンピュータは，どの機種を見ても，F75 APUのように，ベクトル計算を陽に記述する方式のものではなく，FORTRAN77の言語仕様そのものを使用するようである。この場合，コンパイラが自動的にベクトル化できるように再構成しておくことが必要である。スーパコンピュータを本格的に使用する場合には，1～2人月／コード程度の人手をさらに要するであろう。

直接積分法の方が明らかにベクトル計算向きと言えるが，よく使用されているNEA CPUから入手のものは差分法に基づくものに限られる。原研で作成された直接積分法によるコードが，これらに置き換えられる日が来ることを期待する。

またDOT 3.5で提案した反復解法の変更については，この分野の専門家がもっと本格的に研究するべきである。現在の差分法での潜在的に持つ逐次性をどう回避できるかについては，重要かつ興味あるテーマである。

謝　　辞

TWOTRANコードのベクトル化については，昭和52年に富士通・原子力システム開発部（当時）で行った調査[7]を全面的に利用させていただいたことについて，富士通の田子精男，南多善，長谷部芳郎の各氏に感謝する。またANISNコードについては，外来研究員の松浦氏によるベクトル計算適応性調査を引用させていただき，このリポートを多彩なものにすることができましたことを感謝する。

船舶技研の竹内氏（原研外来研究員）にはPALLASコード，住友原子力工業の山野氏（原研出向中）にはDOT 3.5コード，土橋氏（炉物理実験）にはTWOTRANコード，鈴木友雄氏（遮蔽）にはBERMUDAコードについてデータの提供を受け，またコードの利用に際し貴重な助言をいただきましたことを感謝する。

また，報告書作成にあたり，原稿をチェックし，貴重な指摘をいただきました中原（原子炉システム），鈴木友雄，浅井清（計算センタ）の各氏に感謝する。

7. 討　　議

この仕事は，著者ら2名が約6ケ月かかって行ってきた。

輸送コードのベクトル計算の適応性は，扱う問題，計算法，形状に大きく依存することは既に述べた。今後，できるだけ多くの問題について，ケース・スタディが必要である。

今回のベクトル化試行に際し，F75 APUで，意図通り実行できたのは，TWOTRAN，BERMUDAのみで，DOT 3.5，PALLASについては，AP－FORTRANのトラブルに合い，いたずらに日々を費やし，結局，カーネル部分のみしか計算できなかった。将来のスーパコンピュータは，どの機種を見ても，F75 APUのように，ベクトル計算を陽に記述する方式のものではなく，FORTRAN77の言語仕様そのものを使用するようである。この場合，コンパイラが自動的にベクトル化できるように再構成しておくことが必要である。スーパコンピュータを本格的に使用する場合には，1～2人月／コード程度の人手をさらに要するであろう。

直接積分法の方が明らかにベクトル計算向きと言えるが，よく使用されているNEA CPUから入手のものは差分法に基づくものに限られる。原研で作成された直接積分法によるコードが，これらに置き換えられる日が来ることを期待する。

またDOT 3.5で提案した反復解法の変更については，この分野の専門家がもっと本格的に研究するべきである。現在の差分法での潜在的に持つ逐次性をどう回避できるかについては，重要かつ興味あるテーマである。

謝　　辞

TWOTRANコードのベクトル化については，昭和52年に富士通・原子力システム開発部（当時）で行った調査[7]を全面的に利用させていただいたことについて，富士通の田子精男，南多善，長谷部芳郎の各氏に感謝する。またANISNコードについては，外来研究員の松浦氏によるベクトル計算適応性調査を引用させていただき，このリポートを多彩なものにすることができましたことを感謝する。

船舶技研の竹内氏（原研外来研究員）にはPALLASコード，住友原子力工業の山野氏（原研出向中）にはDOT 3.5コード，土橋氏（炉物理実験）にはTWOTRANコード，鈴木友雄氏（遮蔽）にはBERMUDAコードについてデータの提供を受け，またコードの利用に際し貴重な助言をいただきましたことを感謝する。

また，報告書作成にあたり，原稿をチェックし，貴重な指摘をいただきました中原（原子炉システム），鈴木友雄，浅井清（計算センタ）の各氏に感謝する。

参 考 文 献

(1) Mynatt, F,R; Development of Two-Dimensional Discrite Ordinates Transport Theory for Radiation Shielding, PHD Thesis of Univ. Tennessee (1969).

(2) Lathrop, K,D, et al.; TWOTRAN II : An Interfaced, Exportable Version of the TWOTRAN Code for Two-Dimensional Transport, LA-4848-MS (1973).

(3) Engle, W,W, et al.; DTF-II, A One-Dimensional, Multigroup Neutron Transport Program, NAA-SR-10951 (1966).

(4) Takeuchi, K; PALLAS-2DCY-FC, A Calculational Method and Radiation Transport Code in Two-Dimensional (R,Z) Geometry, Papers of ship Research Institute, No.57 (1979).

(5) Suzuki T., et al.; BERMUDA-2DN: A Two-Dimensional Neutron Transport Code, 6-th Inf. Conf. on Rad. Shieldings, Tokyo, May 16-20. (1983).

(6) 石黒，難波：差分法のベクトル計算，情報処理学会誌 24， 1 （1983）

(7) 田子，南，長谷部，他：原子力コードのFACOM230－75APUプログラムへの変換，内部資料，富士通原子力システム開発部（1977）。

(8) 石黒，松浦，他：大型原子力コードのベクトル計算

(9) Mynatt, F,R, et al.; The DOT III Two-Dimensional Discrete Ordinates Transport Codes, ORNL-TM-4280 (1973)