

JAERI-Data/Code

98-025



MOSRA-Light;ベクトル計算機のための  
高速3次元中性子拡散ノード法コード

1998年10月

奥村啓介

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

本レポートは、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。  
入手の問合せは、日本原子力研究所研究情報部研究情報課（〒319-1195 茨城県那珂郡東海村）あて、お申し越しください。なお、このほかに財団法人原子力弘済会資料センター（〒319-1195 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。

This report is issued irregularly.

Inquiries about availability of the reports should be addressed to Research Information Division, Department of Intellectual Resources, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken, 319-1195, Japan.

© Japan Atomic Energy Research Institute, 1998

編集兼発行 日本原子力研究所

## MOSRA-Light ; ベクトル計算機のための高速3次元 中性子拡散ノード法コード

日本原子力研究所東海研究所エネルギーシステム研究部  
奥村 啓介

(1998年9月7日受理)

MOSRA-Lightは、4次の多項式展開ノード法(NEM)に基づく、X-Y-Z体系3次元中性子拡散計算コードである。4次のNEMはメッシュ分割の大きさに敏感でないため、20cm程度の粗いメッシュを使用しても正確な計算が可能である。3次元問題においては未知数の数が劇的に少なくできるため、結果として非常に高速な計算が可能となる。

更に、本コードではベクトル計算機に適した「境界分離チェックカードスワイプ法」を新たに開発して採用している。この手法は、問題の規模が大きくなるほど高速化率も増大する特長があり、極めて効率的である。PWR炉心計算の例では、スカラー計算との比較で20倍～40倍の高速化率が得られる。ベクトル化と粗メッシュ法の両効果を合わせると、従来の有限差分法に基づくスカラーコードに比べて1000倍以上の高速化率となる。

MOSRA-Lightは、UNIXあるいはこれと同等なオペレーティングシステム(例えばLinuxなどのフリーウエア)を使用するほとんどのベクトルまたはスカラー計算機で利用することができる。また、ユーザーは会話型インストーラにより容易にコードをインストールすることができる。

本報告書では、NEMの一般理論、スーパーコンピュータ用の高速計算アルゴリズム、ベンチマーク計算結果、入力の手引きや入力データ例などの詳しいコードの利用方法について記述する。

## MOSRA-Light ; High Speed Three-Dimensional Nodal Diffusion Code for Vector Computers

Keisuke OKUMURA

Department of Nuclear Energy System  
Tokai Research Establishment  
Japan Atomic Energy Research Institute  
Tokai-mura, Naka-gun, Ibaraki-ken

(Received September 7, 1998)

MOSRA-Light is a three-dimensional neutron diffusion calculation code for X-Y-Z geometry. It is based on the 4th order polynomial nodal expansion method (NEM). As the 4th order NEM is not sensitive to mesh sizes, accurate calculation is possible by the use of coarse meshes of about 20 cm. The drastic decrease of number of unknowns in a 3-dimensional problem results in very fast computation. Furthermore, it employs newly developed computation algorithm "boundary separated checkerboard sweep method" appropriate to vector computers. This method is very efficient because the speedup factor by vectorization increases, as a scale of problem becomes larger. Speed-up factor compared to the scalar calculation is from 20 to 40 in the case of PWR core calculation. Considering the both effects by the vectorization and the coarse mesh method, total speedup factor is more than 1000 as compared with conventional scalar code with the finite difference method.

MOSRA-Light can be available on most of vector or scalar computers with the UNIX or its similar operating systems (e.g. freeware like Linux). Users can easily install it by the help of the conversation style installer. This report contains the general theory of NEM, the fast computation algorithm, benchmark calculation results and detailed information for usage of this code including input data instructions and sample input data.

**Keywords:** Three-Dimensional Neutron Diffusion Calculation Code, Nodal Expansion Method, NEM, Vector Computers, Core Calculation, Nuclear Design

## 目 次

1.はじめに .....	1
2.多項式展開拡散ノード法 .....	3
2.1 中性子バランス式とノード平均パラメータ .....	3
2.2 トランスマース積分による1次元中性子バランス式 .....	8
2.3 4次の多項式展開近似 .....	14
2.4 重み付き残差法 .....	19
2.4.1 1次元問題における2次の多項式展開 .....	19
2.4.2 重み付き残差法の導入(4次の多項式展開) .....	20
2.5 応答行列式の導出 .....	25
2.5.1 1次元問題2次多項式展開における応答行列式 .....	25
2.5.2 3次元問題2次多項式展開における応答行列式 .....	26
2.5.3 3次元問題4次多項式展開における応答行列式 .....	28
2.6 ノード内トランスマース漏洩分布の近似 .....	31
2.6.1 平坦近似 .....	31
2.6.2 バックリング近似 .....	32
2.6.3 2次式近似 .....	33
2.7 不連続因子の導入 .....	37
2.7.1 不連続因子の定義 .....	37
2.7.2 不連続因子を用いた部分中性子流の更新 .....	39
2.8 境界条件 .....	41
2.8.1 真空境界条件(外挿境界条件) .....	41
2.8.2 反射境界条件 .....	44
2.8.3 周期境界条件 .....	45
2.8.4 90度回転境界条件 .....	46
2.8.5 内部黒体 .....	49
2.8.6 対角対称境界条件 .....	53
2.8.7 アルベド境界条件 .....	64
2.9 計算フローの概要 .....	66
3.スーパーコンピュータのための高速計算アルゴリズム .....	68
3.1 様々なスーパーコンピュータ .....	68
3.2 スーパーコンピュータのためのプログラミング技法 .....	71
3.2.1 ベクトル計算機 .....	71
3.2.2 並列計算機 .....	78
3.3 高速計算の実現と問題点 .....	85
3.3.1 ソフトウェアの重要性 .....	85

3.3.2 並列計算の問題点 .....	86
3.4 MOSRA-Lightコードの高速計算アルゴリズム .....	90
3.4.1 独立した演算のための準備 .....	90
3.4.2 境界分離チェックボードスワイプ .....	91
3.4.3 並列計算機への適用性 .....	93
3.5 MOSRA-Lightコードの高速計算性能 .....	97
3.5.1 IAEA3次元ベンチマーク問題 .....	97
3.5.2 MOSRA-Lightコードにおける計算時間の分布 .....	98
3.5.3 有限差分コード(CITATION)との比較 .....	99
4. MOSRA-Lightコードの機能と詳細 .....	108
4.1 コードの概要 .....	108
4.2 MOSRA-Lightコードの仕様 .....	109
4.2.1 固定源問題 .....	109
4.2.2 隨伴問題 .....	111
4.2.3 熱群反復計算 .....	114
4.2.4 詳細メッシュ計算モード .....	117
4.2.5 加速法 .....	120
4.2.6 初期値とリストアート計算 .....	124
4.2.7 収束判定と反復計算の制御 .....	127
4.2.8 SRACコードの巨視的断面積ファイルの利用 .....	133
4.2.9 (n,2n)反応の補正 .....	134
4.2.10 計算結果の規格化 .....	134
4.2.11 内部黒体の拡張利用 .....	139
4.2.12 反応率バランス .....	141
4.2.13 入出力座標系の選択 .....	149
4.2.14 編集領域(Edit-Region)の利用 .....	152
4.2.15 ユーザー編集ルーチン .....	155
4.2.16 ユーティリティプログラム(More-MOSRA) .....	157
5. MOSRA-Lightの入力 .....	158
5.1 自由形式入力 .....	158
5.2 入力データの構成 .....	162
5.3 入力データの作成 .....	164
5.3.1 全体計算制御(セクション-1) .....	164
5.3.2 ケース毎の計算制御(セクション-2) .....	164
5.3.3 幾何形状入力(セクション-3) .....	175
5.3.4 物質指定(セクション-4) .....	180
5.3.5 初期値参照ケースの指定(セクション-5) .....	182

5.3.6	断面積値入力(セクション-6) .....	182
5.3.7	編集領域(Edit-Region)の指定(セクション-7) .....	184
5.3.8	バックリング入力(セクション-8) .....	185
5.3.9	黒体の入力(セクション-9) .....	185
5.3.10	外部源の入力(セクション-10) .....	186
5.3.11	不連続因子の入力(セクション-11) .....	187
6.	実行用シェルスクリプトと入力例 .....	191
6.1	実行用シェルスクリプト .....	191
6.2	入力データの例 .....	197
6.2.1	IAEA3次元ベンチマーク .....	197
6.2.2	IAEA2次元ベンチマーク .....	200
6.2.3	多群格子計算 .....	203
6.2.4	その他の入力例 .....	206
7.	入出力ファイル .....	210
7.1	入出力機番とプリント出力ファイル .....	210
7.2	バイナリーファイル(PDS)の構造 .....	211
7.3	MACROファイルの内容 .....	213
7.4	EGGファイルの内容 .....	219
参考文献	.....	227
付録-A	MOSRA-Lightコードのインストール .....	229
A.1	会話型インストーラ .....	229
A.2	ベクトル制御指示行の変更 .....	234
A.3	可変長配列サイズの変更 .....	236
A.4	原研計算機での利用 .....	237
付録-B	トラブル対策 .....	238
B.1	エラーによる計算の中止 .....	238
B.2	反復計算が収束しない場合の対処 .....	241
B.3	計算コストの節約 .....	242

## Contents

1.	Preface .....	1
2.	General Theory of Nodal Diffusion Expansion Method (NEM).....	3
2.1	Neutron Balance Equation and Nodal Averaged Parameters .....	3
2.2	One-Dimensional Neutron Balance Equation by Transverse Integration .....	8
2.3	The 4th-Order Polynomial Nodal Expansion Method.....	14
2.4	Weighted Residual Method .....	19
2.4.1	The 2nd Order NEM for 1-Dimensional Problem.....	19
2.4.2	Introduction of Weighted Residual Method in the 4-th Order NEM .....	20
2.5	Derivation of Response Matrix Equation .....	25
2.5.1	Response Matrix Equation for the 2nd-Order NEM in 1-Dimensional Problem.....	25
2.5.2	Response Matrix Equation for the 2nd-Order NEM in 3-Dimensional Problem.....	26
2.5.3	Response Matrix Equation for the 4th-Order NEM in 3-Dimensional Problem .....	28
2.6	Transverse Leakage Fitting Approximation .....	31
2.6.1	Flat Transverse Leakage Approximation .....	31
2.6.2	Buckling Transverse Leakage Approximation.....	32
2.6.3	Quadratic Transverse Leakage Fitting Approximation .....	33
2.7	Introduction of Discontinuity Factor .....	37
2.7.1	Definition of Discontinuity Factor .....	37
2.7.2	Updating of Partial-In-Current with Discontinuity Factor .....	39
2.8	Boundary Condition.....	41
2.8.1	Vacuum Boundary Condition (Extrapolated Boundary Condition).....	41
2.8.2	Reflective Boundary Condition.....	44
2.8.3	Periodic Boundary Condition .....	45
2.8.4	90-Degrees Rotational Boundary Condition .....	46
2.8.5	Internal Blackness .....	49
2.8.6	Diagonal Symmetric Boundary Condition .....	53
2.8.7	Albedo Boundary Condition .....	64
2.9	Outline of Calculational Flow .....	66
3.	Fast Calculation Algorithm for Super-Computers .....	68
3.1	Various Super-Computers .....	68
3.2	Programing Techniques for Super-Computers .....	71
3.2.1	Vector Computers.....	71
3.2.2	Parallel Computers .....	78
3.3	Realization of Fast Computation and Problems of Super-Computers.....	85

3.3.1	Importance of Software .....	85
3.3.2	Problems of Parallel Computers .....	86
3.4	Fast Computation Algorithm in MOSRA-Light Code .....	90
3.4.1	Preparation for Independent Calculation .....	90
3.4.2	Boundary Separated Checker-Board Sweep .....	91
3.4.2	Aptitude for Parallel Computers.....	93
3.5	Performance of Fast Computation by MOSRA-Light.....	97
3.5.1	IAEA 3-Dimensional Benchmark Problem .....	97
3.5.2	Cost Analysis for Computation Time.....	98
3.5.3	Comparison with Finite Difference Method (CITATION) .....	99
4.	Functions and Details of the MOSRA-Light Code .....	108
4.1	Outline of Code.....	108
4.2	Specification of MOSRA-Light.....	109
4.2.1	Fixed Source Problem .....	109
4.2.2	Adjoint Problem .....	111
4.2.3	Thermal Energy Group Iteration .....	114
4.2.4	Fine Mesh Calculational Mode .....	117
4.2.5	Acceleration Techniques .....	120
4.2.6	Initial Guess and Restart Calculation .....	124
4.2.7	Convergence Criteria and Automatic Control of Iterative Processes.....	127
4.2.8	Use of Macroscopic Cross Section Data from the SRAC Code.....	133
4.2.9	Correction for (n,2n) Reaction .....	134
4.2.10	Normalization of Calculated Results.....	134
4.2.11	Extended Use of Internal Blackness.....	139
4.2.12	Reaction Balance .....	141
4.2.13	Selective ordinate .....	149
4.2.14	Use of Edit-Region .....	152
4.2.15	Users' Editing Routine .....	155
4.2.16	Utility Program (More-MOSRA) .....	157
5.	Input Data Requirements for MOSRA-Light.....	158
5.1	Free Format Input .....	158
5.2	Whole Structure of Input Data.....	162
5.3	Preparation of Input Data.....	164
5.3.1	Control of Whole Calculation (Section-1) .....	164
5.3.2	Control of case-wise (Section-2) .....	164
5.3.3	Geometrical Data (Section-3).....	175
5.3.4	Material Specification (Section-4) .....	180

5.3.5	Reference Case for Initial Guess (Section-5) .....	182
5.3.6	Macroscopic Cross Section Data (Section-6).....	182
5.3.7	Edit-Region (Section-7) .....	184
5.3.8	Buckling (Section-8) .....	185
5.3.9	Blackness (Section-9).....	185
5.3.10	External Source (Section-10) .....	186
5.3.11	Discontinuity Factor (Section-11).....	187
<b>6</b>	<b>Sample Shell-Script and Input Data.....</b>	<b>191</b>
6.1	Sample Shell-Script for Execution .....	191
6.2	Sample Input Data .....	197
6.2.1	IAEA 3-Dimensional Benchmark Problem .....	197
6.2.2	IAEA 2-Dimensional Benchmark Problem.....	200
6.2.3	Multi-Groups Lattice Calculation.....	203
6.2.4	Other Samples .....	206
<b>7</b>	<b>I/O Files of MOSRA-Light.....</b>	<b>210</b>
7.1	I/O Devices and Print-Out Files .....	210
7.2	Structure of Binary File (PDS;Partitioned Data Set File).....	211
7.3	Contents of MACRO File (PDS).....	213
7.4	Contents of Egg File (PDS) .....	219
<b>References.....</b>	<b>227</b>	
<b>APPENDIX-A</b>	<b>Installation of the MOSRA-Light Code.....</b>	<b>229</b>
A.1	Conversational Style Installer .....	229
A.2	Convergence of Vector Control Statements .....	234
A.3	Extension of Variable Dimension Size.....	236
A.4	Use of MOSRA-Light on JAERI Computers .....	237
<b>APPENDIX-B</b>	<b>Trouble Shooting.....</b>	<b>238</b>
B.1	Error Stops .....	238
B.2	Measures against Convergence Failing .....	241
B.3	Reduction of Computation Cost .....	242

## List of Tables

Table 3.5.1	Two group constants for IAEA benchmark problem .....	101
Table 3.5.2	Mesh effect in IAEA 3D benchmark calculation by MOSRA-Light code .....	101
Table 4.2.1	Speedup factor by acceleration of the 4th order NEM using the 2nd order NEM in IAEA 3D benchmark calculation on vector computer (FACOM/VPP-500,1-CPU).....	123
Table B.3.1	Required computation cost for execution of sample problems on several computers. ....	243

## List of Figures

Fig. 3.1.1	Image of data processing on various computers.....	69
Fig. 3.1.2	Classification of parallel computers by memory type .....	70
Fig. 3.2.1	Programing techniques for supercomputers (Vector Computers) .....	81
Fig. 3.2.2	Checkerboard sweep to avoid recursive operation in vector or parallel computers.....	82
Fig. 3.2.3	Memory Address of multi-dimensional array and order of memory access depending on programs. Sequential memory access is recommended for speedup. ....	83
Fig. 3.2.4	Programing techniques for supercomputers (Parallel Computers).....	84
Fig. 3.3.1	Speedup factor depending on vectorization ratio and speed ratio of vector and scalar calculation in vectorized parts ( $\alpha$ ) .....	89
Fig. 3.3.2	The present state of parallel computations in the field of reactor analyses .....	89
Fig. 3.4.1	Outline of NEM (Nodal Expansion Method) Calculation.....	95
Fig. 3.4.2	Boundary separated checkerboard sweep for vector computers .....	96
Fig. 3.5.1	Core geometry of IAEA 3-dimensional benchmark problem .....	102
Fig. 3.5.2	Comparison of k-eff and assembly averaged power distribution in IAEA 3D benchmark..	103
Fig. 3.5.3	Distribution of computation time in IAEA 3D benchmark calculation by MOSRA-Light .	104
Fig. 3.5.4	Distribution of computation time in IAEA 3D benchmark calculation by MOSRA-Light	

(vector mode).....	105
Fig. 3.5.5 k-eff dependence on number of mesh points (N) in IAEA 3D benchmark calculation by CITATION and VENTURE.....	106
Fig. 3.5.6 Mesh effect on k-eff error and required computation time in IAEA 3D benchmark calculation by CITATION .....	106
Fig. 3.5.7 Mesh effect on assembly power error and required computation time in IAEA 3D benchmark calculation by CITATION.....	107
Fig. 4.2.1 Comparison of mesh effect on k-eff among the 4th order NEM (MOSRA-Light), the 2nd order NEM (MOSRA-Light) and the finite difference method (CITATION) in the axial 1D calculation for the inner fuel in IAEA benchmark problem.....	119
Fig. 5.2.1 Structure of input data for the MOSRA-Light code.....	162
Fig. 6.2.1 Core geometry of IAEA 2-dimensional benchmark problem .....	208
Fig. 6.2.2 Comparison of k-eff and assembly power distribution in IAEA 2D benchmark .....	208
Fig. 6.2.3 Geometry of multi-group lattice calculation for sample input for MOSRA-Light .....	209

## 1. はじめに

中性子拡散方程式の解法として最も古典的な数値解法は有限差分法である。有限差分法は、空間を微小メッシュ空間に分割して微分方程式を解くもので、拡散コードとしてはCITATION<sup>1)</sup>及びその改良版であるVENTURE等<sup>2)</sup>に広く使用されている。有限差分法は、それ以外の近似を取り込む必要がないため、メッシュサイズが十分に小さければ厳密解に近づき、しばしば参照解提供コードとして使用される。しかし、軽水炉体系では、中性子の平均自由行程が短いため、正確な解を得るために、1cm～2cm程度の詳細メッシュが必要になることがある。例えば、炉心体積が3万リットル以上ある標準的なPWRを3次元詳細メッシュを使用してまともに解くには、反射体領域を除いても数千万ものメッシュ分割が必要となる。このため、大型炉の設計や炉心燃焼管理に詳細メッシュを使用する3次元有限差分法コードを利用するることは、現在の高速計算機をもってしても計算コスト(必要メモリーと計算時間)が大きくなりすぎて実用的ではない。また、臨界集合体のような場合には、メッシュサイズを変更した計算から外挿によりメッシュ効果の補正を加えることができるが、炉心燃焼計算や空間依存動特性計算では、その都度メッシュ効果の補正を加えるわけにはいかない。

径方向と軸方向の中性子束分布の分離性が良いPWRの炉心計算では、詳細メッシュを用いた径方向2次元計算と軸方向1次元計算を組み合わせて解析する方法が採られてきた。一方、分離性が無視できないBWRの炉心計算では、GEのFLAREコード<sup>3)</sup>に代表されるように、早くから3次元炉心体系を粗メッシュ空間(ノード)に分割する計算法が採用され、いわゆるBWRシミュレータへと発展した<sup>4)</sup>。これら初期のノード法コードは、Green関数の原理に基づいているが、Green関数を論理的に求めるのではなく、これをノード伝達行列(nodal transfer matrix)というものに置き換える。そしてノード伝達行列の各要素(ノード結合係数: nodal coupling coefficients)は、解析対象とする標準的な炉心体系に対する有限差分法の解との比較や多くの運転経験などにより決定される。このため、主要な炉心設計パラメータに変更がある場合にはノード結合係数を見直す必要がある他、解析結果がユーザーに依存し易いという問題点があった。

一方、原研では、計算コストが高い軽水炉の設計や炉心燃焼管理はあまり必要とされず、むしろ臨界集合体の解析を正確に実施することが要求された。このため、初期のノード法のように、経験的な調整因子を導入して計算の高速化を図る手法は避けられ、結果的に近代ノード法の研究へと進展しなかった。むしろ、厳密な解法である有限差分法をベクトル化により高速化する研究が行われた。

近代ノード法の発展により、経験的な調整因子を必要としない高速高精度な計算手法がほぼ確立され、近年では、PWRやBWRメーカなどで広く利用されるようになってきた。原研においても、将来型炉の設計研究や空間依存動特性解析コードの需要により、より高速な3次元拡散コードが必要になってきた。このため、近代ノード法の手法とこれまで培つて

きたスーパーコンピュータの技術を取り入れ、高速な近代ノード法をより高速にする研究を行い、MOSRA-Lightコード(X-Y-Z体系版)として完成した。

近代ノード法の原理を大雑把に言うと、粗メッシュ内の中性子束分布を拡散方程式に従う解析関数または多項式などにより表現することにより、粗メッシュの採用による計算精度の低下を補い、空間変数を大幅に減らして高速化を図るものである。粗メッシュ内の中性子束分布をどのような関数で表現するかにより、近代ノード法に多くのバリエーションが存在し、解析的ノード法(Analytical Nodal Method)<sup>5)</sup>、多項式展開ノード法(Nodal Expansion Method)<sup>6)</sup>、解析的多項式ノード法<sup>7)</sup>、グリーン関数ノード法(Nodal Green's Function Method)<sup>8)</sup>などが存在する。近代ノード法の発展と様々な手法については、LawrenceやTakeda等のレビュー論文<sup>9)-11)</sup>に良くまとめられている。それぞれの解法に一長一短があるが、以下の観点からMOSRA-Lightコードには多項式展開拡散ノード法を採用した。

- (1) 多群コードとして的一般性(2群に限定されない)
- (2) 六角形状への発展性
- (3) 要素コードとしての手法の単純性(小規模)
- (3) ベクトル計算機に適した高速計算アルゴリズムの適用性

MOSRA-Lightは、近代粗メッシュノード法の採用とこれに適したベクトル計算アルゴリズム(境界分離チェックボードスウェーブ)の開発により、従来の有限差分法コードと比べ、同等かつ十分な計算精度を得るために必要な計算時間を数千分の一以下に短縮している。また、MOSRA-Lightは、モジュラーコードとしての特徴を持ち、高速な計算を必要とする大型の核熱結合炉心燃焼計算コードや空間依存動特性解析コードに容易に移植できるようプログラム上の配慮をしている。また、会話型のインストーラを準備しており、UNIXをOSとするほとんどのマシンにだれもが容易に移植することができる。

本報告書の構成は以下の通りである。第2章では、Lawrenceのレビュー論文<sup>9)</sup>に基づき、MOSRA-Lightが採用する多項式展開拡散ノード法(X-Y-Z体系)の一般的な理論について述べる。第3章では、スーパーコンピュータ(ベクトル計算機と並列計算機)の仕組みを簡単に説明するとともに、これらに適したMOSRA-Lightコードの高速計算アルゴリズムやベンチマーク計算問題への適用結果について述べる。第4章では、第2章の一般論では述べなかったMOSRA-Lightコード特有の計算手法や機能について述べる。第5章から第7章では、入力データ、ジョブ制御文(シェルスクリプト)の作成、I/Oファイルの内容など、MOSRA-Lightの具体的な利用方法について述べる。MOSRA-Lightの利用者は第4章から第7章までを読めば十分であるが、ノード法の一般論やスーパーコンピュータによる高速化プログラミングに興味がある方は、ぜひ第2章及び第3章も読んで頂きたい。最後の付録にはMOSRA-Lightコードのユーザー利用機種へのインストール方法や計算実行時のトラブル対策などについて記述してある。

## 2. 多項式展開拡散ノード法

### 2.1 中性子バランス式とノード平均パラメータ

本節では、X-Y-Z座標系において、着目する粗メッシュノードに対する中性子のバランス式を導出するとともに、種々のノード(体積及び表面)平均パラメータを定義する。基礎方程式は以下の定常多群の中性子バランス式である。

$$\nabla \cdot J_g(r) + \Sigma_{r,g}(r)\phi_g(r) = Q_g(r) \quad (2.1.1)$$

これに、次のFickの法則を取り込めば定常拡散方程式となる。

$$J_g(r) = -D_g(r)\nabla\phi_g(r) \quad (2.1.2)$$

全体系を粗いメッシュ空間(ノード)に分割しても、各ノードで(2.1.1)式と(2.1.2)式は成立する。そこで、x-y-z体系でk番目のノードに着目し、ノード内の断面積は一様であると仮定すると、ノード内部で次式が成立する。ただし、座標中心は着目ノードの中心とする。

$$\frac{\partial}{\partial x} J_{g,x}^k(x,y,z) + \frac{\partial}{\partial y} J_{g,y}^k(x,y,z) + \frac{\partial}{\partial z} J_{g,z}^k(x,y,z) + \Sigma_{r,g}^k \phi_g(x,y,z) = Q_g^k(x,y,z) \quad (2.1.3)$$

$$(-\frac{1}{2}\Delta_x^k \leq x \leq \frac{1}{2}\Delta_x^k), \quad (-\frac{1}{2}\Delta_y^k \leq y \leq \frac{1}{2}\Delta_y^k), \quad (-\frac{1}{2}\Delta_z^k \leq z \leq \frac{1}{2}\Delta_z^k), \quad V^k = \Delta_x^k \Delta_y^k \Delta_z^k$$

ここで、 $J_{g,x}^k(x,y,z)$ は、体積 $V^k$ のk番目ノード内(x,y,z)点におけるx方向正味中性子流であり、Fickの法則とBenoistの方向依存拡散係数<sup>12)</sup>により以下のように表される(y, z方向についても同様である)。

$$J_{g,x}^k(x,y,z) = -D_{g,x}^k \frac{\partial}{\partial x} \phi_g^k(x,y,z) \quad (2.1.4)$$

$\Sigma_{r,g}^k$ は、k番目ノードにおける吸収と散乱(out-scattering)項から成る除去断面積である。

$$\Sigma_{r,g}^k \equiv \Sigma_{a,g}^k + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{s,g \rightarrow g'}^k \quad (2.1.5)$$

中性子源の項は、核分裂源と散乱(in-scattering)源から成り、固有値問題の場合には以下の式で表される。

$$Q_g^k(x,y,z) \equiv \frac{\chi_g^k}{k_{\text{eff}}} \sum_{g'=1}^G v \Sigma_{f,g}^k \phi_g^k(x,y,z) + \sum_{\substack{g'=1 \\ g' \neq g}} \Sigma_{s,g' \rightarrow g}^k \phi_{g'}^k(x,y,z) \quad (2.1.6)$$

$k_{\text{eff}}$ は固有値(実効中性子増倍率)である。(2.1.3)式の両辺をノード内で体積積分しノード体積で割ると、以下のノードバランス式を得る。

$$\frac{1}{\Delta_x^k} (J_{g,x+}^k - J_{g,x-}^k) + \frac{1}{\Delta_y^k} (J_{g,y+}^k - J_{g,y-}^k) + \frac{1}{\Delta_z^k} (J_{g,z+}^k - J_{g,z-}^k) + \Sigma_{r,g}^k \phi_{g,0}^k = Q_{g,0}^k \quad (2.1.7)$$

$x+$ はノードのx軸正方向の境界表面、 $x-$ はx軸負方向の境界表面を表しており、(2.1.7)式中の各パラメータは、以下のように定義している。

$$\phi_{g,0}^k \equiv \frac{1}{\Delta_x^k \Delta_y^k \Delta_z^k} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} \phi_g^k(x,y,z) dx dy dz \quad (2.1.8)$$

$$\begin{aligned} J_{g,x\pm}^k &\equiv \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} J_g^k(x = \pm \frac{\Delta_x^k}{2}, y, z) dy dz \\ &= \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \left\{ -D_{g,x}^k \frac{\partial}{\partial x} \phi_g^k(x, y, z) \Big|_{x=\pm \frac{\Delta_x^k}{2}} \right\} dy dz \end{aligned} \quad (2.1.9a)$$

$$\begin{aligned} J_{g,y\pm}^k &\equiv \frac{1}{\Delta_z^k \Delta_x^k} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} J_g^k(x, y = \pm \frac{\Delta_y^k}{2}, z) dz dx \\ &= \frac{1}{\Delta_z^k \Delta_x^k} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \left\{ -D_{g,y}^k \frac{\partial}{\partial y} \phi_g^k(x, y, z) \Big|_{y=\pm \frac{\Delta_y^k}{2}} \right\} dz dx \end{aligned} \quad (2.1.9b)$$

$$\begin{aligned} J_{g,z\pm}^k &\equiv \frac{1}{\Delta_x^k \Delta_y^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} J_g^k(x, y, z = \pm \frac{\Delta_z^k}{2}) dx dy \\ &= \frac{1}{\Delta_x^k \Delta_y^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} \left\{ -D_{g,z}^k \frac{\partial}{\partial z} \phi_g^k(x, y, z) \Big|_{z=\pm \frac{\Delta_z^k}{2}} \right\} dx dy \end{aligned} \quad (2.1.9c)$$

$$\begin{aligned}
Q_{g,0}^k &\equiv \frac{1}{\Delta_x^k \Delta_y^k \Delta_z^k} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} Q_g^k(x,y,z) dx dy dz \\
&= \frac{1}{\Delta_x^k \Delta_y^k \Delta_z^k} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} \left\{ \frac{\chi_g^k}{k_{\text{eff}}} \sum_{g'=1}^G v \Sigma_{f,g}^k \phi_g^k(x,y,z) + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{s,g' \rightarrow g}^k \phi_g^k(x,y,z) \right\} dx dy dz \\
&= \frac{\chi_g^k}{k_{\text{eff}}} \sum_{g'=1}^G v \Sigma_{f,g}^k \phi_{g',0}^k + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{s,g' \rightarrow g}^k \phi_{g',0}^k
\end{aligned} \tag{2.1.10}$$

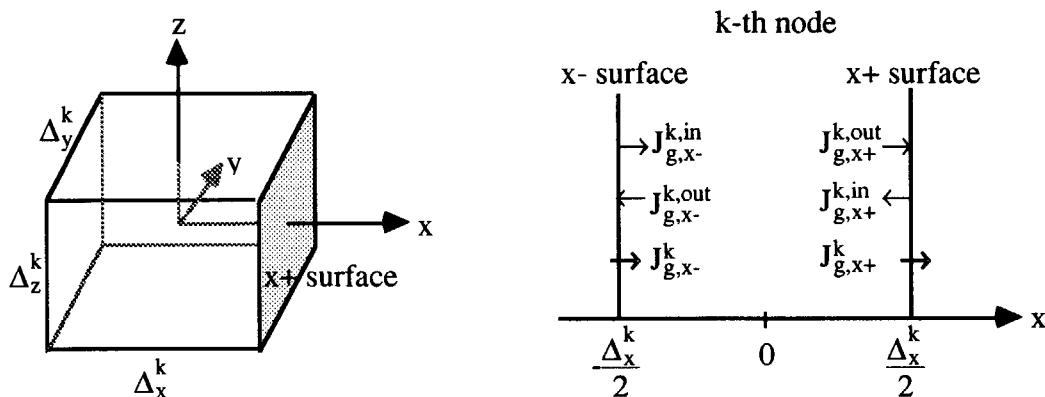
すなわち、ノードバランス式(2.1.7)は、1つのノード体積平均中性子束( $\phi_{g,0}^k$ )と、6つのノード境界表面に対する表面平均正味中性子流( $J_{g,x+}^k, J_{g,x-}^k, J_{g,y+}^k, J_{g,y-}^k, J_{g,z+}^k, J_{g,z-}^k$ )により記述される。

次に、表面平均中性子束と表面平均部分中性子流を導入する。拡散理論における部分中性子流は、角度中性子束にP1近似を適用することにより、一般に以下のように表すことができる<sup>13)</sup>。

$$J_g^{\text{out}}(\mathbf{r}) = \frac{\phi_g(\mathbf{r})}{4} - \frac{D_g}{2} \mathbf{e}_s \cdot \nabla \phi(\mathbf{r}) \tag{2.1.11a}$$

$$J_g^{\text{in}}(\mathbf{r}) = \frac{\phi_g(\mathbf{r})}{4} + \frac{D_g}{2} \mathbf{e}_s \cdot \nabla \phi(\mathbf{r}) \tag{2.1.11b}$$

$\mathbf{e}_s$ は着目する表面に垂直な単位法線ベクトルであり、法線ベクトルに沿う方向の部分中性子流(out-going partial current)を  $J^{\text{out}}$ 、逆方向の部分中性子流(in-comming partial current)を  $J^{\text{in}}$ としている。下図に示すように、 $k$ 番目ノードの右側境界面( $x+$ 面)に着目し、(2.1.11a)式の両辺を $x+$ 境界面上で積分して表面積で割ると(2.1.12)式を得る。



$$J_{g,x+}^{out,k} = \frac{1}{4} \phi_{g,x+}^k + \frac{1}{2} \left[ \frac{1}{\Delta_y \Delta_z} \int_{-\Delta_y/2}^{\Delta_y/2} \int_{-\Delta_z/2}^{\Delta_z/2} \left\{ -D_{g,x}^k \frac{\partial}{\partial x} \phi_g^k(x,y,z) \Big|_{x=\frac{\Delta_x}{2}} \right\} dy dz \right] \quad (2.1.12)$$

ただし、 $J_{g,x+}^{out,k}$ 、 $\phi_{g,x+}^k$ は、それぞれ(2.1.13)式、(2.1.14)式で定義されるx+境界面での外向き表面平均部分中性子流と表面平均中性子束である。

$$J_{g,x+}^{out,k} \equiv \frac{1}{\Delta_y \Delta_z} \int_{-\Delta_y/2}^{\Delta_y/2} \int_{-\Delta_z/2}^{\Delta_z/2} J_g^{k,out}(x=\frac{\Delta_x}{2},y,z) dy dz \quad (2.1.13)$$

$$\phi_{g,x+}^k \equiv \frac{1}{\Delta_y \Delta_z} \int_{-\Delta_y/2}^{\Delta_y/2} \int_{-\Delta_z/2}^{\Delta_z/2} \phi_g^k(x=\frac{\Delta_x}{2},y,z) dy dz \quad (2.1.14)$$

(2.1.12)式に(2.1.9a)式を代入すると、表面平均部分中性子流と表面平均正味中性子流、及び表面平均中性子束の関係が得られる。

$$J_{g,x+}^{out,k} = \frac{1}{4} \phi_{g,x+}^k + \frac{1}{2} J_{g,x+}^k \quad (2.1.15a)$$

同様にして(2.1.11b)式から(2.1.15b)式を得る。

$$J_{g,x+}^{in,k} = \frac{1}{4} \phi_{g,x+}^k - \frac{1}{2} J_{g,x+}^k \quad (2.1.15b)$$

(2.1.15a), (2.1.15b)式から

$$J_{g,x+}^k = J_{g,x+}^{out,k} - J_{g,x+}^{in,k} \quad (2.1.16)$$

$$\phi_{g,x+}^k = 2(J_{g,x+}^{out,k} + J_{g,x+}^{in,k}) \quad (2.1.17)$$

x-面についても同様にして以下の式を得る。

$$J_{g,x-}^k = J_{g,x-}^{in,k} - J_{g,x-}^{out,k} \quad (2.1.18)$$

$$\phi_{g,x-}^k = 2(J_{g,x-}^{out,k} + J_{g,x-}^{in,k}) \quad (2.1.19)$$

x軸方向表面に対して得られた(2.1.12)式～(2.1.19)式は、y軸及びz軸方向に対しても適用できる。(2.1.16)式及び(2.1.18)式に示されるように、表面平均正味中性子流は表面平均部分中性子流により表されるため、(2.1.7)式のノードバランス式は以下のように6つの面に対する表

面平均部分中性子流を用いて表すことができる。

$$\frac{1}{\Delta_x^k} (J_{g,x+}^{out,k} - J_{g,x+}^{in,k} - J_{g,x-}^{in,k} + J_{g,x-}^{out,k}) + \frac{1}{\Delta_y^k} (J_{g,y+}^{out,k} - J_{g,y+}^{in,k} - J_{g,y-}^{in,k} + J_{g,y-}^{out,k}) + \frac{1}{\Delta_z^k} (J_{g,z+}^{out,k} - J_{g,z+}^{in,k} - J_{g,z-}^{in,k} + J_{g,z-}^{out,k}) + \sum_{r,g} \phi_{g,0}^k = Q_{g,0}^k \quad (2.1.20)$$

これまでに得られたノード平均パラメータを以下に整理しておく。

- $\phi_{g,0}^k$  : ノードkに対する第g群の体積平均中性子束
- $Q_{g,0}^k$  : ノードkに対する第g群の体積平均中性子源
- $\phi_{g,x\pm}^k$  : ノードkに対する第g群のx±境界面における表面平均中性子束
- $\phi_{g,y\pm}^k$  : ノードkに対する第g群のy±境界面における表面平均中性子束
- $\phi_{g,z\pm}^k$  : ノードkに対する第g群のz±境界面における表面平均中性子束
- $J_{g,x\pm}^k$  : ノードkに対する第g群のx±境界面における表面平均正味中性子流
- $J_{g,y\pm}^k$  : ノードkに対する第g群のy±境界面における表面平均正味中性子流
- $J_{g,z\pm}^k$  : ノードkに対する第g群のz±境界面における表面平均正味中性子流
- $J_{g,x\pm}^{out,k}$  : ノードkに対する第g群のx±境界面においてノードの外側に向かう表面平均部分中性子流
- $J_{g,x\pm}^{in,k}$  : 同じくノードの内側に向かう表面平均部分中性子流
- $J_{g,y\pm}^{out,k}$  : ノードkに対する第g群のy±境界面においてノードの外側に向かう表面平均部分中性子流
- $J_{g,y\pm}^{in,k}$  : 同じくノードの内側に向かう表面平均部分中性子流
- $J_{g,z\pm}^{out,k}$  : ノードkに対する第g群のz±境界面においてノードの外側に向かう表面平均部分中性子流
- $J_{g,z\pm}^{in,k}$  : 同じくノードの内側に向かう表面平均部分中性子流

中性子源  $Q_{g,0}^k$  が既知であると仮定し、各境界面に対する計12個の表面平均部分中性子流が全て与えられるならば、バランス式(2.1.20)からノード体積平均中性子束が得られる。また、(2.1.16)式～(2.1.19)式から、6つの表面平均正味中性子流と6つの表面平均中性子束が得られる。しかし、実際の固有値問題や核分裂源を含む固定源問題では、(2.1.10)式に見られるように、 $Q_{g,0}^k$  に  $\phi_{g,0}^k$  が含まれる回帰的な問題となる。このため外部反復計算により  $\phi_{g,0}^k$  を決定し、この過程で固有値  $k_{eff}$  も決定される(第2.9節参照)。多項式展開拡散ノード法では、 $Q_{g,0}^k$  が暫定的に与えられる内部反復計算において、表面平均部分中性子流による体系全体のスウェーブ計算を行い、その収束値が求められる。以降では、主として12個の表面平均部分中性子流を決定する方法を論ずる。

## 2.2 トランスバース積分による1次元中性子バランス式

近代ノード法が粗メッシュを使用しても高い精度で解が得られるのは、ノード内の中性子束分布を、拡散方程式を近似的に満足させる多項式または解析的な関数で表現しているためである。この際、3次元の分布を直接精度良く表現することは容易ではない<sup>\*</sup>。このため、多くのノード法コードは、注目外の方向について面積分した1次元拡散方程式(transverse-integrated equation)を扱い、そこで現れる1次元の中性子束分布を1変数関数で表現する。以下では、x軸方向に着目した場合の1次元拡散方程式を導出し、それに関わるパラメータを定義する。

先ず、k番目のノードに対する3次元中性子バランス式である(2.1.3)式の両辺をy-z面について積分し、面積 $\Delta_y^k \Delta_z^k$ で割る。

$$\begin{aligned} & \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \left\{ \frac{\partial}{\partial x} J_{g,x}^k(x,y,z) \right\} dy dz + \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \left\{ \frac{\partial}{\partial y} J_{g,y}^k(x,y,z) \right\} dy dz \\ & + \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \left\{ \frac{\partial}{\partial z} J_{g,z}^k(x,y,z) \right\} dy dz + \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \Sigma_{r,g}^k \phi_g^k(x,y,z) dy dz \\ & = \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} Q_g^k(x,y,z) dy dz \end{aligned} \quad (2.2.1)$$

$$\text{左辺第1項} = \frac{d}{dx} \left\{ \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} J_{g,x}^k(x,y,z) dy dz \right\} \quad (2.2.2)$$

$$\begin{aligned} \text{左辺第2項} &= \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \frac{\partial}{\partial y} J_{g,y}^k(x,y,z) dy dz \\ &= \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \frac{\partial}{\partial y} \left\{ -D_{g,y}^k \frac{\partial \phi_g^k(x,y,z)}{\partial y} \right\} dy dz \quad (2.1.4\text{式参照}) \\ &= \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \left[ -D_{g,y}^k \frac{\partial \phi_g^k(x,y,z)}{\partial y} \right]_{-\Delta_z^k/2}^{\Delta_z^k/2} dz \end{aligned} \quad (2.2.3)$$

第3項については第2項と同様にして、

<sup>\*</sup>最近では、このような扱いをする手法も開発されている<sup>14),15)</sup>。

$$\text{左辺第3項} = \frac{1}{\Delta_z^k \Delta_y^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \left[ -D_{g,z}^k \frac{\partial \phi_g^k(x,y,z)}{\partial z} \right]_{-\Delta_z^k/2}^{\Delta_z^k/2} dy \quad (2.2.4)$$

$$\text{左辺第4項} = \frac{\sum_{r,g}^k}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \phi_g^k(x,y,z) dy dz \quad (2.2.5)$$

$$\begin{aligned} \text{右辺} &= \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} Q_g^k(x,y,z) dy dz \\ &= \frac{\chi_g^k}{k_{\text{eff}}} \sum_{g'=1}^G \frac{v \Sigma_{f,g'}^k}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \phi_{g'}^k(x,y,z) dy dz + \sum_{\substack{g'=1 \\ g' \neq g}}^G \frac{\Sigma_{s,g' \rightarrow g}^k}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \phi_g^k(x,y,z) dy dz \end{aligned} \quad (2.2.6)$$

ここで、 $k$ 番目ノード内の任意x方向位置でy-z面について積分された以下の諸量を定義する。

$$\bar{\phi}_{g,x}^k(x) \equiv \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \phi_g^k(x,y,z) dy dz \quad (2.2.7)$$

$$\bar{J}_{g,x}^k(x) \equiv \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} J_{g,x}^k(x,y,z) dy dz \quad (2.2.8)$$

$$\begin{aligned} \bar{Q}_{g,x}^k(x) &\equiv \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} Q_g^k(x,y,z) dy dz \\ &= \frac{\chi_g^k}{k_{\text{eff}}} \sum_{g'=1}^G v \Sigma_{f,g'}^k \bar{\phi}_{g'}^k(x) + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{s,g' \rightarrow g}^k \bar{\phi}_{g'}^k(x) \quad (2.2.6\text{式参照}) \end{aligned} \quad (2.2.9)$$

$$\bar{L}_{g,y}^k(x) \equiv \frac{1}{\Delta_z^k} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \left[ -D_{g,y}^k \frac{\partial \phi_g^k(x,y,z)}{\partial y} \right]_{-\Delta_y^k/2}^{\Delta_y^k/2} dz \quad (2.2.10)$$

$$\bar{L}_{g,z}^k(x) \equiv \frac{1}{\Delta_y^k} \int_{-\Delta_y^k/2}^{\Delta_y^k/2} \left[ -D_{g,z}^k \frac{\partial \phi_g^k(x,y,z)}{\partial z} \right]_{-\Delta_z^k/2}^{\Delta_z^k/2} dy \quad (2.2.11)$$

これらのパラメータを使用して(2.2.1)式を書き表すと以下のようになる。

$$\frac{d}{dx} \bar{J}_{g,x}^k(x) + \Sigma_{r,g}^k \bar{\phi}_{g,x}^k(x) = \bar{Q}_{g,x}^k(x) - \frac{1}{\Delta_y^k} \bar{L}_{g,y}^k(x) - \frac{1}{\Delta_z^k} \bar{L}_{g,z}^k(x) \quad (2.2.12a)$$

y方向とz方向についても同様に、

$$\frac{d}{dy} \bar{J}_{g,y}^k(y) + \Sigma_{r,g}^k \bar{\phi}_{g,y}^k(y) = \bar{Q}_{g,y}^k(y) - \frac{1}{\Delta_z^k} \bar{L}_{g,z}^k(y) - \frac{1}{\Delta_x^k} \bar{L}_{g,x}^k(y) \quad (2.2.12b)$$

$$\frac{d}{dz} \bar{J}_{g,z}^k(z) + \Sigma_{r,g}^k \bar{\phi}_{g,z}^k(z) = \bar{Q}_{g,z}^k(z) - \frac{1}{\Delta_x^k} \bar{L}_{g,x}^k(z) - \frac{1}{\Delta_y^k} \bar{L}_{g,y}^k(z) \quad (2.2.12c)$$

(2.1.12)式は、k番目のノードに対するトランスマース積分された中性子バランス式(transverse-integrated equations)である。通常の1次元問題に対し、中性子源項に着目する方向外への中性子漏洩項を加えた式として表現される。

次に3次元式であるFickの法則についても、同様にトランスマース積分を行う。即ち(2.1.4)式の両辺をy-z面について積分し面積 $\Delta_y^k \Delta_z^k$ で割る。

$$\frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^{k/2}}^{\Delta_y^{k/2}} \int_{-\Delta_z^{k/2}}^{\Delta_z^{k/2}} J_{g,x}^k(x,y,z) dy dz = \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^{k/2}}^{\Delta_y^{k/2}} \int_{-\Delta_z^{k/2}}^{\Delta_z^{k/2}} \left\{ -D_{g,x}^k \frac{\partial \phi_{g,x}^k(x,y,z)}{\partial x} \right\} dy dz \quad (2.2.13)$$

(2.2.7)式と(2.2.8)式の定義より、

$$\begin{aligned} \bar{J}_{g,x}^k(x) &= -D_{g,x}^k \frac{d}{dx} \left\{ \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^{k/2}}^{\Delta_y^{k/2}} \int_{-\Delta_z^{k/2}}^{\Delta_z^{k/2}} \phi_{g,x}^k(x,y,z) dy dz \right\} \\ &= -D_{g,x}^k \frac{d}{dx} \bar{\phi}_{g,x}^k(x) \end{aligned} \quad (2.2.14a)$$

y方向とz方向についても同様に、

$$\bar{J}_{g,y}^k(y) = -D_{g,y}^k \frac{d}{dy} \bar{\phi}_{g,y}^k(y) \quad (2.2.14b)$$

$$\bar{J}_{g,z}^k(z) = -D_{g,z}^k \frac{d}{dz} \bar{\phi}_{g,z}^k(z) \quad (2.2.14c)$$

これらを(2.2.12)式の各方向の式に代入すれば、1次元拡散方程式の形となる。例えば、x方向については、以下の式で表される。

$$-D_{g,x}^k \frac{d^2}{dx^2} \bar{\phi}_{g,x}^k(x) + \Sigma_{r,g}^k \bar{\phi}_{g,x}^k(x) = \bar{Q}_{g,x}^k(x) - \frac{1}{\Delta_y^k} \bar{L}_{g,y}^k(x) - \frac{1}{\Delta_z^k} \bar{L}_{g,z}^k(x) \quad (2.2.15)$$

自明ではあるが、y-z面で積分されたx方向1次元中性子バランス式(2.2.12a)を、更に( $-\frac{1}{2}\Delta_x^k \leq x \leq \frac{1}{2}\Delta_x^k$ )の範囲で積分すれば、前節で得られたノード平均パラメータによる中性子バランス式(2.1.7)が得られることを示しておく。

$$\begin{aligned} & \frac{1}{\Delta_x^k} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} \frac{d}{dx} \bar{J}_{g,x}^k(x) dx + \frac{1}{\Delta_x^k} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} \Sigma_{r,g}^k \bar{\phi}_{g,x}^k(x) dx \\ &= \frac{1}{\Delta_x^k} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} \bar{Q}_{g,x}^k(x) dx - \frac{1}{\Delta_x^k \Delta_y^k} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} \bar{L}_{g,y}^k(x) dx - \frac{1}{\Delta_x^k \Delta_z^k} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} \bar{L}_{g,z}^k(x) dx \end{aligned} \quad (2.2.16)$$

$$\begin{aligned} \text{左辺第1項} &= \frac{1}{\Delta_x^k} [\bar{J}_{g,x}^k(x)]_{-\Delta_x^k/2}^{\Delta_x^k/2} \\ &= \frac{1}{\Delta_x^k} \left\{ \bar{J}_{g,x}^k(x = \frac{\Delta_x^k}{2}) - \bar{J}_{g,x}^k(x = -\frac{\Delta_x^k}{2}) \right\} \\ &= \frac{1}{\Delta_x^k} \left\{ J_{g,x+}^k - J_{g,x-}^k \right\} \quad (2.1.9a, 2.2.8式参照) \end{aligned} \quad (2.2.17)$$

(2.1.8)式と(2.2.7)式より、

$$\frac{1}{\Delta_x^k} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} \bar{\phi}_{g,x}^k(x) dx = \phi_{g,0}^k \quad (2.2.18)$$

よって、

$$\text{左辺第2項} = \Sigma_{r,g}^k \phi_{g,0}^k \quad (2.2.19)$$

また、(2.1.10)式、(2.2.9)、(2.2.18)式から

$$\begin{aligned} \text{右辺第1項} &= \frac{1}{\Delta_x^k} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} \bar{Q}_{g,x}^k(x) dx \\ &= \frac{\chi_g^k}{k_{eff}} \sum_{g'=1}^G v \Sigma_{f,g'}^k \phi_{g',0}^k + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{s,g' \rightarrow g}^k \phi_{g,0}^k \\ &= Q_{g,0}^k \end{aligned} \quad (2.2.20)$$

ここで、後の便宜のため以下の諸量を定義しておく。

$$\begin{aligned}
 L_{g,y}^k &\equiv \frac{1}{\Delta_x^k} \int_{-\Delta_y/2}^{\Delta_y/2} \bar{L}_{g,y}^k(x) dx \\
 &= \frac{1}{\Delta_x^k \Delta_z^k} \int_{-\Delta_y/2}^{\Delta_y/2} \int_{-\Delta_z/2}^{\Delta_z/2} \left[ -D_{g,y}^k \frac{\partial \phi_g^k(x,y,z)}{\partial y} \right]_{-\Delta_z/2}^{\Delta_z/2} dz dx \quad (2.2.10\text{式参照}) \\
 &= J_{g,y+}^k - J_{g,y-}^k \quad (2.1.9\text{b式参照})
 \end{aligned} \tag{2.2.21a}$$

$$\begin{aligned}
 L_{g,z}^k &\equiv \frac{1}{\Delta_x^k} \int_{-\Delta_y/2}^{\Delta_y/2} \bar{L}_{g,z}^k(x) dx \\
 &= \frac{1}{\Delta_x^k \Delta_y^k} \int_{-\Delta_y/2}^{\Delta_y/2} \int_{-\Delta_z/2}^{\Delta_z/2} \left[ -D_{g,z}^k \frac{\partial \phi_g^k(x,y,z)}{\partial z} \right]_{-\Delta_y/2}^{\Delta_y/2} dy dx \quad (2.2.11\text{式参照}) \\
 &= J_{g,z+}^k - J_{g,z-}^k \quad (2.1.9\text{c式参照})
 \end{aligned} \tag{2.2.21b}$$

同様にして、

$$\begin{aligned}
 L_{g,x}^k &\equiv \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y/2}^{\Delta_y/2} \int_{-\Delta_z/2}^{\Delta_z/2} \left[ -D_{g,x}^k \frac{\partial \phi_g^k(x,y,z)}{\partial x} \right]_{-\Delta_y/2}^{\Delta_y/2} dy dz \\
 &= J_{g,x+}^k - J_{g,x-}^k \quad (2.1.9\text{a式参照})
 \end{aligned} \tag{2.2.21c}$$

これらのパラメータを使用すると、(2.2.16)式において、

$$\text{右辺第2項及び第3項} = -\frac{1}{\Delta_y^k} L_{g,y}^k - \frac{1}{\Delta_z^k} L_{g,z}^k \tag{2.2.22}$$

(2.2.17), (2.2.19), (2.2.20), (2.2.22)式より、(2.2.16)式は以下のように表わされる。

$$\frac{1}{\Delta_x^k} \left( J_{g,x+}^k - J_{g,x-}^k \right) + \sum_{r,g}^k \phi_{g,0}^k = Q_{g,0}^k - \frac{1}{\Delta_y^k} L_{g,y}^k - \frac{1}{\Delta_z^k} L_{g,z}^k \tag{2.2.23}$$

これに、(2.2.21a)式と(2.2.21b)式を代入すれば、前節で得られた表面平均正味中性子流によるノードバランス式(2.1.7)が得られる。

本節でx方向に着目して得られた、k番目ノードに対する主要なパラメータの物理的意味を以下に整理しておく。

$\bar{\phi}_{g,x}^k(x)$  : ノードk内のy-z面で面平均された第g群中性子束のx方向分布 (2.2.7式参照)

(2.1.14)式と(2.2.7)式の定義より、 $x=\pm\frac{\Delta_x^k}{2}$ において $\phi_{g,x,\pm}^k$ に一致する。

$\bar{Q}_{g,x}^k(x)$  : ノードk内のy-z面で平均された第g群中性子源のx方向分布 (2.2.9式参照)

$\bar{J}_{g,x}^k(x)$  : ノードk内のy-z面で平均された第g群正味中性子流のx方向分布 (2.2.8式参照)

(2.1.9a)式と(2.2.8)式の定義より、 $x=\pm\frac{\Delta_x^k}{2}$ において $J_{g,x,\pm}^k$ に一致する。

$\bar{L}_{g,y}^k(x)$  : ノードk内のz方向について平均されたy方向からの第g群正味中性子漏洩のx方向分布 (2.2.10式参照)

$\bar{L}_{g,z}^k(x)$  : ノードk内のy方向について平均されたz方向からの第g群正味中性子漏洩のx方向分布 (2.2.11式参照)

y方向およびz方向についても、同様の1次元分布パラメータが定義される。その他に、本節で新たに定義したパラメータとして以下のものがある。

$L_{g,y}^k$  : ノードkのy方向両境界面(y±面)からの第g群正味中性子漏洩 (2.2.21a式)

$L_{g,z}^k$  : ノードkのz方向両境界面(z±面)からの第g群正味中性子漏洩 (2.2.21b式)

$L_{g,x}^k$  : ノードkのx方向両境界面(x±面)からの第g群正味中性子漏洩 (2.2.21c式)

## 2.3 4次の多項式展開近似

ここまで議論では、拡散近似以外の本質的な近似は取り込んでいない。3次元拡散方程式から3方向に対する1次元トランスマース方程式を導く際にも、ノード内中性子束分布の変数分離を仮定しているわけではない。本節において、多項式展開拡散ノード法で最も本質的とも言える仮定を導入する。それは、各方向に対するノード内の1次元中性子束分布を多項式で近似するものである。例えば、 $x$ 方向に着目して(2.2.7)式で定義される $\phi_{g,x}^k(x)$ を以下のようにN次の多項式で近似する。

$$\phi_{g,x}^k(x) \approx \sum_{n=0}^N a_{g,x,n}^k f_n(x) \quad (2.3.1)$$

$f_n(x)$ は多項式による展開関数(basis function)であり、 $a_{g,x,n}^k$ はその係数である。ただし、(2.3.1)式は常に(2.2.18)式を満足しなければならない。

$$\begin{aligned} \phi_{g,0}^k &= \frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} \phi_{g,x}^k(x) dx \\ &\approx \frac{a_{g,x,0}^k}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} f_0(x) dx + \sum_{n=1}^N \frac{a_{g,x,n}^k}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} f_n(x) dx \end{aligned} \quad (2.3.2)$$

$a_{g,x,0}^k = \phi_{g,0}^k$ として、以下の式を満足する展開関数を選択する。

$$\frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} f_n(x) dx = \begin{cases} 1 & (n=0) \\ 0 & (n=1, 2, \dots, N) \end{cases} \quad (2.3.3)$$

展開次数Nや関数形 $f_n(x)$ の選択、さらに後に述べる重み付き残差法における重み関数の採り方により、多項式展開ノード法(NEM)にいくつかのバリエーションを与えることになる。それらを最適化することが計算を効率的に行う上で重要となる。一般に展開次数を小さくすると、粗メッシュを使用したときの計算精度が低下し、十分な精度を期待するにはノードのサイズを小さくする必要がある。その結果、計算コスト(計算時間や必要メモリー量)が多くなる。一方、次数を大きくすると、粗いメッシュでも高い計算精度が得られるようになるが、高次の展開係数を決定するために必要となる余分な方程式(次節の高次モーメントバランス式)の数が増加し、同じメッシュサイズで低次の展開を用いた場合に比べ計算コストが高くなる。NEMが高速な計算手法として効果があるのは、適切な展開次数を選択すれば、メッシュ数を増やすことによる計算負荷の増大に比べ、少ないメッシュ数を使用して方程式の数を増やす方が、計算負荷がはるかに小さくなるためである。後章で示すように、

典型的な軽水炉体系では、N=4とすることによりノード幅を10cm~20cmとしても十分な計算精度が得られる。以下では、Lawrenceのレビュー論文に倣い、N=4として(2.3.4)式に示す展開関数を用いる。ただし、ここでは原点をノード中心にとり、注目方向のノード幅に対する相対座標(- $\frac{1}{2} \leq \xi \leq \frac{1}{2}$ )の関数として表示する。

$$f_0(x) \equiv 1 \quad (2.3.4a)$$

$$f_1(x) \equiv \frac{x}{\Delta_x^k} \equiv \xi \quad (-\frac{1}{2} \leq \xi \leq \frac{1}{2}) \quad (2.3.4b)$$

$$f_2(x) \equiv 3\xi^2 - \frac{1}{4} \quad (2.3.4c)$$

$$f_3(x) \equiv \xi(\xi - \frac{1}{2})(\xi + \frac{1}{2}) \quad (2.3.4d)$$

$$f_4(x) \equiv (\xi^2 - \frac{1}{20})(\xi - \frac{1}{2})(\xi + \frac{1}{2}) \quad (2.3.4e)$$

(2.3.4)式において、原点を負側のノード境界に設定し振幅を変更すれば、(2.3.5)式を得る。

$$h_0(x) \equiv f_0(\xi) = 1 \quad (2.3.5a)$$

$$h_1(x) \equiv 2f_1(\xi) = 2u-1 \quad (2.3.5b)$$

$$h_2(x) \equiv -2f_2(\xi) = 6u(1-u)-1 \quad (2.3.5c)$$

$$h_3(x) \equiv -12f_3(\xi) = 6u(1-u)(2u-1) \quad (2.3.5d)$$

$$h_4(x) \equiv -30f_4(\xi) = 6u(1-u)(5u^2-5u+1) \quad (2.3.5e)$$

$$\text{ただし、 } u \equiv \xi + \frac{1}{2}, \quad (0 \leq u \leq 1)$$

展開関数として、(2.3.5)式の表現をとる論文をしばしば見かけるが、これは、NEMの基礎を確立したFinnenmann等の論文<sup>6)</sup>に倣ったもので、係数の値が変わるだけで(2.3.4)式と同等である。

(2.3.4)式で定義する展開関数は、一見すると直交関数に見受けられるがそうではない。以下のようないくつかの関係式が成立している。

$$\text{記号 } \langle \quad \rangle \equiv \frac{1}{\Delta_x^k} \int_{-\frac{1}{2}}^{\frac{1}{2}} dx = \int_{-\frac{1}{2}}^{\frac{1}{2}} d\xi \quad \text{とすると、}$$

$$\langle f_0(x)f_0(x) \rangle = \langle f_0(x) \rangle = 1 \quad (2.3.6a)$$

$$\langle f_0(x)f_1(x) \rangle = \langle f_1(x) \rangle = 0 \quad (2.3.6b)$$

$$\langle f_0(x)f_2(x) \rangle = \langle f_2(x) \rangle = 0 \quad (2.3.6c)$$

$$\langle f_0(x)f_3(x) \rangle = \langle f_3(x) \rangle = 0 \quad (2.3.6d)$$

$$\langle f_0(x)f_4(x) \rangle = \langle f_4(x) \rangle = 0 \quad (2.3.6e)$$

$$\langle f_1(x)f_1(x) \rangle = \frac{1}{12} \quad (2.3.7a)$$

$$\langle f_1(x)f_2(x) \rangle = 0 \quad (2.3.7b)$$

$$\langle f_1(x)f_3(x) \rangle = \frac{-1}{120} \quad (2.3.7c)$$

$$\langle f_1(x)f_4(x) \rangle = 0 \quad (2.3.7d)$$

$$\langle f_2(x)f_2(x) \rangle = \frac{1}{20} \quad (2.3.8a)$$

$$\langle f_2(x)f_3(x) \rangle = 0 \quad (2.3.8b)$$

$$\langle f_2(x)f_4(x) \rangle = \frac{-1}{700} \quad (2.3.8c)$$

(2.3.1)式でN=4とし、これに(2.3.4)式を代入すれば、

$$\begin{aligned} \bar{\phi}_{g,x}^k(x) &\approx a_{g,x,0}^k + \sum_{n=1}^4 a_{g,x,n}^k f_n(x) \\ &\approx \phi_{g,0}^k + a_{g,x,1}^k \xi + a_{g,x,2}^k \left(3\xi^2 - \frac{1}{4}\right) + a_{g,x,3}^k \left\{ \xi \left(\xi - \frac{1}{2}\right) \left(\xi + \frac{1}{2}\right) \right\} + a_{g,x,4}^k \left\{ \left(\xi^2 - \frac{1}{20}\right) \left(\xi - \frac{1}{2}\right) \left(\xi + \frac{1}{2}\right) \right\} \\ \text{ただし、 } \frac{-1}{2} \leq \left| \xi \equiv \frac{x}{\Delta_x^k} \right| &\leq \frac{1}{2} \end{aligned} \quad (2.3.9)$$

$a_{g,x,0}^k$ はノード体積平均中性子束と一致するように決められたが、 $a_{g,x,1}^k$ と $a_{g,x,2}^k$ は、x方向の境界表面平均中性子束から決めることできる。すなわち(2.1.14)式と(2.2.7)式から、

$$\bar{\phi}_{g,x}^k(x = \pm \frac{\Delta_x^k}{2}) = \phi_{g,x\pm}^k$$

これを、(2.3.9)式に代入すれば、

$$\bar{\phi}_{g,x}^k(x = \frac{\Delta_x^k}{2}) = \phi_{g,0}^k + \frac{1}{2} a_{g,x,1}^k + \frac{1}{2} a_{g,x,2}^k = \phi_{g,x+}^k \quad (2.3.10a)$$

$$\bar{\phi}_{g,x}^k(x = -\frac{\Delta_x^k}{2}) = \phi_{g,0}^k - \frac{1}{2} a_{g,x,1}^k - \frac{1}{2} a_{g,x,2}^k = \phi_{g,x-}^k \quad (2.3.10b)$$

これらより、

$$a_{g,x,1}^k = \phi_{g,x+}^k - \phi_{g,x-}^k \quad (2.3.11a)$$

$$a_{g,x,2}^k = \phi_{g,x+}^k + \phi_{g,x-}^k - 2\phi_{g,0}^k \quad (2.3.11b)$$

ちなみに、(2.3.11)式に3次以上の展開係数が入ってこないのは、むしろそのように展開関数

を決めているためである。すなわち、以下のような条件を展開関数に課している。

$$f_n(x = \pm \frac{\Delta_x}{2}) = 0, \quad (n=3,4,\dots,N) \quad (2.3.12)$$

残された展開係数  $a_{g,x,3}^k$  と  $a_{g,x,4}^k$  を  $a_{g,x,0}^k$ 、 $a_{g,x,1}^k$ 、 $a_{g,x,2}^k$  と同様にノード平均パラメータで表現したいところであるが、これについては次節に譲り、NEMが体系スウェーブに使用する表面平均部分中性子流と展開係数との関係を得ておく。

先ず  $x+$  境界面に着目すると、(2.1.16)式と(2.2.14a)式より、

$$\begin{aligned} J_{g,x+}^{out,k} &= J_{g,x+}^k + J_{g,x+}^{in,k} \\ &= -D_{g,x}^k \frac{d\phi_{g,x}(x)}{dx} \Big|_{x=\frac{\Delta_x}{2}} + J_{g,x+}^{in,k} \end{aligned} \quad (2.3.13)$$

これに(2.3.9)式を代入すれば、以下の式を得る。

$$J_{g,x+}^{out,k} \approx -\frac{D_{g,x}^k}{\Delta_x} \left\{ a_{g,x,1}^k + 3a_{g,x,2}^k + \frac{1}{2}a_{g,x,3}^k + \frac{1}{5}a_{g,x,4}^k \right\} + J_{g,x+}^{in,k} \quad (2.3.14a)$$

$x-$  境界面についても同様に、(2.1.18)式と(2.2.14a)式から、

$$J_{g,x-}^{out,k} \approx \frac{D_{g,x}^k}{\Delta_x} \left\{ a_{g,x,1}^k - 3a_{g,x,2}^k + \frac{1}{2}a_{g,x,3}^k - \frac{1}{5}a_{g,x,4}^k \right\} + J_{g,x-}^{in,k} \quad (2.3.14b)$$

さらに、(2.3.11a)式と(2.1.17)及び(2.1.19)式より、

$$\begin{aligned} a_{g,x,1}^k &= \phi_{g,x+}^k - \phi_{g,x-}^k \\ &= 2(J_{g,x+}^{out,k} + J_{g,x+}^{in,k}) - 2(J_{g,x-}^{out,k} + J_{g,x-}^{in,k}) \end{aligned} \quad (2.3.15)$$

また、(2.3.11b)式より、

$$\begin{aligned} a_{g,x,2}^k &= \phi_{g,x+}^k + \phi_{g,x-}^k - 2\phi_{g,0}^k \\ &= 2(J_{g,x+}^{out,k} + J_{g,x+}^{in,k}) + 2(J_{g,x-}^{out,k} + J_{g,x-}^{in,k}) - 2\phi_{g,0}^k \end{aligned} \quad (2.3.16)$$

これらを(2.3.14a)及び(2.3.14b)式に代入することにより、それぞれ(2.3.17a)、(2.3.17b)式を得る。

$$J_{g,x+}^{out,k} \approx -\frac{D_{g,x}^k}{\Delta_x^k} \left\{ 8J_{g,x+}^{out,k} + 8J_{g,x+}^{in,k} + 4J_{g,x-}^{out,k} + 4J_{g,x-}^{in,k} - 6\phi_{g,0}^k + \frac{1}{2}a_{g,x,3}^k + \frac{1}{5}a_{g,x,4}^k \right\} + J_{g,x+}^{in,k} \quad (2.3.17a)$$

$$J_{g,x-}^{out,k} \approx \frac{D_{g,x}^k}{\Delta_x^k} \left\{ -4J_{g,x+}^{out,k} - 4J_{g,x+}^{in,k} - 8J_{g,x-}^{out,k} - 8J_{g,x-}^{in,k} + 6\phi_{g,0}^k + \frac{1}{2}a_{g,x,3}^k - \frac{1}{5}a_{g,x,4}^k \right\} + J_{g,x-}^{in,k} \quad (2.3.17b)$$

なお、x方向に着目して本節で得られた関係式は、y方向とz方向についての1次元表面平均中性子束分布を以下のように展開すれば、同様の関係式が得られる。

$$\bar{\phi}_{g,y}^k(y) \approx \phi_{g,0}^k + \sum_{n=1}^4 a_{g,y,n}^k f_n(y) \quad (2.3.18a)$$

$$\bar{\phi}_{g,z}^k(z) \approx \phi_{g,0}^k + \sum_{n=1}^4 a_{g,z,n}^k f_n(z) \quad (2.3.18b)$$

## 2.4 重み付き残差法

### 2.4.1 1次元問題における2次の多項式展開

しばらくの間、理解を促すためにx方向1次元問題で展開次数が2の場合を考察してみる。  
(2.3.1)式においてN=2とし、k番目ノード内の中性子束分布を展開する。

$$\bar{\phi}_{g,x}^k(x) \approx \phi_{g,0}^k + a_{g,x,1}^k \xi + a_{g,x,2}^k (3\xi^2 - \frac{1}{4}) \quad (2.4.1)$$

$a_{g,x,1}^k$ と $a_{g,x,2}^k$ はノード境界中性子束の拘束条件から次のように与えられる(2.3.15式及び2.3.16式参照)。

$$a_{g,x,1}^k = 2(J_{g,x+}^{out,k} + J_{g,x+}^{in,k}) - 2(J_{g,x-}^{out,k} + J_{g,x-}^{in,k}) \quad (2.4.2a)$$

$$a_{g,x,2}^k = 2(J_{g,x+}^{out,k} + J_{g,x+}^{in,k}) + 2(J_{g,x-}^{out,k} + J_{g,x-}^{in,k}) - 2\phi_{g,0}^k \quad (2.4.2b)$$

また、前節で得られた(2.3.17)式において、 $a_{g,x,3}^k$ と $a_{g,x,4}^k$ をゼロとすれば、

$$J_{g,x+}^{out,k} \approx -\frac{D_{g,x}^k}{\Delta_x} \left\{ 8J_{g,x+}^{out,k} + 8J_{g,x+}^{in,k} + 4J_{g,x-}^{out,k} + 4J_{g,x-}^{in,k} - 6\phi_{g,0}^k \right\} + J_{g,x+}^{in,k} \quad (2.4.3)$$

$$J_{g,x-}^{out,k} \approx -\frac{D_{g,x}^k}{\Delta_x} \left\{ 4J_{g,x+}^{out,k} + 4J_{g,x+}^{in,k} + 8J_{g,x-}^{out,k} + 8J_{g,x-}^{in,k} - 6\phi_{g,0}^k \right\} + J_{g,x-}^{in,k} \quad (2.4.4)$$

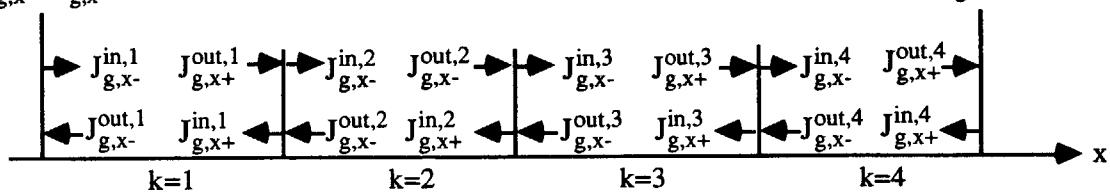
さらに、1次元問題におけるノードバランス式は(2.1.20)式より、

$$\frac{1}{\Delta_x} \left( J_{g,x+}^{out,k} - J_{g,x+}^{in,k} + J_{g,x-}^{out,k} - J_{g,x-}^{in,k} \right) + \sum_{r,g}^k \phi_{g,0}^k = Q_{g,0}^k \quad (2.4.5)$$

ノード総数を例えれば4とすると、 $k=1, 2, 3, 4$ の各ノードに対し、(2.4.3)式～(2.3.5)式が成立し、12個の方程式が与えられる。

Reflective B.C.

$$J_{g,x-}^{in,1} = J_{g,x-}^{out,1}$$



Vacuum B.C.

$$J_{g,x+}^{in,4} = 0$$

一方、 $Q_{g,0}^k$ が既知である固定源問題を仮定すると未知数は以下のとおりである。

- 1番目ノードに対し、 $J_{g,x-}^{out,1}, J_{g,x-}^{in,1}, J_{g,x+}^{out,1}, J_{g,x+}^{in,1}, \phi_{g,0}^1$
- 2番目ノードに対し、 $(J_{g,x-}^{out,2}=J_{g,x+}^{in,1}), (J_{g,x-}^{in,2}=J_{g,x+}^{out,1}), J_{g,x+}^{out,2}, J_{g,x+}^{in,2}, \phi_{g,0}^2$
- 3番目ノードに対し、 $(J_{g,x-}^{out,3}=J_{g,x+}^{in,2}), (J_{g,x-}^{in,3}=J_{g,x+}^{out,2}), J_{g,x+}^{out,3}, J_{g,x+}^{in,3}, \phi_{g,0}^3$
- 4番目ノードに対し、 $(J_{g,x-}^{out,4}=J_{g,x+}^{in,3}), (J_{g,x-}^{in,4}=J_{g,x+}^{out,3}), J_{g,x+}^{out,4}, J_{g,x+}^{in,4}, \phi_{g,0}^4$

( )内は部分中性子流のノード境界における連続条件である。結局、未知数の数は14個となるが、方程式の不足分は両端の境界条件により与えられる。例えば、左側境界を反射境界条件、右側境界を真空境界条件とすれば、 $J_{g,x-}^{in,1}=J_{g,x-}^{out,1}$ と $J_{g,x+}^{in,4}=0$ の2つの式が与えられ、未知数と方程式の数が過不足なくなる。これにより、体系全体のノード平均中性子束と部分中性子流が求められる。また、各ノード内の中性子束分布は、(2.4.1)式と(2.4.2)式により2次式として得られる。

$Q_{g,0}^k$ を既知としたが、固有値問題の場合には、 $Q_{g,0}^k$ の初期値を与えて上述のようにして得られるノード平均中性子束を(2.1.10)式に代入し、 $Q_{g,0}^k$ を更新して外部反復計算を行えば良い。3次元問題の場合にも、同様に計算することができる(第2.5.2節参照)。

ノード境界における部分中性子流の連続は、(2.1.16)式と(2.1.17)式から、ノード境界における中性子束と正味中性子流の連続条件と等価である。すなわち、N=2の低次の多項式展開ノード法では、ノード境界での中性子束と正味中性子流の連続性を保証しながら、体系全体で中性子バランス式(2.4.5式)を満足させれば、方程式と未知数の数が一致し拡散方程式の解を得ることができる。これは、扱う変数が部分中性子流であることを除けば、通常の有限差分法とほとんど同じプロセスである。また、ノード内の中性子束分布は2次式で与えられるが、ノード内での拡散方程式をなんら満足することを保証していない。結局、2次の多項式展開で正確な解を得るには、有限差分法と同様にノード幅を十分に小さくする必要がある。

#### 2.4.2 重み付き残差法の導入 (4次の多項式展開)

再び3次元問題に対する4次の多項式展開に話を戻す。とりあえずx方向に着目して、高次の展開係数 $a_{g,x,3}^k$ と $a_{g,x,4}^k$ をノード平均パラメータと関係付けたいが、N=2の場合に既に方程式と未知数の数が一致しており、何らかの新しい条件式が必要になる。そこで、ノード内の1次元中性子束分布 $\phi_{g,x}(x)$ を近似的に拡散方程式に従わせようとの観点から、重み付き残差法(weighted residual method)を導入する。(2.2.12a)式に重み関数 $w_n(x)$ を乗じてノード内で積分し、さらに $\Delta_x^k$ で割る。

$$\frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} w_n(x) \left\{ \frac{d}{dx} \bar{J}_{g,x}^k(x) + \sum_{r,g}^k \bar{\phi}_{g,x}^k(x) \right\} dx = \frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} w_n(x) \left\{ \bar{Q}_{g,x}^k(x) - \frac{1}{\Delta_y^k} \bar{L}_{g,y}^k(x) - \frac{1}{\Delta_z^k} \bar{L}_{g,z}^k(x) \right\} dx \quad (2.4.6)$$

ここで、記号 $\langle \rangle \equiv \frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} dx = \int_{-\frac{1}{2}}^{\frac{1}{2}} d\xi$ とおき、以下の諸量を定義しておく。

$$\langle w_n(x), \bar{\phi}_{g,x}^k(x) \rangle \equiv \frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} w_n(x) \bar{\phi}_{g,x}^k(x) dx \equiv \phi_{g,xn}^k \quad (2.4.7)$$

$$\langle w_n(x), \bar{Q}_{g,x}^k(x) \rangle \equiv \frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} w_n(x) \bar{Q}_{g,x}^k(x) dx \equiv Q_{g,xn}^k \quad (2.4.8)$$

$$\langle w_n(x), \bar{L}_{g,y}^k(x) \rangle \equiv \frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} w_n(x) \bar{L}_{g,y}^k(x) dx \equiv L_{g,y,xn}^k \quad (2.4.9)$$

$$\langle w_n(x), \bar{L}_{g,z}^k(x) \rangle \equiv \frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} w_n(x) \bar{L}_{g,z}^k(x) dx \equiv L_{g,z,xn}^k \quad (2.4.10)$$

これらを使用して(2.4.6)式を書き換えると、

$$\left\langle w_n(x), \frac{d}{dx} \bar{J}_{g,x}^k(x) \right\rangle + \sum_{r,g}^k \phi_{g,xn}^k = Q_{g,xn}^k - \frac{1}{\Delta_y^k} L_{g,y,xn}^k - \frac{1}{\Delta_z^k} L_{g,z,xn}^k \quad (2.4.11)$$

ちなみに、(2.4.11)式で $w_0(x) \equiv 1$ とすれば、ノードバランス式(2.1.7)に一致する。展開係数 $a_{g,x,3}^k$ と $a_{g,x,4}^k$ を求めるためには、2つの重み関数 $w_1(x)$ 、 $w_2(x)$ を与える必要があるが、その与え方によりNEMのバリエーションが増え計算の精度にも影響する。多くは(2.3.4)式で定義した展開関数を使用する。

$$w_1(x) \equiv f_1(x) \quad (2.4.12a)$$

$$w_2(x) \equiv f_2(x) \quad (2.4.12b)$$

または、

$$w_1(x) \equiv f_3(x) \quad (2.4.13a)$$

$$w_2(x) \equiv f_4(x) \quad (2.4.13b)$$

重み関数として(2.4.12)式を使用する場合をモーメント重み法、(2.4.13)式を使用する場合をGalerkin重み法として知られている。Finnemann等の論文<sup>6)</sup>に依れば、Galerkin重み法よりも

モーメント重み法の方が計算精度が高くなることが数値計算により示されている。そこで、以下ではモーメント重みを採用する。

(2.4.11)式で、 $n=1$ とすれば、

$$\begin{aligned}
 \text{左辺第1項} &= \left\langle w_1(x), \frac{d}{dx} \bar{J}_{g,x}^k(x) \right\rangle \\
 &= \frac{1}{\Delta_x^k} \left\{ \left[ \frac{x}{\Delta_x^k} \bar{J}_{g,x}^k(x) \right]_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} - \frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} \bar{J}_{g,x}^k(x) dx \right\} \quad (\text{部分積分を実行}) \\
 &= \frac{1}{2\Delta_x^k} (J_{g,x+}^k + J_{g,x-}^k) + \frac{D_{g,x}^k}{(\Delta_x^k)^2} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} \frac{d\phi_{g,x}^k(x)}{dx} dx \quad (2.2.14a\text{式を代入}) \\
 &= \frac{1}{2\Delta_x^k} (J_{g,x+}^k + J_{g,x-}^k) + \frac{D_{g,x}^k}{(\Delta_x^k)^2} (\phi_{g,x+}^k - \phi_{g,x-}^k) \\
 &= \frac{1}{2\Delta_x^k} (J_{g,x+}^k + J_{g,x-}^k) + \frac{D_{g,x}^k}{(\Delta_x^k)^2} a_{g,x,1}^k \quad (2.3.11a\text{式を代入}) \tag{2.4.14}
 \end{aligned}$$

よって、1次のモーメントバランス式は以下のように書き表される。

$$\frac{1}{2\Delta_x^k} T_{g,x}^k + \frac{1}{\Delta_x^k} \frac{D_{g,x}^k}{\Delta_x^k} a_{g,x,1}^k + \sum_r \phi_{g,x1}^k = Q_{g,x1}^k - \frac{1}{\Delta_y^k} L_{g,y,x1}^k - \frac{1}{\Delta_z^k} L_{g,z,x1}^k \tag{2.4.15}$$

$$\text{ただし、} \quad T_{g,x}^k \equiv J_{g,x+}^k + J_{g,x-}^k \tag{2.4.16}$$

(2.4.11)式で、 $n=2$ とすれば、

$$\begin{aligned}
 \text{左辺第2項} &= \left\langle w_2(x), \frac{d}{dx} \bar{J}_{g,x}^k(x) \right\rangle \\
 &= \frac{1}{\Delta_x^k} \left\{ \left[ \left( \frac{3x^2}{(\Delta_x^k)^2} - \frac{1}{4} \right) \bar{J}_{g,x}^k(x) \right]_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} - \frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} \frac{6x}{(\Delta_x^k)^2} \bar{J}_{g,x}^k(x) dx \right\} \quad (\text{部分積分を実行}) \\
 &= \frac{1}{2\Delta_x^k} (J_{g,x+}^k - J_{g,x-}^k) + \frac{6D_{g,x}^k}{(\Delta_x^k)^3} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} x \frac{d\phi_{g,x}^k(x)}{dx} dx \quad (2.2.14a\text{式を代入})
 \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2\Delta_x^k} (J_{g,x+}^k - J_{g,x-}^k) + \frac{3D_{g,x}^k}{(\Delta_x^k)^2} (\phi_{g,x+}^k + \phi_{g,x-}^k - 2\phi_{g,0}^k) \\
&= \frac{1}{2\Delta_x^k} L_{g,x}^k + \frac{3}{\Delta_x^k} \frac{D_{g,x}^k}{\Delta_x^k} a_{g,x,2}^k \quad (2.2.21c, 2.3.11b式を代入)
\end{aligned} \tag{2.4.17}$$

よって、2次のモーメントバランス式は以下のように書き表される。

$$\frac{1}{2\Delta_x^k} L_{g,x}^k + \frac{1}{\Delta_x^k} \frac{D_{g,x}^k}{\Delta_x^k} a_{g,x,1}^k + \Sigma_{r,g}^k \phi_{g,x2}^k = Q_{g,x2}^k - \frac{1}{\Delta_y^k} L_{g,y,x2}^k - \frac{1}{\Delta_z^k} L_{g,z,x2}^k \tag{2.4.18}$$

また、(2.4.7), (2.4.8)式及び(2.2.9)式を使用して、x方向についてのモーメント中性子源は、モーメント中性子束を使用して以下のように表される。

$$Q_{g,x1}^k = \frac{\chi_g^k}{k_{eff}} \sum_{g'=1}^G v \Sigma_{f,g}^k \phi_{g',x1}^k + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{s,g' \rightarrow g}^k \phi_{g,x1}^k \tag{2.4.19a}$$

$$Q_{g,x2}^k = \frac{\chi_g^k}{k_{eff}} \sum_{g'=1}^G v \Sigma_{f,g}^k \phi_{g',x2}^k + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{s,g' \rightarrow g}^k \phi_{g,x2}^k \tag{2.4.19b}$$

次に、4次式に展開された1次元中性子束分布(2.3.9)式を(2.4.7)式に代入すれば、展開係数  $a_{g,x,3}^k$  および  $a_{g,x,4}^k$  と新たに現れたモーメント中性子束との関係式が得られる。

$$\begin{aligned}
\phi_{g,x1}^k &= \left\langle f_1(x), \phi_{g,0}^k + \sum_{n=1}^4 a_{g,x,n}^k f_n(x) \right\rangle \\
&= \frac{1}{12} a_{g,x,1}^k - \frac{1}{120} a_{g,x,3}^k \quad (2.3.7\text{式参照})
\end{aligned} \tag{2.4.20}$$

$$\begin{aligned}
\phi_{g,x2}^k &= \left\langle f_2(x), \phi_{g,0}^k + \sum_{n=1}^4 a_{g,x,n}^k f_n(x) \right\rangle \\
&= \frac{1}{20} a_{g,x,2}^k - \frac{1}{700} a_{g,x,4}^k \quad (2.3.8\text{式参照})
\end{aligned} \tag{2.4.21}$$

よって、

$$a_{g,x,3}^k = -120\phi_{g,x1}^k + 10a_{g,x,1}^k \tag{2.4.22}$$

$$a_{g,x,4}^k = -700\phi_{g,x2}^k + 35a_{g,x,2}^k \tag{2.4.23}$$

これらの式と、(2.3.15)式及び(2.3.16)式を使用して、(2.3.17)式から展開係数を全て消去すると、以下の式を得る。

$$J_{g,x+}^{out,k} = \frac{D_{g,x}^k}{\Delta_x^k} \left\{ -32J_{g,x+}^{out,k} - 32J_{g,x+}^{in,k} - 8J_{g,x-}^{out,k} - 8J_{g,x-}^{in,k} + 20\phi_{g,0}^k + 60\phi_{g,x1}^k + 140\phi_{g,x2}^k \right\} + J_{g,x+}^{in,k} \quad (2.4.24a)$$

$$J_{g,x-}^{out,k} = \frac{D_{g,x}^k}{\Delta_x^k} \left\{ -8J_{g,x+}^{out,k} - 8J_{g,x+}^{in,k} - 32J_{g,x-}^{out,k} - 32J_{g,x-}^{in,k} + 20\phi_{g,0}^k - 60\phi_{g,x1}^k + 140\phi_{g,x2}^k \right\} + J_{g,x-}^{in,k} \quad (2.4.24b)$$

(2.4.6)式～(2.4.24)式は、x方向に着目して得られた式であるが、y方向とz方向についても同様な関係式が得られる。結局、1つのノードバランス式の他に、6つのモーメントバランス式が得されることになる。式が煩雑になったが、各方向に対する3次及び4次の展開係数を求めるために重み付き残差法を導入して、ノード内の中性子束分布が拡散方程式を近似的に満足するように方程式の数を増やしたわけである。このことは、粗メッシュを使用する代償として、ノードバランス式に加えて解くべき方程式の数が増えたことを意味する。したがって、最終的に得られる計算精度を無視するならば、同じメッシュ(ノード)サイズを使用する有限差分法(または2次の多項式展開ノード法)の方が、余分な方程式を解く必要がない分、計算時間は速くなる。しかし十分な精度を得ようとする場合、粗メッシュを使用して余分な方程式を解く手間は、メッシュを細かくしてノードバランス式だけで解く手間に比べるとはるかに小さい。このことが近代ノード法を優れた解法にしている。

## 2.5 応答行列式の導出

前節で得られた方程式を整理して、部分中性子流に関する応答行列式に整理する。得られる結果は、体系スウェーブ(内部反復計算)に直接利用することができる。

### 2.5.1 1次元問題2次多項式展開における応答行列式

先ず、 $x$ 方向1次元問題を考え展開次数Nが2の場合の応答行列式を導出する。2次の多項式展開では、重み付き残差法は必要としないため、第2.4.1節で得られた結果から応答行列式を作成することができる。先ず、1次元問題におけるノードバランス式(2.4.5)式をノード平均中性子束について整理する。

$$\phi_{g,0}^k = \frac{1}{\sum_{r,g}^k} \left\{ Q_{g,0}^k - \frac{1}{\Delta_x^k} \left( J_{g,x+}^{out,k} - J_{g,x+}^{in,k} + J_{g,x-}^{out,k} - J_{g,x-}^{in,k} \right) \right\} \quad (2.5.1)$$

$$Q_{g,0}^k = \frac{\chi_g^k}{k_{eff}} \sum_{g'=1}^G v \sum_{f,g}^k \phi_{g',0}^k + \sum_{g'=1}^G \sum_{s,g' \rightarrow g}^k \phi_{g',0}^k \quad (2.5.2)$$

(2.5.1)式を、(2.4.3)式及び(2.4.4)式に代入して整理すると以下のような行列式を得る。

$$\begin{bmatrix} 1+8d_x+6d_x\lambda_x & 4d_x+6d_x\lambda_x \\ 4d_x+6d_x\lambda_x & 1+8d_x+6d_x\lambda_x \end{bmatrix} \begin{pmatrix} J_{g,x+}^{out,k} \\ J_{g,x-}^{out,k} \end{pmatrix} = \begin{bmatrix} 1-8d_x+6d_x\lambda_x & -4d_x+6d_x\lambda_x \\ -4d_x+6d_x\lambda_x & 1-8d_x+6d_x\lambda_x \end{bmatrix} \begin{pmatrix} J_{g,x+}^{in,k} \\ J_{g,x-}^{in,k} \end{pmatrix} + \begin{bmatrix} 6d_x\lambda_x\Delta_x^k \\ 6d_x\lambda_x\Delta_x^k \end{bmatrix} \begin{pmatrix} Q_{g,0}^k \\ Q_{g,0}^k \end{pmatrix} \quad (2.5.3)$$

ただし、

$$d_x \equiv \frac{D_{g,x}^k}{\Delta_x^k} \quad [\text{無次元}] \quad (2.5.4a)$$

$$\lambda_x \equiv \frac{1}{\sum_{r,g}^k \Delta_x^k} \quad [\text{無次元}] \quad (2.5.4b)$$

ここで、(2.5.3)式を以下のように表す。

$$[A_g^k] J_g^{out,k} = [B_g^k] J_g^{in,k} + [C_g^k] Q_g^k \quad (2.5.5)$$

$[A_g^k]$ は正則であり、以下の行列式を得る。

$$\begin{aligned}
J_g^{out,k} &= [A_g^k]^{-1} [B_g^k] J_g^{in,k} + [A_g^k]^{-1} [C_g^k] Q_g^k \\
&= [P_g^k] J_g^{in,k} + [R_g^k] Q_g^k \quad \text{ただし、} [P_g^k] \equiv [A_g^k]^{-1} [B_g^k], [R_g^k] \equiv [A_g^k]^{-1} [C_g^k]
\end{aligned} \tag{2.5.6}$$

外部反復計算において  $Q_{g,0}^k$  ( $k=1,2,\dots,K$ ) と  $J_g^{in,k}$  ( $k=1,2,\dots,K$ ) の漸近値(または初期値)が与えられた場合、内部反復計算は以下のように行われる。

- (a) (2.5.6)式により、 $J_g^{out,k}$  ( $k=1,2,\dots,K$ ) を更新する。
- (b) 境界条件及び隣接ノード境界面における部分中性子流の連続条件から  $J_g^{in,k}$  ( $k=1,2,\dots,K$ ) を更新する。
- (c) 更新された  $J_g^{out,k}$  と  $J_g^{in,k}$  を(2.5.1)式に代入して  $\phi_{g,0}^k$  ( $k=1,2,\dots,K$ ) を更新する。
- (d) 内部反復計算として  $\phi_{g,0}^k$  の収束判定をする。内部反復回数の上限内で収束していないければ(a)に戻る。

## 2.5.2 3次元問題2次多項式展開における応答行列式

第2.5.1節で扱った2次の多項式展開法を3次元問題に拡張して応答行列式を導出する。3次元問題に対するノードバランス式(2.1.7)式をノード平均中性子束について整理すると、

$$\begin{aligned}
\phi_{g,0}^k &= \frac{1}{\sum_{r,g}^k} \left\{ Q_{g,0}^k - \frac{1}{\Delta_x} \left( J_{g,x+}^{out,k} - J_{g,x+}^{in,k} + J_{g,x-}^{out,k} - J_{g,x-}^{in,k} \right) - \frac{1}{\Delta_y} \left( J_{g,y+}^{out,k} - J_{g,y+}^{in,k} + J_{g,y-}^{out,k} - J_{g,y-}^{in,k} \right) \right. \\
&\quad \left. - \frac{1}{\Delta_z} \left( J_{g,z+}^{out,k} - J_{g,z+}^{in,k} + J_{g,z-}^{out,k} - J_{g,z-}^{in,k} \right) \right\}
\end{aligned} \tag{2.5.7}$$

(2.5.7)式を(2.4.3)式及び(2.4.4)式に代入して整理すると以下の2式を得る。

$$\begin{aligned}
&(1+8d_x+6d_x\lambda_x)J_{g,x+}^{out,k} + (4d_x+6d_x\lambda_x)J_{g,x+}^{out,k} + 6d_x\lambda_y J_{g,y+}^{out,k} + 6d_x\lambda_y J_{g,y-}^{out,k} + 6d_x\lambda_z J_{g,z+}^{out,k} + 6d_x\lambda_z J_{g,z-}^{out,k} \\
&= (1-8d_x+6d_x\lambda_x)J_{g,x+}^{in,k} + (-4d_x+6d_x\lambda_x)J_{g,x-}^{in,k} + 6d_x\lambda_y J_{g,y+}^{in,k} + 6d_x\lambda_y J_{g,y-}^{in,k} + 6d_x\lambda_z J_{g,z+}^{in,k} + 6d_x\lambda_z J_{g,z-}^{in,k} \\
&\quad + 6d_x\lambda_x \Delta_x^k Q_{g,0}^k
\end{aligned} \tag{2.5.8a}$$

$$\begin{aligned}
&(4d_x+6d_x\lambda_x)J_{g,x+}^{out,k} + (1+8d_x+6d_x\lambda_x)J_{g,x+}^{out,k} + 6d_x\lambda_y J_{g,y+}^{out,k} + 6d_x\lambda_y J_{g,y-}^{out,k} + 6d_x\lambda_z J_{g,z+}^{out,k} + 6d_x\lambda_z J_{g,z-}^{out,k} \\
&= (-4d_x+6d_x\lambda_x)J_{g,x+}^{in,k} + (1-8d_x+6d_x\lambda_x)J_{g,x-}^{in,k} + 6d_x\lambda_y J_{g,y+}^{in,k} + 6d_x\lambda_y J_{g,y-}^{in,k} + 6d_x\lambda_z J_{g,z+}^{in,k} + 6d_x\lambda_z J_{g,z-}^{in,k} \\
&\quad + 6d_x\lambda_x \Delta_x^k Q_{g,0}^k
\end{aligned} \tag{2.5.8b}$$

(2.4.3)式と(2.4.4)式はx方向に関して得られた式であるが、y方向及びz方向についても同様な式が得られる。それらに(2.5.7)式を代入すれば、(2.5.8a)式及び(2.5.8b)式に対応する式が4つ得られ、計6つの式を以下のような行列式で表わすことができる。

$$[A_g^k]J_g^{out,k} = [B_g^k]J_g^{in,k} + [C_g^k]Q_g^k \quad (2.5.9)$$

ただし、

$$[A_g^k] \equiv \begin{bmatrix} 1+8d_x+6d_x\lambda_x & 4d_x+6d_x\lambda_x & 6d_x\lambda_y & 6d_x\lambda_y & 6d_x\lambda_z & 6d_x\lambda_z \\ 4d_x+6d_x\lambda_x & 1+8d_x+6d_x\lambda_x & 6d_x\lambda_y & 6d_x\lambda_y & 6d_x\lambda_z & 6d_x\lambda_z \\ 6d_y\lambda_x & 6d_y\lambda_x & 1+8d_y+6d_y\lambda_y & 4d_y+6d_y\lambda_y & 6d_y\lambda_z & 6d_y\lambda_z \\ 6d_y\lambda_x & 6d_y\lambda_x & 4d_y+6d_y\lambda_y & 1+8d_y+6d_y\lambda_y & 6d_y\lambda_z & 6d_y\lambda_z \\ 6d_z\lambda_x & 6d_z\lambda_x & 6d_z\lambda_y & 6d_z\lambda_y & 1+8d_z+6d_z\lambda_z & 4d_z+6d_z\lambda_z \\ 6d_z\lambda_x & 6d_z\lambda_x & 6d_z\lambda_y & 6d_z\lambda_y & 4d_z+6d_z\lambda_z & 1+8d_z+6d_z\lambda_z \end{bmatrix} \quad (2.5.10)$$

$$[B_g^k] \equiv \begin{bmatrix} 1-8d_x+6d_x\lambda_x & -4d_x+6d_x\lambda_x & 6d_x\lambda_y & 6d_x\lambda_y & 6d_x\lambda_z & 6d_x\lambda_z \\ -4d_x+6d_x\lambda_x & 1-8d_x+6d_x\lambda_x & 6d_x\lambda_y & 6d_x\lambda_y & 6d_x\lambda_z & 6d_x\lambda_z \\ 6d_y\lambda_x & 6d_y\lambda_x & 1-8d_y+6d_y\lambda_y & -4d_y+6d_y\lambda_y & 6d_y\lambda_z & 6d_y\lambda_z \\ 6d_y\lambda_x & 6d_y\lambda_x & -4d_y+6d_y\lambda_y & 1-8d_y+6d_y\lambda_y & 6d_y\lambda_z & 6d_y\lambda_z \\ 6d_z\lambda_x & 6d_z\lambda_x & 6d_z\lambda_y & 6d_z\lambda_y & 1-8d_z+6d_z\lambda_z & -4d_z+6d_z\lambda_z \\ 6d_z\lambda_x & 6d_z\lambda_x & 6d_z\lambda_y & 6d_z\lambda_y & -4d_z+6d_z\lambda_z & 1-8d_z+6d_z\lambda_z \end{bmatrix} \quad (2.5.11)$$

$$[C_g^k] \equiv [6d_x\lambda_x\Delta_x^k, 6d_x\lambda_x\Delta_x^k, 6d_y\lambda_y\Delta_y^k, 6d_y\lambda_y\Delta_y^k, 6d_z\lambda_z\Delta_z^k, 6d_z\lambda_z\Delta_z^k] \quad (2.5.12)$$

$$J_g^{out,k} \equiv \text{col}[J_{g,x+}^{out,k}, J_{g,x-}^{out,k}, J_{g,y+}^{out,k}, J_{g,y-}^{out,k}, J_{g,z+}^{out,k}, J_{g,z-}^{out,k}] \quad (2.5.13)$$

$$J_g^{in,k} \equiv \text{col}[J_{g,x+}^{in,k}, J_{g,x-}^{in,k}, J_{g,y+}^{in,k}, J_{g,y-}^{in,k}, J_{g,z+}^{in,k}, J_{g,z-}^{in,k}] \quad (2.5.14)$$

$$Q_g^k \equiv \text{col}[Q_{g,0}^k, Q_{g,0}^k, Q_{g,0}^k, Q_{g,0}^k, Q_{g,0}^k, Q_{g,0}^k] \quad (2.5.15)$$

$d_y, d_z, \lambda_y, \lambda_z$ は、x方向に関する(2.5.4)式の定義に準ずるものとする。

よって、3次元問題においてもk番目ノードに対する表面平均部分中性子流の応答行列式(2.5.16式)を得る。

$$\begin{aligned} J_g^{out,k} &= [A_g^k]^{-1} [B_g^k] J_g^{in,k} + [A_g^k]^{-1} [C_g^k] Q_g^k \\ &= [P_g^k] J_g^{in,k} + [R_g^k] Q_g^k \quad \text{ただし、} [P_g^k] \equiv [A_g^k]^{-1} [B_g^k], [R_g^k] \equiv [A_g^k]^{-1} [C_g^k] \end{aligned} \quad (2.5.16)$$

### 2.5.3 3次元問題4次多項式展開における応答行列式

第2.5.1節及び第2.5.2節においては、2次の多項式展開を扱うため、3次及び4次の展開係数を無視した(2.4.3)式と(2.4.4)式を用いた。ここでは、これらの式の代わりに(2.4.24a)式と(2.4.24b)式を使用する。これらの式には、ノード平均中性子束の他に1次及び2次のモーメント中性子束が含まれる。これらを消去するために、モーメントバランス式(2.4.15)式及び(2.4.18)式を以下のように変形する。

先ず、1次のモーメントバランス式から、

$$\phi_{g,x1}^k = \frac{1}{\sum_{r,g}^k} \left\{ Q_{g,x1}^k - \frac{1}{\Delta_y} L_{g,y,x1}^k - \frac{1}{\Delta_z} L_{g,z,x1}^k - \frac{1}{2\Delta_x} T_{g,x}^k - \frac{1}{\Delta_x} \frac{D_{g,x}^k}{\Delta_x} a_{g,x,1}^k \right\} \quad (2.5.17)$$

$$\text{ここで、 } \frac{1}{\Delta_y} L_{g,y,xn}^k + \frac{1}{\Delta_z} L_{g,z,xn}^k \equiv L_{g,xn}^k \quad (n=1,2) \quad (2.5.18)$$

を定義し、(2.3.15)式により(2.5.17)式から  $a_{g,x,1}^k$  を消去すると、

$$\phi_{g,x1}^k = \frac{1}{\sum_{r,g}^k} \left\{ Q_{g,x1}^k - L_{g,x1}^k - \frac{1}{2\Delta_x} (J_{g,x+}^{out,k} - J_{g,x+}^{in,k} - J_{g,x-}^{out,k} + J_{g,x-}^{in,k}) - \frac{2D_{g,x}^k}{(\Delta_x)^2} (J_{g,x+}^{out,k} + J_{g,x+}^{in,k} - J_{g,x-}^{out,k} - J_{g,x-}^{in,k}) \right\} \quad (2.5.19)$$

2次のモーメントバランス式についても同様に、

$$\phi_{g,x2}^k = \frac{1}{\sum_{r,g}^k} \left\{ Q_{g,x2}^k - L_{g,x2}^k - \frac{1}{2\Delta_x} (J_{g,x+}^{out,k} - J_{g,x+}^{in,k} + J_{g,x-}^{out,k} - J_{g,x-}^{in,k}) - \frac{6D_{g,x}^k}{(\Delta_x)^2} (J_{g,x+}^{out,k} + J_{g,x+}^{in,k} + J_{g,x-}^{out,k} + J_{g,x-}^{in,k} - \phi_{g,0}^k) \right\} \quad (2.5.20)$$

(2.5.19)式と(2.5.20)式を(2.4.24a)式に代入して  $\phi_{g,1}^k$  と  $\phi_{g,2}^k$  を消去する。 $\phi_{g,0}^k$  が残るが、これを(2.5.7)式により消去し、部分中性子流に関して整理する。次に(2.5.19)式と(2.5.20)式を(2.4.24b)式に代入して整理する。y方向及びz方向についても同様の式が得られ、計6つの式が得られる。これを行列式で表すと以下のようになる。

$$[A_g^k] J_g^{out,k} = [B_g^k] J_g^{in,k} + [C_g^k] (Q_g^k - L_g^k) \quad (2.5.21)$$

ただし、

$$J_g^{out,k} \equiv \text{col} [J_{g,x+}^{out,k}, J_{g,x-}^{out,k}, J_{g,y+}^{out,k}, J_{g,y-}^{out,k}, J_{g,z+}^{out,k}, J_{g,z-}^{out,k}] \quad (2.5.22)$$

$$\mathbf{J}_g^{in,k} \equiv \text{col} [J_{g,x+}^{in,k}, J_{g,x-}^{in,k}, J_{g,y+}^{in,k}, J_{g,y-}^{in,k}, J_{g,z+}^{in,k}, J_{g,z-}^{in,k}] \quad (2.5.23)$$

$$\begin{aligned} \mathbf{Q}_g^k &\equiv \text{col} [Q_{g,0}^k, Q_{g,x1}^k, Q_{g,y1}^k, Q_{g,z1}^k, Q_{g,x2}^k, Q_{g,y2}^k, Q_{g,z2}^k] \\ &= \frac{\chi_g^k}{k_{\text{eff}}} \sum_{g'=1}^G v \Sigma_{f,g'}^k \Phi_{g'}^k + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{s,g' \rightarrow g}^k \Phi_{g'}^k \end{aligned} \quad (2.5.24)$$

$$\Phi_g^k \equiv \text{col} [\phi_{g,0}^k, \phi_{g,x1}^k, \phi_{g,y1}^k, \phi_{g,z1}^k, \phi_{g,x2}^k, \phi_{g,y2}^k, \phi_{g,z2}^k] \quad (2.5.25)$$

$$\mathbf{L}_g^k \equiv \text{col} [0, L_{g,x1}^k, L_{g,y1}^k, L_{g,z1}^k, L_{g,x2}^k, L_{g,y2}^k, L_{g,z2}^k] \quad (2.5.26)$$

また、 $6 \times 6$ 行列の $[A_g^k]$ 、 $[B_g^k]$ はそれぞれ、次ページの(2.5.28)式、(2.5.29)式で与えられる。 $[C_g^k]$ は(2.5.30)式に示すように $6 \times 7$ 行列となる。(2.5.21)式を次のように変形すれば、3次元問題4次の多項式展開に対する応答行列式が得られる。

$$\begin{aligned} \mathbf{J}_g^{out,k} &= [A_g^k]^{-1} [B_g^k] \mathbf{J}_g^{in,k} + [A_g^k]^{-1} [C_g^k] (\mathbf{Q}_g^k - \mathbf{L}_g^k) \\ &= [P_g^k] \mathbf{J}_g^{in,k} + [R_g^k] (\mathbf{Q}_g^k - \mathbf{L}_g^k) \end{aligned} \quad (2.5.27)$$

$$\text{ただし、} [P_g^k] \equiv [A_g^k]^{-1} [B_g^k], [R_g^k] \equiv [A_g^k]^{-1} [C_g^k]$$

同じ3次元問題でも、2次の多項式展開の応答行列式と大きく異なる点は、 $\mathbf{L}_g^k$ が現れることがある。この項を決定するために、更にもう一つ重要な近似を導入することになる。これについては次節で述べる。

$$[A] = \begin{bmatrix} 1+32d_x + 12d_x\lambda_x + 96d_x^2\lambda_x^2 + 840d_x^3\lambda_x^3 & 8d_x + 60d_x^2\lambda_x^2 + 720d_x^3\lambda_x^3 & 20d_x\lambda_y + 840d_x^2\lambda_x^2\lambda_y & 20d_x\lambda_z + 840d_x^2\lambda_x^2\lambda_z & 20d_x\lambda_x + 840d_x^2\lambda_x^2\lambda_x \\ 8d_x + 60d_x^2\lambda_x^2 + 720d_x^3\lambda_x^3 + 840d_x^4\lambda_x^4 & 1+32d_x + 120d_x\lambda_x + 960d_x^2\lambda_x^2 + 840d_x^3\lambda_x^3 & 20d_x\lambda_y + 840d_x^2\lambda_x^2\lambda_y & 20d_x\lambda_z + 840d_x^2\lambda_x^2\lambda_z & 20d_x\lambda_x + 840d_x^2\lambda_x^2\lambda_x \\ 20d_x\lambda_x + 840d_x^2\lambda_x^2\lambda_y & 20d_x\lambda_x + 840d_x^2\lambda_x^2\lambda_z & 1+32d_y + 120d_y\lambda_y + 960d_y^2\lambda_y^2 + 840d_y^3\lambda_y^3 & 20d_y + 60d_y\lambda_y + 720d_y^2\lambda_y^2 + 840d_y^3\lambda_y^3 & 20d_y\lambda_x + 840d_y^2\lambda_y^2\lambda_x \\ 20d_y\lambda_x + 840d_y^2\lambda_y^2\lambda_x & 20d_y\lambda_x + 840d_y^2\lambda_y^2\lambda_z & 8d_y + 60d_y\lambda_y + 720d_y^2\lambda_y^2 + 840d_y^3\lambda_y^3 & 8d_y + 60d_y\lambda_y + 720d_y^2\lambda_y^2 + 840d_y^3\lambda_y^3 & 20d_y\lambda_x + 840d_y^2\lambda_y^2\lambda_z \\ 20d_z\lambda_x + 840d_z^2\lambda_z^2\lambda_y & 20d_z\lambda_x + 840d_z^2\lambda_z^2\lambda_z & 20d_z\lambda_y + 840d_z^2\lambda_z^2\lambda_y & 20d_z\lambda_y + 840d_z^2\lambda_z^2\lambda_y & 20d_z\lambda_x + 840d_z^2\lambda_z^2\lambda_x \\ 20d_z\lambda_x + 840d_z^2\lambda_z^2\lambda_z & & 20d_z\lambda_y + 840d_z^2\lambda_z^2\lambda_z & 20d_z\lambda_y + 840d_z^2\lambda_z^2\lambda_z & 20d_z\lambda_x + 840d_z^2\lambda_z^2\lambda_x \end{bmatrix} \quad (2.5.28)$$

$$[B] = \begin{bmatrix} 1-32d_y + 120d_y\lambda_y - 960d_y^2\lambda_y^2 + 840d_y^3\lambda_y^3 & -8d_y + 60d_y\lambda_y - 720d_y^2\lambda_y^2 + 840d_y^3\lambda_y^3 & 20d_y\lambda_x + 60d_y^2\lambda_x^2\lambda_y & 20d_y\lambda_x + 840d_y^2\lambda_x^2\lambda_x & 20d_y\lambda_x + 840d_y^2\lambda_x^2\lambda_x \\ -8d_y + 60d_y\lambda_y - 720d_y^2\lambda_y^2 + 840d_y^3\lambda_y^3 & 1-32d_x + 120d_x\lambda_x - 960d_x^2\lambda_x^2 + 840d_x^3\lambda_x^3 & 20d_x\lambda_y + 840d_x^2\lambda_x^2\lambda_y & 20d_x\lambda_y + 840d_x^2\lambda_x^2\lambda_x & 20d_x\lambda_x + 840d_x^2\lambda_x^2\lambda_x \\ 20d_y\lambda_x + 840d_y^2\lambda_x^2\lambda_y & 20d_y\lambda_x + 840d_y^2\lambda_x^2\lambda_z & 1-32d_y + 120d_y\lambda_y - 960d_y^2\lambda_y^2 + 840d_y^3\lambda_y^3 & -8d_y + 60d_y\lambda_y - 720d_y^2\lambda_y^2 + 840d_y^3\lambda_y^3 & 20d_y\lambda_x + 840d_y^2\lambda_x^2\lambda_x \\ 20d_y\lambda_x + 840d_y^2\lambda_y^2\lambda_x & 20d_z\lambda_x + 840d_z^2\lambda_z^2\lambda_x & -8d_y + 60d_y\lambda_y - 720d_y^2\lambda_y^2 + 840d_y^3\lambda_y^3 & 1-32d_y + 120d_y\lambda_y - 960d_y^2\lambda_y^2 + 840d_y^3\lambda_y^3 & 20d_y\lambda_x + 840d_y^2\lambda_x^2\lambda_x \\ 20d_z\lambda_x + 840d_z^2\lambda_z^2\lambda_x & 20d_z\lambda_x + 840d_z^2\lambda_z^2\lambda_y & 20d_z\lambda_y + 840d_z^2\lambda_z^2\lambda_y & 20d_z\lambda_y + 840d_z^2\lambda_z^2\lambda_y & 20d_z\lambda_x + 840d_z^2\lambda_x^2\lambda_x \\ 20d_z\lambda_x + 840d_z^2\lambda_z^2\lambda_y & & 20d_z\lambda_y + 840d_z^2\lambda_z^2\lambda_y & 20d_z\lambda_y + 840d_z^2\lambda_z^2\lambda_y & 20d_z\lambda_x + 840d_z^2\lambda_x^2\lambda_x \end{bmatrix} \quad (2.5.29)$$

$$[C] = \begin{bmatrix} 20d_x\lambda_x\Delta_x^k + 840d_x^2\lambda_x^2\Delta_x^k & 60d_x\lambda_x\Delta_x^k & 0 & 0 & 140d_x\lambda_x\Delta_x^k & 0 & 0 \\ 20d_x\lambda_x\Delta_x^k + 840d_x^2\lambda_x^2\Delta_x^k & -60d_x\lambda_x\Delta_x^k & 0 & 0 & 140d_x\lambda_x\Delta_x^k & 0 & 0 \\ 20d_y\lambda_y\Delta_y^k + 840d_y^2\lambda_y^2\Delta_y^k & 0 & 60d_y\lambda_y\Delta_y^k & 0 & 0 & 140d_y\lambda_y\Delta_y^k & 0 \\ 20d_y\lambda_y\Delta_y^k + 840d_y^2\lambda_y^2\Delta_y^k & 0 & -60d_y\lambda_y\Delta_y^k & 0 & 0 & 140d_y\lambda_y\Delta_y^k & 0 \\ 20d_z\lambda_z\Delta_z^k + 840d_z^2\lambda_z^2\Delta_z^k & 0 & 0 & 60d_z\lambda_z\Delta_z^k & 0 & 0 & 140d_z\lambda_z\Delta_z^k \\ 20d_z\lambda_z\Delta_z^k + 840d_z^2\lambda_z^2\Delta_z^k & 0 & 0 & -60d_z\lambda_z\Delta_z^k & 0 & 0 & 140d_z\lambda_z\Delta_z^k \end{bmatrix} \quad (2.5.30)$$

## 2.6 ノード内トランスバース漏洩分布の近似

4次の多項式展開に対応する応答行列式(2.5.27)式を体系スウェーブに使用する場合、ベクトル $\mathbf{L}_g^k$ を与える必要がある。この中には重み付き残差法の導入により現れた1次及び2次のトランスバース漏洩項が含まれている。例えばx方向に関して、

$$L_{g,xn}^k = \frac{1}{\Delta_y^k} L_{g,y,xn}^k + \frac{1}{\Delta_z^k} L_{g,z,xn}^k \quad (n=1,2) \quad (2.6.1)$$

を与える必要がある。ここで、第2.4.2節で定義したように、(2.6.1)式中の $L_{g,y,xn}^k$ と $L_{g,z,xn}^k$ は以下の式で与えられる。

$$L_{g,y,xn}^k = \frac{1}{\Delta_x^k} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} w_n(x) \bar{L}_{g,y}^k(x) dx \quad (2.6.2a)$$

$$L_{g,z,xn}^k = \frac{1}{\Delta_x^k} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} w_n(x) \bar{L}_{g,z}^k(x) dx \quad (2.6.2b)$$

$\bar{L}_{g,y}^k(x)$ と $\bar{L}_{g,z}^k(x)$ は、(2.2.10)式と(2.2.11)式で定義される量であるが、これまでの議論ではこれら各ノード内でのy方向及びz方向への正味中性子漏洩のx方向1次元分布を与えることはできない。この分布を与えるため幾つかの近似法が存在する。

### 2.6.1 平坦近似 (Flat Transverse Leakage Approximation)

これは、 $\bar{L}_{g,y}^k(x)$ と $\bar{L}_{g,z}^k(x)$ のノード内の分布を無視してノード平均値で扱う方法である。すなわち、

$$\bar{L}_{g,y}^k(x) \approx \frac{1}{\Delta_x^k} \int_{-\Delta_x^k/2}^{\Delta_x^k/2} \bar{L}_{g,y}^k(x) dx = L_{g,y}^k \quad (2.2.21\text{式参照}) \quad (2.6.3a)$$

$$\bar{L}_{g,z}^k(x) \approx \frac{1}{\Delta_z^k} \int_{-\Delta_z^k/2}^{\Delta_z^k/2} \bar{L}_{g,z}^k(x) dx = L_{g,z}^k \quad (2.6.3b)$$

これらを(2.6.2)式に代入すると、(2.4.12)式と(2.3.6)式より

$$L_{g,y,xn}^k = \frac{L_{g,y}^k}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} w_n(x) dx = 0 \quad (n=1,2) \quad (2.6.4a)$$

$$L_{g,z,xn}^k = \frac{L_{g,z}^k}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} w_n(x) dx = 0 \quad (n=1,2) \quad (2.6.4b)$$

結局、 $L_{g,xn}^k=0$ 、 $L_{g,yn}^k=0$ 、 $L_{g,zn}^k=0$ となり、平坦近似は(2.5.21)式の応答行列式において、ベクトル $\mathbf{L}_g^k$ を無視する扱いとなる。

## 2.6.2 バックリング近似 (Buckling Transverse Leakage Approximation)

バックリング近似は初期のノード法コードに使用された方法で、着目方向に対してノード側面からの漏洩(Transverse Leakage)は、1次元中性子束分布と同じ分布をすると仮定するものである。例えばk番目ノードのx方向中性子束分布は以下の式に従う(2.2.15式参照)が、

$$-D_{g,x}^k \frac{d^2 \bar{\phi}_{g,x}^k(x)}{dx^2} + \sum_{r,g}^k \bar{\phi}_{g,x}^k(x) = \bar{Q}_{g,x}^k(x) - \left\{ \frac{1}{\Delta_y^k} \bar{L}_{g,y}^k(x) + \frac{1}{\Delta_z^k} \bar{L}_{g,z}^k(x) \right\} \quad (2.6.5)$$

ここで、

$$\frac{1}{\Delta_y^k} \bar{L}_{g,y}^k(x) + \frac{1}{\Delta_z^k} \bar{L}_{g,z}^k(x) \approx (DB^2)_{g,x}^k \bar{\phi}_g^k(x) \quad (2.6.6)$$

と近似する。 $(DB^2)_{g,x}^k$ は(2.6.6)式の両辺を $(-\frac{1}{2}\Delta_x^k \leq x \leq \frac{1}{2}\Delta_x^k)$ で積分し更に $\Delta_x^k$ で割ることにより得られる。

$$\frac{1}{\Delta_y^k} \frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} \bar{L}_{g,y}^k(x) dx + \frac{1}{\Delta_z^k} \frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} \bar{L}_{g,z}^k(x) dx = (DB^2)_{g,x}^k \frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} \bar{\phi}_g^k(x) dx \quad (2.6.7)$$

(2.2.18)式と(2.2.21)式より、

$$\begin{aligned} (DB^2)_{g,x}^k &= \frac{1}{\phi_{g,0}^k} \left\{ \frac{1}{\Delta_y^k} L_{g,y}^k + \frac{1}{\Delta_z^k} L_{g,z}^k \right\} \\ &= \frac{1}{\phi_{g,0}^k} \left\{ \frac{1}{\Delta_y^k} (J_{g,y+}^k - J_{g,y-}^k) + \frac{1}{\Delta_z^k} (J_{g,z+}^k - J_{g,z-}^k) \right\} \\ &= \frac{1}{\phi_{g,0}^k} \left\{ \frac{1}{\Delta_y^k} (J_{g,y+}^{out,k} - J_{g,y+}^{in,k} - J_{g,y-}^{in,k} + J_{g,y-}^{out,k}) + \frac{1}{\Delta_z^k} (J_{g,z+}^{out,k} - J_{g,z+}^{in,k} - J_{g,z-}^{in,k} + J_{g,z-}^{out,k}) \right\} \quad (2.6.8a) \end{aligned}$$

y方向及びz方向についても同様に、

$$(DB^2)_{g,y}^k = \frac{1}{\phi_{g,0}^k} \left\{ \frac{1}{\Delta_z^k} (J_{g,z+}^{out,k} - J_{g,z+}^{in,k} - J_{g,z-}^{in,k} + J_{g,z-}^{out,k}) + \frac{1}{\Delta_y^k} (J_{g,y+}^{out,k} - J_{g,y+}^{in,k} - J_{g,y-}^{in,k} + J_{g,y-}^{out,k}) \right\} \quad (2.6.8b)$$

$$(DB^2)_{g,z}^k = \frac{1}{\phi_{g,0}^k} \left\{ \frac{1}{\Delta_x^k} (J_{g,x+}^{out,k} - J_{g,x+}^{in,k} - J_{g,x-}^{in,k} + J_{g,x-}^{out,k}) + \frac{1}{\Delta_y^k} (J_{g,y+}^{out,k} - J_{g,y+}^{in,k} - J_{g,y-}^{in,k} + J_{g,y-}^{out,k}) \right\} \quad (2.6.8c)$$

$(DB^2)_{g,x}^k, (DB^2)_{g,y}^k, (DB^2)_{g,z}^k$ は、第2.5.3節の応答行列式を使用した内部反復計算において得られる部分中性子流  $J_g^k$ (または初期値)を使用して(2.6.8)式により計算することができる。  
(2.6.1)式と(2.6.2)式より、

$$L_{g,xn}^k = \frac{1}{\Delta_x^k} \int_{-\Delta_k/2}^{\Delta_k/2} w_n(x) \left\{ \frac{\bar{L}_{g,y}^k(x)}{\Delta_y^k} + \frac{\bar{L}_{g,z}^k(x)}{\Delta_z^k} \right\} dx \quad (n=1,2) \quad (2.6.9)$$

これに(2.6.6)式を代入すると、

$$\begin{aligned} L_{g,xn}^k &= \frac{(DB^2)_{g,x}^k}{\Delta_x^k} \int_{-\Delta_k/2}^{\Delta_k/2} w_n(x) \bar{\phi}_g^k(x) dx \\ &= (DB^2)_{g,x}^k \phi_{g,xn}^k \quad (n=1,2) \end{aligned} \quad (2.6.10a)$$

y方向及びz方向についても同様に、

$$L_{g,yn}^k = (DB^2)_{g,y}^k \phi_{g,yn}^k \quad (n=1,2) \quad (2.6.10b)$$

$$L_{g,zn}^k = (DB^2)_{g,z}^k \phi_{g,zn}^k \quad (n=1,2) \quad (2.6.10c)$$

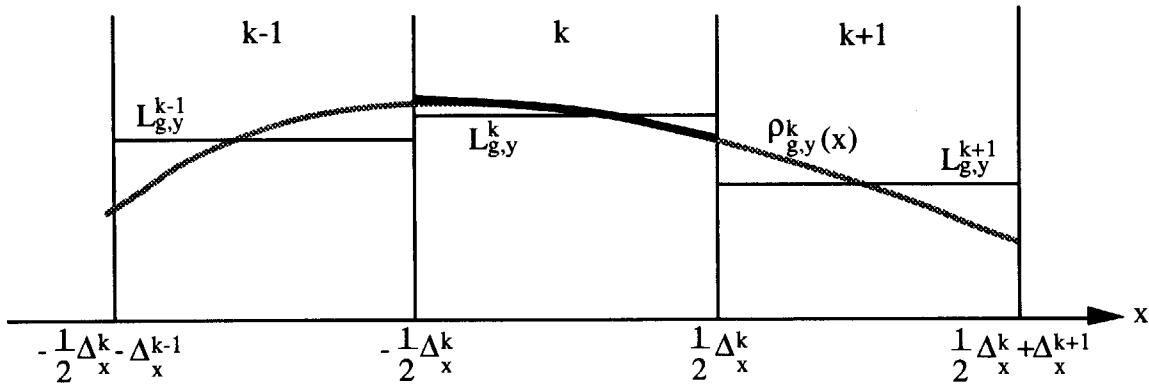
### 2.6.3 2次式近似 (Quadratic Transverse Leakage Approximation)

これは、近年のノード法コードで広く使用されている方法で、着目するノード及びこれに隣接する2つのノードのTransverse Leakageの平均値を用いて、着目ノード内の分布を2次式で近似するものである。例えば、(2.3.4)式で定義した  $f_1(x)$  と  $f_2(x)$  を使用して、

$$\bar{L}_{g,y}^k(x) \approx \rho_{g,y,x}^k \equiv \rho_{0g,y,x}^k + \rho_{1g,y,x}^k f_1(x) + \rho_{2g,y,x}^k f_2(x) \quad (2.6.11)$$

$\rho_{0g,y,x}^k, \rho_{1g,y,x}^k, \rho_{2g,y,x}^k$ は、k番目ノードにおける第g群のy方向Transverse Leakageのx方向

分布を定める係数を意味する。ここで、下図に示すようにk番目ノードに隣接する2つのノードを考える。



$\rho_{0g,y,x}^k$ ,  $\rho_{1g,y,x}^k$ ,  $\rho_{2g,y,x}^k$ を定めるため、以下の3つの式を使用する。

$$\text{ノード } k \text{について、 } \frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} \rho_{g,y}^k(x) dx = L_{g,y}^k \quad (2.2.21 \text{式参照}) \quad (2.6.12)$$

また、ノードkに対する2次式を両側ノードにも拡張することにより、

$$\text{ノード } k-1 \text{に対して、 } \frac{1}{\Delta_x^{k-1}} \int_{-\Delta_x^{k/2}-\Delta_x^{k-1}}^{-\Delta_x^{k/2}} \rho_{g,y}^k(x) dx = L_{g,y}^{k-1} \quad (2.6.13)$$

$$\text{ノード } k+1 \text{に対して、 } \frac{1}{\Delta_x^{k+1}} \int_{\Delta_x^{k/2}}^{\Delta_x^{k/2}+\Delta_x^{k+1}} \rho_{g,y}^k(x) dx = L_{g,y}^{k+1} \quad (2.6.14)$$

先ず、(2.6.12)式に(2.6.11)式を代入すると(2.3.6)式により直ちに $\rho_{0g,y,x}^k$ が定まる。

$$\rho_{0g,y,x}^k = L_{g,y}^k \quad (2.6.15)$$

次に、(2.6.11)式を(2.6.13)式及び(2.6.14)式に代入して連立方程式を解けば、 $\rho_{1g,y,x}^k$ と $\rho_{2g,y,x}^k$ が以下のように定まる。

$$\rho_{1g,y,x}^k = \frac{(\eta^k+1)(2\eta^k+1)L_{g,y}^{k+1} - (\eta^k+1)(2\eta^k+1)L_{g,y}^{k-1} + \{(\eta^k+1)(2\eta^k+1) - (\eta^k+1)(2\eta^k+1)\}L_{g,y}^k}{(\eta^k+1)(\eta^k+1)(\eta^k+1)} \quad (2.6.16)$$

$$\rho_{2g,y,x} = \frac{(\eta^k+1)L_{g,y}^{k+1} + (\eta^k+1)L_{g,y}^{k-1} - (\eta^k+\eta^k+2)L_{g,y}^k}{(\eta^k+1)(\eta^k+1)(\eta^k+\eta^k+1)} \quad (2.6.17)$$

$$\text{ただし、 } \eta^k \equiv \frac{\Delta_x^{k-1}}{\Delta_x^k}, \quad \eta^k_+ \equiv \frac{\Delta_x^{k+1}}{\Delta_x^k} \quad (2.6.18)$$

(2.6.16)式及び(2.6.17)式中に現れる  $L_{g,y}^{k-1}$ ,  $L_{g,y}^k$ ,  $L_{g,y}^{k+1}$  は、以下のように部分中性子流で表すことができ、応答行列式を使用した内部反復計算において計算することができる。

$$L_{g,y}^{k-1} = J_{g,y+}^{k-1} - J_{g,y-}^{k-1} = J_{g,y+}^{\text{out},k-1} - J_{g,y+}^{\text{in},k-1} - J_{g,y-}^{\text{in},k-1} + J_{g,y-}^{\text{out},k-1} \quad (2.6.19a)$$

$$L_{g,y}^k = J_{g,y+}^k - J_{g,y-}^k = J_{g,y+}^{\text{out},k} - J_{g,y+}^{\text{in},k} - J_{g,y-}^{\text{in},k} + J_{g,y-}^{\text{out},k} \quad (2.6.19b)$$

$$L_{g,y}^{k+1} = J_{g,y+}^{k+1} - J_{g,y-}^{k+1} = J_{g,y+}^{\text{out},k+1} - J_{g,y+}^{\text{in},k+1} - J_{g,y-}^{\text{in},k+1} + J_{g,y-}^{\text{out},k+1} \quad (2.6.19c)$$

ノード内の分布形状が得られたので、次に(2.6.2a)式の  $L_{g,y,xn}^k$  を求める。先ず、 $n=1$  の場合、

$$\begin{aligned} L_{g,y,x1}^k &= \frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} w_1(x) \bar{L}_{g,y}^k(x) dx \\ &= \frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} f_1(x) \{ \rho_{0g,y,x}^k + \rho_{1g,y,x}^k f_1(x) + \rho_{2g,y,x}^k f_2(x) \} dx \\ &= \frac{\rho_{1g,y,x}^k}{12} \quad (2.3.7 \text{式参照}) \end{aligned} \quad (2.6.20)$$

$n=2$  の場合も同様に、

$$L_{g,y,x2}^k = \frac{\rho_{2g,y,x}^k}{20} \quad (2.6.21)$$

$L_{g,z,x1}^k$  と  $L_{g,z,x2}^k$  についても同様にして得られ、(2.6.1)式により、 $x$  方向側面からの漏洩に対する 1 次及び 2 次のモーメントが以下のように与えられる。

$$L_{g,x1}^k = \frac{1}{\Delta_y^k} L_{g,y,x1}^k + \frac{1}{\Delta_z^k} L_{g,z,x1}^k = \frac{\rho_{1g,y,x}^k}{12\Delta_y^k} + \frac{\rho_{1g,z,x}^k}{12\Delta_z^k} \quad (2.6.22a)$$

$$L_{g,x2}^k = \frac{1}{\Delta_y^k} L_{g,y,x2}^k + \frac{1}{\Delta_z^k} L_{g,z,x2}^k = \frac{\rho_{2g,y,x}^k}{20\Delta_y^k} + \frac{\rho_{2g,z,x}^k}{20\Delta_z^k} \quad (2.6.22b)$$

y方向についても、これまでと全く同様にして、

$$L_{g,y1}^k = \frac{1}{\Delta_z^k} L_{g,z,y1}^k + \frac{1}{\Delta_x^k} L_{g,x,y1}^k = \frac{\rho_{1g,z,y}^k}{12\Delta_z^k} + \frac{\rho_{1g,x,y}^k}{12\Delta_x^k} \quad (2.6.23a)$$

$$L_{g,y2}^k = \frac{1}{\Delta_z^k} L_{g,z,y2}^k + \frac{1}{\Delta_x^k} L_{g,x,y2}^k = \frac{\rho_{2g,z,y}^k}{20\Delta_z^k} + \frac{\rho_{2g,x,y}^k}{20\Delta_x^k} \quad (2.6.23b)$$

z方向については、

$$L_{g,z1}^k = \frac{1}{\Delta_x^k} L_{g,x,z1}^k + \frac{1}{\Delta_y^k} L_{g,y,z1}^k = \frac{\rho_{1g,x,z}^k}{12\Delta_x^k} + \frac{\rho_{1g,y,z}^k}{12\Delta_y^k} \quad (2.6.24a)$$

$$L_{g,z2}^k = \frac{1}{\Delta_x^k} L_{g,x,z2}^k + \frac{1}{\Delta_y^k} L_{g,y,z2}^k = \frac{\rho_{2g,x,z}^k}{20\Delta_x^k} + \frac{\rho_{2g,y,z}^k}{20\Delta_y^k} \quad (2.6.24b)$$

(2.6.22)式～(2.6.24)式により、(2.5.27)式中のベクトル  $\mathbf{L}_g^k$  を与えることができる。なお、(2.6.11)式において  $\rho_{1g,y,x}^k$ 、 $\rho_{2g,y,x}^k$  を共にゼロとすると第2.6.1節で扱った平坦近似に一致する。すなわち、(2.6.22)式～(2.6.24)式は全てゼロとなり(2.5.27)式の  $\mathbf{L}_g^k$  は不要となる。

## 2.7 不連続因子の導入

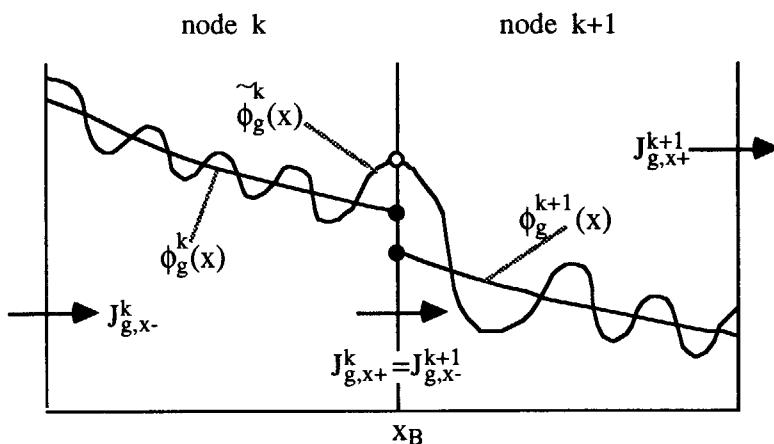
### 2.7.1 不連続因子の定義

粗メッシュを扱うノード法は必然的に均質化断面積を使用することになる。例えば、1つの燃料集合体断面を1ノードに割り当てて炉心計算をする場合、あらかじめ集合体内で均質化された断面積を用意する。多くの場合、この均質化断面積を作成するのに、非均質な集合体の無限格子モデルを用いて、これにより得られる非均質中性子束分布で反応率を保存するように荷重平均した均質化断面積を作成する。すなわち、

$$\Sigma_g^{\text{homo}} = \frac{\int \Sigma_g^{\text{hetero}}(\mathbf{r}) \phi_g^{\text{hetero}}(\mathbf{r}) d\mathbf{r}}{\phi_g^{\text{homo}} V^{\text{homo}}} \quad (2.7.1a)$$

$$\phi_g^{\text{homo}} V^{\text{homo}} = \int \phi_g^{\text{hetero}}(\mathbf{r}) d\mathbf{r} \quad (2.7.1b)$$

このようにして得られた均質化断面積を使用して炉心計算を行っても、全炉心を非均質に扱う計算と完全に一致する解は得られない。むろん、均質化断面積作成の際に無限格子モデルで得られる中性子束で荷重平均することも1つの理由ではあるが、それだけではない。仮に全炉心計算を非均質のまま扱い、これにより得られる全炉心非均質中性子束を使用して、(2.7.1)式により燃料集合体毎に均質化断面積を作成したとする。このようにして得られた均質化断面積を使用し、同じコードで再度全炉心計算を行っても、やはり全炉心非均質計算の結果を完全に再現することはできない<sup>16)</sup>。例えば、下図のような1次元問題を考える。



k番目ノードとk+1番目ノードの内部には、非均質性(断面積分布)があるものとする。この非均質性を考慮して体系計算をすれば、各ノード内の非均質中性子束分布 $\tilde{\phi}_g^k(x)$ 及び $\tilde{\phi}_g^{k+1}(x)$

が得られる。同時にノード境界における正味中性子流が得られる。ここで得られた非均質中性子束を使用して、(2.7.1)式によりノードk及びノードk+1の均質化断面積を作成して再び体系計算を行っても、非均質計算の場合と反応率と漏れを同時に保存するような結果は得られない。次に2つのノードを分離して考え、非均質計算で得られたノード境界の正味中性子流を境界条件として課し均質化断面積を使用して各ノード毎に計算すると、同図に示すような均質化中性子束の分布 $\tilde{\phi}_g(x)$ 、 $\tilde{\phi}_g^{k+1}(x)$ を得る。この時、非均質計算と各ノードの均質計算では反応率と漏れを保存する結果が得られるが、ノード境界 $x=x_B$ で均質化中性子束は不連続となる。逆に言えば、均質化断面積を使用する計算では、均質化媒質境界において非均質中性子束の連続性を考慮すべきであり、結果として均質化中性子束の不連続性を許容する必要がある。そこで、均質化媒質境界において不連続因子を次のように定義する。

$$f_{g,x+}^k \equiv \frac{\tilde{\phi}_g(x=x_B)}{\phi_g^k(x=x_B)}, \quad f_{g,x-}^{k+1} \equiv \frac{\tilde{\phi}_g(x=x_B)}{\phi_g^{k+1}(x=x_B)} \quad (2.7.2)$$

ここで、 $f_{g,x+}^k$ と $f_{g,x-}^{k+1}$ はそれぞれk番目ノードの右側境界とk+1番目ノードの左側境界における第g群の不連続因子である。境界における非均質中性子束は連続であり、

$$f_{g,x+}^k \phi_g^k(x=x_B) = f_{g,x-}^{k+1} \phi_g^{k+1}(x=x_B) = \tilde{\phi}_g(x=x_B) \quad (2.7.3)$$

一般に、3次元X-Y-Z体系の問題では、1つのノードが持つ6つの表面に対し、6つの不連続因子が与えられることになる。この時、(2.7.2)式の中性子束は表面平均中性子束を意味することになる。

(2.7.2)式に示されるように、不連続因子は非均質表面平均中性子束の均質化表面平均中性子束に対する比として定義される。しかし、非均質中性子束、及びこれにより得られる均質化断面積と不連続因子は、全炉心を非均質のまま解いた時にのみ正確な値として得られ、それが得られているのならば均質化定数を作成して炉心計算をする意味はない。結局、均質化断面積と不連続因子は、燃料集合体の2次元無限格子モデルを使用した非均質計算により得られる中性子束で代用して作成することになる。集合体計算により得られる近似的な不連続因子は集合体不連続因子(Assembly Discontinuity Factor)と呼ばれ、以下のように与えられる。

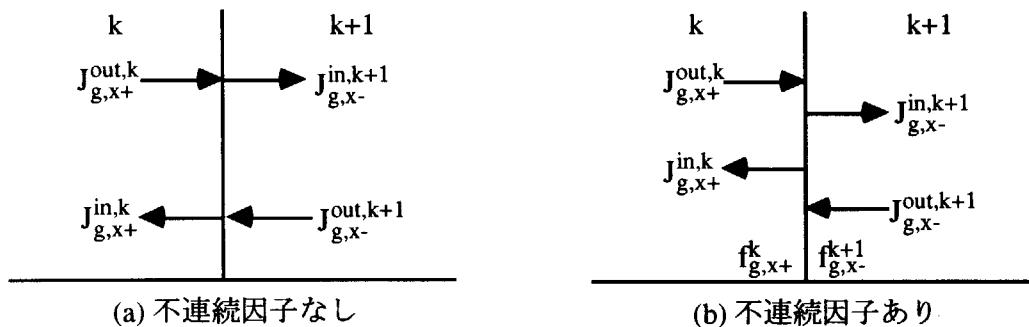
$$\begin{aligned} \text{集合体不連続因子} &= \frac{\text{着目する集合体表面における面平均非均質中性子束}}{\text{着目する集合体表面における面平均均質化中性子束}} \\ &= \frac{\text{着目する集合体表面における面平均非均質中性子束}}{\text{集合体平均中性子束}} \end{aligned} \quad (2.7.4)$$

不連続因子導入の経緯やその効果については、K. S. Smithにより詳しくレビューされているので<sup>16)</sup>、これ以上は詳しく述べない。以下ではノード法コードにおいて、具体的に不連続

因子を使用する方法について述べる。

## 2.7.2 不連続因子を用いた部分中性子流の更新

多項式展開法に限らずほとんどの近代ノード法は、表面平均パラメータを利用するため、不連続因子の導入は容易である。不連続因子を導入しない場合には、下図に示すように部分中性子流は隣接ノード間で連続であるとして扱われる。



不連続因子を扱わない場合には、ノード $k$ とこれに隣接するノード $k+1$ の表面平均部分中性子流の間には以下のようないくつかの関係が成立する。

$$J_{g,x-}^{in,k+1} = J_{g,x+}^{out,k} \quad (2.7.5a)$$

$$J_{g,x+}^{in,k} = J_{g,x-}^{out,k+1} \quad (2.7.5b)$$

これらの式は、(2.1.16)式と(2.1.17)式から、ノード境界における表面平均中性子束と表面平均正味中性子流の連続性を保証している。一方、不連続因子を導入する場合には、表面平均正味中性子流を連続としながらも、表面平均中性子束は不連続とし、代わりに非均質表面平均中性子束を連続にする。すなわち、

$$J_{g,x+}^k = J_{g,x-}^{k+1} \quad (2.7.6)$$

$$f_{g,x+}^k \phi_{g,x+}^k = f_{g,x-}^{k+1} \phi_{g,x-}^{k+1} \quad (2.7.7)$$

これらを、(2.1.16)式と(2.1.17)式により、表面平均部分中性子流で表すと、

$$J_{g,x+}^{out,k} - J_{g,x+}^{in,k} = J_{g,x-}^{in,k+1} - J_{g,x-}^{out,k+1} \quad (2.7.8)$$

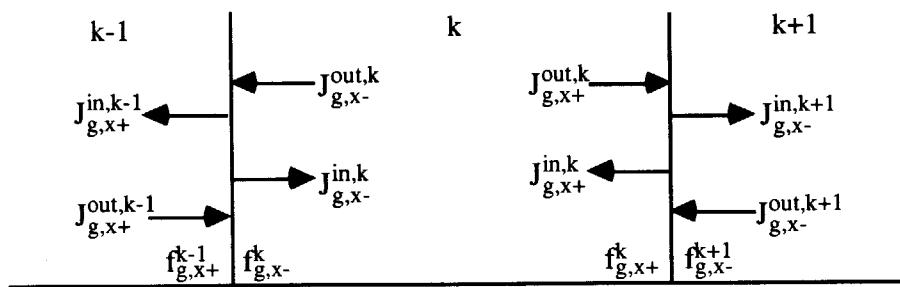
$$2f_{g,x+}^k (J_{g,x+}^{out,k} + J_{g,x+}^{in,k}) = 2f_{g,x-}^{k+1} (J_{g,x-}^{in,k+1} + J_{g,x-}^{out,k+1}) \quad (2.7.9)$$

(2.7.8)式と(2.7.9)式から $J_{g,x-}^{in,k+1}$ を消去して式を整理すると以下の式を得る。

$$J_{g,x+}^{in,k} = \left( \frac{1}{1 - \alpha_{g,x+}^k} \right) (J_{g,x-}^{out,k+1} - \alpha_{g,x+}^k J_{g,x+}^{out,k}) \quad (2.7.10)$$

$$\text{ただし、 } \alpha_{g,x+}^k \equiv \frac{1}{2} \left( 1 - \frac{f_{g,x+}^k}{f_{g,x-}^{k+1}} \right) \quad (2.7.11)$$

ノード境界における不連続因子が全て1.0(均質化表面平均中性子束=非均質表面平均中性子束)、あるいは不連続因子の比が1.0( $\alpha_{g,x+}^k = 0$ )の場合には、 $J_{g,x+}^{in,k} = J_{g,x-}^{out,k+1}$ (よって、2.7.8式から $J_{g,x+}^{out,k} = J_{g,x-}^{in,k+1}$ )となり、表面平均部分中性子流はノード境界で連続となり、不連続因子を使用しない場合の扱いと一致する。



次章で述べるチェックカードスウェーブをする場合には、 $J_g^{in,k}$ が与えられ時、応答行列式(2.5.27)により $J_g^{out,k}$ を得る。次に $J_g^{in,k}$ と $J_g^{out,k}$ から隣接ノードの $J_g^{in,k\pm 1}$ を更新することになる。したがって、(2.7.10)式を隣接ノードの入射部分中性流を更新する式に並び替えておくと便利である。(2.7.8)式を使って、(2.7.10)式から $J_{g,x-}^{out,k+1}$ を消去すると、

$$J_{g,x-}^{in,k+1} = (1 - \alpha_{g,x-}^k) J_{g,x-}^{out,k} - \alpha_{g,x-}^k J_{g,x-}^{in,k} \quad (2.7.12)$$

k-1とk番目ノードの境界についても同様にして

$$J_{g,x+}^{in,k-1} = (1 - \alpha_{g,x+}^k) J_{g,x+}^{out,k} - \alpha_{g,x+}^k J_{g,x+}^{in,k} \quad (2.7.13)$$

$$\text{ただし、 } \alpha_{g,x-}^k \equiv \frac{1}{2} \left( 1 - \frac{f_{g,x-}^k}{f_{g,x+}^{k-1}} \right) \quad (2.7.14)$$

y方向及びz方向についても、(2.7.12)式と(2.7.13)式と同様な関係式が得られる。不連続因子を使用する場合には、(2.5.27)式により $J_g^{out,k}$ を更新した後、(2.7.12)式及び(2.7.13)式により、隣接する6つのノードの $J_g^{in,k\pm 1}$ を更新する。これを全てのノードについて繰り返せば、内部反復計算となる。

## 2.8 境界条件

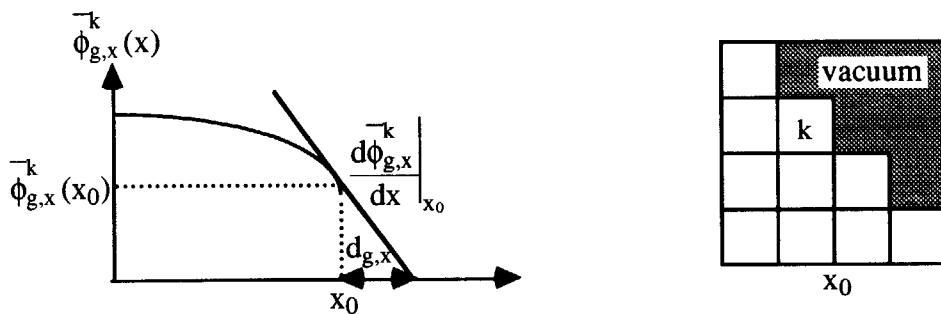
境界条件の処理に関しては、Lawrenceのレビュー論文などにはほとんど記述されていないが、実際のプログラミングにおいては最も頭を悩ませる部分であろう。

これまで、着目方向において両隣にノードが存在することを前提として議論してきた。しかし、外部境界に隣接するノードにおいては、境界条件に応じた例外処理をする必要がある。先ず、多項式展開ノード法では体系スウェープに部分中性子流を扱うため、各種境界条件を部分中性子流を使用して表現する必要がある。また、Quadratic Transverse Leakage Approximation (2.6.3節)では、隣接ノードにおけるTransverse Leakageを利用する。その際、着目するノードが境界に隣接している場合に例外処理が必要となる。

### 2.8.1 真空境界条件(外挿境界条件)

真空境界条件と言う場合、1)表面入射部分中性子流をゼロとする場合と、2)外挿距離において中性子束をゼロとする扱いがある。後者は、外挿境界条件(extrapolated boundary condition)あるいは対数境界条件と呼ばれる。外挿境界条件は、後に述べる外挿定数の与え方により幾つかの境界条件を表現することが可能であり、表面入射部分中性子流をゼロとする場合もこれに包含される。したがって、以下ではより一般的な外挿境界条件の扱いについて説明する。

外挿境界条件は、境界点における中性子束分布の接線を外挿し、指定する境界面からの距離(外挿距離)において外挿値がゼロとなるように境界中性子束の微係数を定めるものである。



$x$ 方向の外挿距離 $d_{g,x}$ は上図から、(2.8.1)式により与えられる。

$$\frac{1}{d_{g,x}} = \frac{-1}{\phi_{g,x+}^k} \left. \frac{d\phi_{g,x}^k}{dx} \right|_{x_0} \quad (2.8.1)$$

ただし、 $\phi_{g,x+}^k = \phi_{g,x}(x=x_0)$ である。外挿距離は群依存となるため、一般に以下のように定義する外挿定数が使用される。

$$C_g \equiv \frac{-D_{g,x}}{\phi_{g,x+}^k} \left. \frac{d\phi_{g,x}(x)}{dx} \right|_{x_0} \quad (2.8.2)$$

(2.8.1)式と(2.8.2)式より、

$$C_g = \frac{D_{g,x}}{d_{g,x}} = \frac{1}{d_{g,x}} \frac{1}{3\Sigma_{tr,g}} = \frac{\lambda_{tr,g}}{3d_{g,x}} \quad (2.8.3)$$

の関係がある。ここで、Milne問題<sup>17)</sup>の外挿距離  $d_{g,x} \approx 0.7105\lambda_{tr,g}$ を使用すると、外挿定数  $C_g$ は群に依存せずに0.4692となる。よって、外表面に対しては、一般に全群に対して、

$$C_g = 0.4692 \quad (2.8.4)$$

とすることが多い。ちなみに、外挿定数に十分大きな値を入力すると、 $d_g = \frac{D_g}{C_g} \approx 0.0$ となり、外表面で中性子束ゼロを仮定した計算となる。また、外挿定数をゼロとすれば、(2.8.2)式から境界表面で中性子束の勾配はゼロとなり、これは反射境界条件に対応する。ここで、(2.8.2)式を表面平均部分中性子流による表現に変換する。不連続因子を使用する場合を考え、(2.1.17)式を(2.7.2)式により補正し、

$$f_{g,x+}^k \phi_{g,x+}^k = 2(J_{g,x+}^{out,k} + J_{g,x+}^{in,k}) \quad (2.8.5)$$

また、(2.3.13)式から

$$-D_{g,x} \left. \frac{d\phi_{g,x}(x)}{dx} \right|_{x_0} = J_{g,x+}^{out,k} - J_{g,x+}^{in,k} \quad (2.8.6)$$

これらの式を(2.8.2)式に代入して整理すると、

$$J_{g,x+}^{in,k} = \begin{cases} \frac{1-2C_g f_{g,x+}^k}{1+2C_g f_{g,x+}^k} J_{g,x+}^{out,k} & [x+面が真空の場合] \end{cases} \quad (2.8.7a)$$

$x$ -面が真空の場合にも同様にして、

$$J_{g,x-}^{in,k} = \begin{cases} \frac{1-2C_g f_{g,x-}^k}{1+2C_g f_{g,x-}^k} J_{g,x-}^{out,k} & [x-面が真空の場合] \end{cases} \quad (2.8.7b)$$

不連続因子を使用しない場合には、(2.8.7)式は、

$$J_{g,x\pm}^{in,k} = \begin{cases} 1-2C_g \\ 1+2C_g \end{cases} J_{g,x\pm}^{out,k} \quad (2.8.8)$$

となり、 $C_g$ の値の与え方で、反射、アルベドなどの境界条件も記述することができる。この時、 $C_g$ の取りうる値の範囲は、部分中性子流が正の値を持つことから、 $(-\frac{1}{2} < C_g \leq \frac{1}{2})$ となる(負の場合は增幅反射)。ただし、 $C_g \rightarrow \infty$ で(2.8.8)式は $J_{g,x\pm}^{in,k} + J_{g,x\pm}^{out,k} = 0$ となり、(2.8.5)式から表面平均中性子束をゼロとする境界条件となる。結局、真空境界に接するノードにおいては、隣接ノードの部分中性子流を(2.7.12)式または(2.7.13)式で更新する代わりに、境界面からの入射部分中性子流を(2.8.7)式を使用して更新することになる。

一般に、真空境界条件は部分中性子流によりもっと直接に表現することができる。

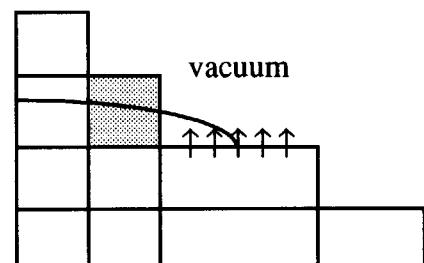
$$J_{g,x+}^{in,k} = 0 \quad [x+面が真空の場合] \quad (2.8.9a)$$

$$J_{g,x-}^{in,k} = 0 \quad [x-面が真空の場合] \quad (2.8.9b)$$

これは、不連続因子を使用しない(2.8.8)式において $C_g = 0.5$ とすることに相当し、(2.8.4)式と矛盾する。拡散理論の場合には、一見正確な条件に見える(2.8.9)式も近似にすぎず、結局どちらの境界条件も真空境界から平均自由行程の数倍以内の距離では正確とは言えない。実際問題の多くは反射体付き原子炉を扱うため、これらの差異は実用上問題にはならない。

3つのノードに対するTransverse Leakageのノード平均値を利用するQuadratic Transverse Leakage Approximation (以下QTLA)は、真空境界に隣接するノードには適用できない。もともと、QTLAは拡散理論などの理論的な根拠に基づく近似ではなく、物理量の比較的緩やかな連続性を期待しての近似であるため、適切な対処法は見あたらない。中性子束の外挿境界条件に倣って外挿距離でTransverse Leakageをゼロにするようにフィッティングする方法はもっともらしいが、右の図のようにぎざぎざ状の真空境界を考えると必ずしも適切とは言いがたい。結局、Transverse Leakageのノード内分布は積分量の計算精度に関して直接影響を及ぼすものではないことと、QTLAを適用できないのはほとんどの場合が反射体領域であることから、便宜的な方法が採られることが多い。MOSRA-Lightコードの場合、反対側にノードが存在する場合には1次式近似を、両側が真空境界の場合にはバックリング近似(2.6.2節参照)を使用している。

1次式近似の場合は、例えば $x+$ 面のみが真空境界に接している場合、(2.6.11)式において2次の係数 $\rho_{2g,y,x}^k$ をゼロとし、 $\rho_{0g,y,x}^k$ と $\rho_{1g,y,x}^k$ を $k-1$ 番目ノードと $k$ 番目ノードに対するノード平均Transverse Leakage( $L_{g,y}^k$ 、 $L_{g,y}^{k-1}$ )から求める(2.6.12式と2.6.13式参照)。 $\rho_{0g,z,x}^k$ 、 $\rho_{1g,z,x}^k$ 、 $\rho_{2g,z,x}^k$ についても同様であり、結局以下のような係数が得られる。



$$\left\{ \begin{array}{l} \rho_{0g,y,x}^k = \rho_{0g,z,x}^k = L_{g,x}^k \\ \rho_{1g,y,x}^k = \frac{2(L_{g,y}^k - L_{g,y}^{k-1})}{\eta_+^{k+1}} \\ \rho_{1g,z,x}^k = \frac{2(L_{g,z}^k - L_{g,z}^{k-1})}{\eta_+^{k+1}} \\ \rho_{2g,y,x}^k = \rho_{2g,z,x}^k = 0 \end{array} \right. \quad [x+\text{面が真空の場合}] \quad (2.8.10a)$$

$x$ -面のみが真空境界に接している場合も同様にして、

$$\left\{ \begin{array}{l} \rho_{0g,y,x}^k = \rho_{0g,z,x}^k = L_{g,x}^k \\ \rho_{1g,y,x}^k = \frac{2(L_{g,y}^{k+1} - L_{g,y}^k)}{\eta_+^{k+1}} \\ \rho_{1g,z,x}^k = \frac{2(L_{g,z}^{k+1} - L_{g,z}^k)}{\eta_+^{k+1}} \\ \rho_{2g,y,x}^k = \rho_{2g,z,x}^k = 0 \end{array} \right. \quad [x-\text{面が真空の場合}] \quad (2.8.10b)$$

これらを(2.6.22)式に適用する。

## 2.8.2 反射境界条件

反射境界面では、たとえ不連続因子が与えられていたとしても正味中性子流はゼロである。よって、

$$J_{g,x+}^{in,k} = J_{g,x+}^{out,k} \quad [x+\text{面が反射の場合}] \quad (2.8.11a)$$

$$J_{g,x-}^{in,k} = J_{g,x-}^{out,k} \quad [x-\text{面が反射の場合}] \quad (2.8.11b)$$

これらを内部反復計算において、(2.7.12)式または(2.7.13)式の代わりに使用する。(2.8.11)式は、(2.8.8)式で  $C_g = 0.5$  とすることに等しいが、不連続因子を使用した(2.8.7a)式では  $C_g = 0.5$  としても(2.8.11)式と一致しないことに注意する必要がある。

QTLAの対応としては、全く同じノードが隣に対称に存在するものとして処理すれば良いから話は簡単である。例えば、 $x+$ 面が反射の場合には、(2.6.16)式及び(2.6.17)式により、 $\rho_{1g,y,x}^k$ 、 $\rho_{2g,y,x}^k$ 、 $\rho_{1g,z,x}^k$ 、 $\rho_{2g,z,x}^k$ を求める際、

$$\left\{ \begin{array}{l} L_{g,y}^{k+1} \Rightarrow L_{g,y}^k \\ L_{g,z}^{k+1} \Rightarrow L_{g,z}^k \\ \eta_+^k \Rightarrow 1 \end{array} \right. \quad [x+\text{面が反射の場合}] \quad (2.8.12a)$$

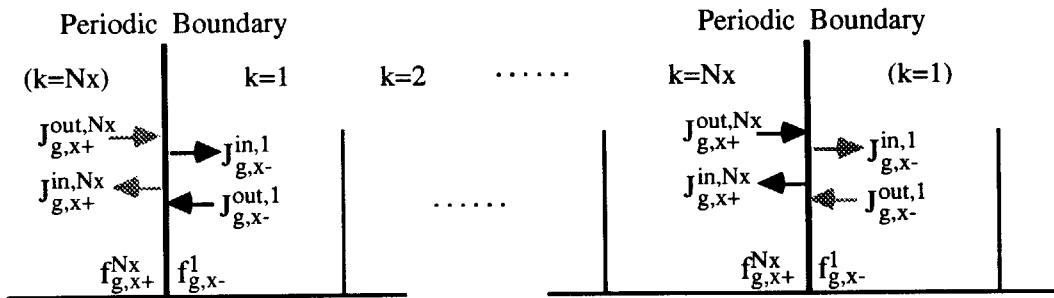
と置き換えるべき。x-面が反射の場合にも同様に、

$$\begin{cases} L_{g,y}^{k-1} \Rightarrow L_{g,y}^k \\ L_{g,z}^{k-1} \Rightarrow L_{g,z}^k & [x\text{-面が反射の場合}] \\ \eta^k \Rightarrow 1 \end{cases} \quad (2.8.12b)$$

このようにして得られる  $\rho_{1g,y,x}^k$ ,  $\rho_{2g,y,x}^k$ ,  $\rho_{1g,z,x}^k$ ,  $\rho_{2g,z,x}^k$  を(2.6.22)式に適用する。

### 2.8.3 周期境界条件

x方向に  $N_x$  個のノード分割がされており、x方向境界条件が周期境界条件であるとする。この場合、内部反復計算において、 $k=1$  と  $k=N_x$  のノードに対しては、(2.7.12)式または(2.7.13)式の代わりに次のような処理をする。



$k=N_x$  のノードに着目して、(2.7.12)式の  $k+1$  ノードを1番目ノードと見なし、

$$J_{g,x-}^{in,1} = (1 - \alpha_{g,x+}^k) J_{g,x+}^{out,k} - \alpha_{g,x+}^k J_{g,x+}^{in,k} \quad [k=N_x \text{ の場合}] \quad (2.8.13a)$$

$$\text{ただし、} \alpha_{g,x+}^k \equiv \frac{1}{2} \left( 1 - \frac{f_{g,x+}^k}{f_{g,x-}^1} \right) \quad (2.8.13b)$$

$k=1$  のノードに着目して、(2.7.13)式の  $k-1$  ノードを  $N_x$  番目ノードと見なし、

$$J_{g,x+}^{in,Nx} = (1 - \alpha_{g,x-}^k) J_{g,x-}^{out,k} - \alpha_{g,x-}^k J_{g,x-}^{in,k} \quad [k=1 \text{ の場合}] \quad (2.8.14a)$$

$$\text{ただし、} \alpha_{g,x-}^k \equiv \frac{1}{2} \left( 1 - \frac{f_{g,x-}^k}{f_{g,x+}^{N_x}} \right) \quad (2.8.14b)$$

QTLAの対応としては、例えば、x+面が周期境界の場合には、(2.6.16)式及び(2.6.17)式により  $\rho_{1g,y,x}^k$ ,  $\rho_{2g,y,x}^k$ ,  $\rho_{1g,z,x}^k$ ,  $\rho_{2g,z,x}^k$  を求める際、

$$\begin{cases} L_{g,y}^{k+1} \Rightarrow L_{g,y}^1 \\ L_{g,z}^{k+1} \Rightarrow L_{g,z}^1 & [k=Nx \text{の場合}] \\ \eta_+^k \Rightarrow \frac{\Delta_x^1}{\Delta_x^k} \end{cases} \quad (2.8.15a)$$

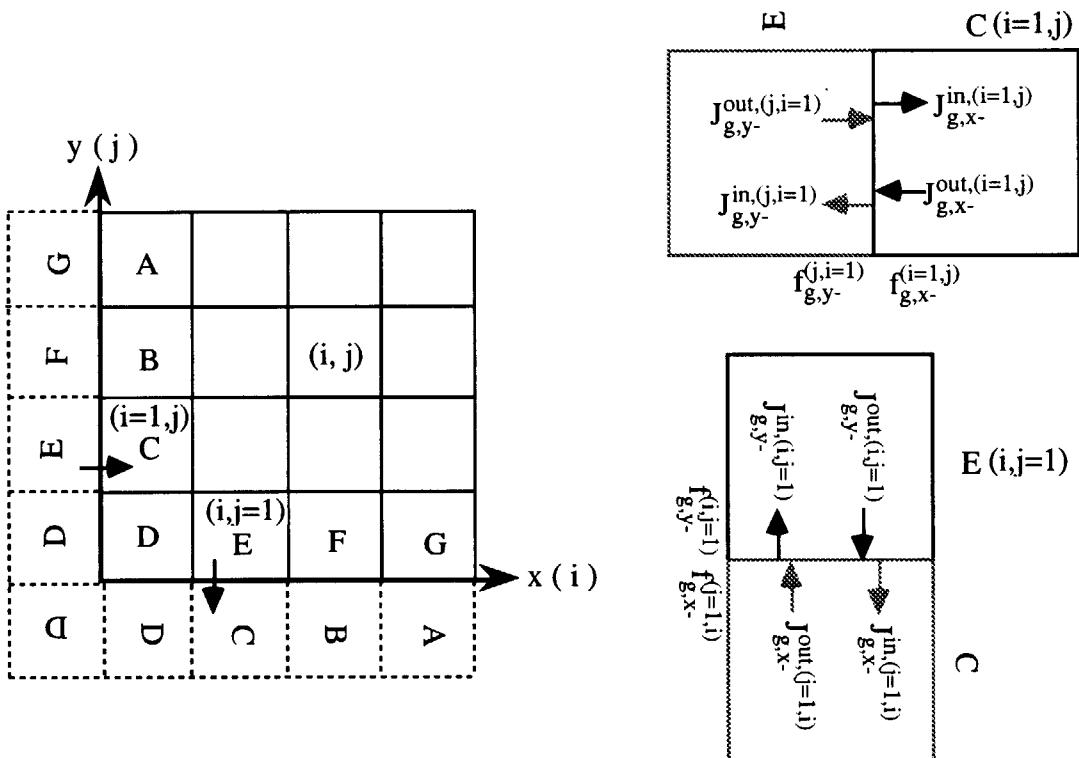
と置き換えれば良い。x-面が周期境界の場合にも同様に、

$$\begin{cases} L_{g,y}^{k-1} \Rightarrow L_{g,y}^{Nx} \\ L_{g,z}^{k-1} \Rightarrow L_{g,z}^{Nx} & [k=1 \text{の場合}] \\ \eta_-^k \Rightarrow \frac{\Delta_x^{Nx}}{\Delta_x^k} \end{cases} \quad (2.8.15b)$$

このようにして得られる  $\rho_{1g,y,x}^k$ 、 $\rho_{2g,y,x}^k$ 、 $\rho_{1g,z,x}^k$ 、 $\rho_{2g,z,x}^k$  を(2.6.22)式に適用する。

#### 2.8.4 90度回転境界条件

90度回転対称は炉心構成としてしばしば見られるが、回転境界条件が扱えない場合には、全炉心で計算する必要に迫られる。したがって、計算時間節約の観点からもぜひ導入すべき境界条件である。以下の議論では、z軸方向は関係無いため省略し、x-y座標系に対応してノードの水平方向位置を(i, j)で表わすことにする。



x方向のノード分割数とy方向のノード分割数をNx及びNyとすると、90度回転境界条件が適用できるのは、以下の条件が成立する場合に限られる。

$$\bullet \quad Nx = Ny \equiv N \quad (2.8.16)$$

$$\bullet \quad \Delta_x^i = \Delta_y^i \quad (i=1,2,\dots,N) \quad (2.8.17)$$

先ず、上図のノードCのように、x-面に接するノード(i=1, j)に着目する。この場合、(2.7.13)式及び(2.7.14)式に対応する式は以下のようになる。

$$J_{g,x^-}^{in,(j,i=1)} = (1 - \alpha_{g,x^-}^{(i=1,j)}) J_{g,x^-}^{out,(i=1,j)} - \alpha_{g,x^-}^{(i=1,j)} J_{g,x^-}^{in,(i=1,j)} \quad (2.8.18)$$

$$\alpha_{g,x^-}^{(i=1,j)} \equiv \frac{1}{2} \left( 1 - \frac{f_{g,x^-}^{(i=1,j)}}{f_{g,y^-}^{(j,i=1)}} \right) \quad (2.8.19)$$

(2.8.18)式は、体系スウェーブ(内部反復計算)において、x-面に接するノード(i=1,j)の部分中性子流を使用して、y-面に接するノード(j,i=1)の入射部分中性子流を更新する式である。

一方、ノードEのようにy-面に接するノード(i, j=1)に着目すると次式を得る。

$$J_{g,y^-}^{in,(j=1,i)} = (1 - \alpha_{g,y^-}^{(i,j=1)}) J_{g,y^-}^{out,(i,j=1)} - \alpha_{g,y^-}^{(i,j=1)} J_{g,y^-}^{in,(i,j=1)} \quad (2.8.20)$$

$$\alpha_{g,y^-}^{(i,j=1)} \equiv \frac{1}{2} \left( 1 - \frac{f_{g,y^-}^{(i,j=1)}}{f_{g,x^-}^{(j=1,i)}} \right) \quad (2.8.21)$$

これは、y-面に接するノード(i,j=1)の部分中性子流を使用して、x-面に接するノード(j=1,i)の入射部分中性子流を更新する式である。

次にQTLAの対応を考える。(2.6.16)式と(2.6.17)式を(i,j)を使用して2次元的に表現する。(i>1)の場合には、

$$\begin{aligned} \rho_{1g,y,x}^{(i,j)} &= \frac{(\eta_x^{(i-1,j)}+1)(2\eta_x^{(i-1,j)}+1)L_{g,y}^{(i+1,j)}}{(\eta_x^{(i+1,j)}+1)(\eta_x^{(i-1,j)}+1)(\eta_x^{(i+1,j)}+\eta_x^{(i-1,j)}+1)} \\ &\quad - \frac{(\eta_x^{(i+1,j)}+1)(2\eta_x^{(i+1,j)}+1)L_{g,y}^{(i-1,j)}}{(\eta_x^{(i+1,j)}+1)(\eta_x^{(i-1,j)}+1)(\eta_x^{(i+1,j)}+\eta_x^{(i-1,j)}+1)} \\ &\quad + \frac{\{(\eta_x^{(i+1,j)}+1)(2\eta_x^{(i+1,j)}+1)-(\eta_x^{(i-1,j)}+1)(2\eta_x^{(i-1,j)}+1)\}L_{g,y}^{(i,j)}}{(\eta_x^{(i+1,j)}+1)(\eta_x^{(i-1,j)}+1)(\eta_x^{(i+1,j)}+\eta_x^{(i-1,j)}+1)}, \quad (i>1) \end{aligned} \quad (2.8.22a)$$

$$\rho_{2g,y,x}^{(i,j)} = \frac{(\eta_x^{(i-1,j)}+1)L_{g,y}^{(i+1,j)} + (\eta_x^{(i+1,j)}+1)L_{g,y}^{(i-1,j)} - (\eta_x^{(i+1,j)}+\eta_x^{(i-1,j)}+2)L_{g,y}^{(i,j)}}{(\eta_x^{(i+1,j)}+1)(\eta_x^{(i-1,j)}+1)(\eta_x^{(i+1,j)}+\eta_x^{(i-1,j)}+1)}, \quad (i>1) \quad (2.8.22b)$$

$$\text{ただし、 } \eta_x^{(i-1,j)} \equiv \frac{\Delta_x^{(i-1,j)}}{\Delta_x^{(i,j)}} \quad (2.8.23a)$$

$$\eta_x^{(i+1,j)} \equiv \frac{\Delta_x^{(i+1,j)}}{\Delta_x^{(i,j)}} \quad (2.8.23b)$$

$x$ -面に接するノード( $i=1, j$ )においては、90度回転対称性を利用し、(2.8.22)式において、次のような置き換えをすればよい。

$$\begin{cases} L_{g,y}^{(i-1,j)} \Rightarrow L_{g,x}^{(j,i=1)} & [i=1 \text{の場合}] \\ \eta_x^{(i-1,j)} \Rightarrow \frac{\Delta_y^{(j,i=1)}}{\Delta_x^{(i,j)}} \end{cases} \quad (2.8.24a)$$

また、 $\rho_{1g,z,x}^{(i=1,j)}$ 及び $\rho_{2g,z,x}^{(i=1,j)}$ については、

$$\begin{cases} L_{g,z}^{(i-1,j)} \Rightarrow L_{g,z}^{(j,i=1)} & [i=1 \text{の場合}] \\ \eta_x^{(i-1,j)} \Rightarrow \frac{\Delta_y^{(j,i=1)}}{\Delta_x^{(i,j)}} \end{cases} \quad (2.8.24b)$$

このようにして得られる $\rho_{1g,y,x}^{(i,j)}$ 、 $\rho_{2g,y,x}^{(i,j)}$ 、 $\rho_{1g,z,x}^{(i,j)}$ 、 $\rho_{2g,z,x}^{(i,j)}$ を(2.6.22)式に適用する。

一方、 $y$ -面に接するノード( $i, j=1$ )においては、 $y$ 方向基準の $x$ 方向及び $z$ 方向へのTransverse Leakageが問題になる。ノードが $y$ -面に接しないとき( $j>1$ )には、 $\rho_{1g,x,y}^{(i,j)}$ と $\rho_{2g,x,y}^{(i,j)}$ は次の式で表される。

$$\begin{aligned} \rho_{1g,x,y}^{(i,j)} &= \frac{(\eta_y^{(i,j-1)}+1)(2\eta_y^{(i,j-1)}+1)L_{g,x}^{(i,j+1)}}{(\eta_y^{(i,j+1)}+1)(\eta_y^{(i,j-1)}+1)(\eta_y^{(i,j+1)}+\eta_y^{(i,j-1)}+1)} \\ &\quad - \frac{(\eta_y^{(i,j+1)}+1)(2\eta_y^{(i,j+1)}+1)L_{g,x}^{(i,j-1)}}{(\eta_y^{(i,j+1)}+1)(\eta_y^{(i,j-1)}+1)(\eta_y^{(i,j+1)}+\eta_y^{(i,j-1)}+1)} \\ &\quad + \frac{\{(\eta_y^{(i,j+1)}+1)(2\eta_y^{(i,j+1)}+1) - (\eta_y^{(i,j-1)}+1)(2\eta_y^{(i,j-1)}+1)\}L_{g,x}^{(i,j)}}{(\eta_y^{(i,j+1)}+1)(\eta_y^{(i,j-1)}+1)(\eta_y^{(i,j+1)}+\eta_y^{(i,j-1)}+1)}, \quad (j>1) \end{aligned} \quad (2.8.25a)$$

$$\rho_{2g,x,y}^{(i,j)} = \frac{(\eta_y^{(i,j-1)}+1)L_{g,x}^{(i,j+1)} + (\eta_y^{(i,j+1)}+1)L_{g,x}^{(i,j-1)} - (\eta_y^{(i,j+1)}+\eta_y^{(i,j-1)}+2)L_{g,x}^{(i,j)}}{(\eta_y^{(i,j+1)}+1)(\eta_y^{(i,j-1)}+1)(\eta_y^{(i,j+1)}+\eta_y^{(i,j-1)}+1)}, \quad (j>1) \quad (2.8.25b)$$

$$\text{ただし、} \eta_y^{(i,j-1)} \equiv \frac{\Delta_y^{(i,j-1)}}{\Delta_y^{(i,j)}} \quad (2.8.26a)$$

$$\eta_y^{(i,j+1)} \equiv \frac{\Delta_y^{(i,j+1)}}{\Delta_y^{(i,j)}} \quad (2.8.26b)$$

y-面に接するノード( $i, j=1$ )においては、90度回転対称性を利用し、(2.8.25)式において、次のような置き換えをすればよい。

$$\begin{cases} L_{g,x}^{(i,j-1)} \Rightarrow L_{g,y}^{(j=1,i)} \\ \eta_y^{(i,j-1)} \Rightarrow \frac{\Delta_x^{(j=1,i)}}{\Delta_y^{(i,j)}} \end{cases} \quad [j=1\text{の場合}] \quad (2.8.27a)$$

また、 $\rho_{1g,z,y}^{(i,j=1)}$ 及び $\rho_{2g,z,y}^{(i,j=1)}$ については、

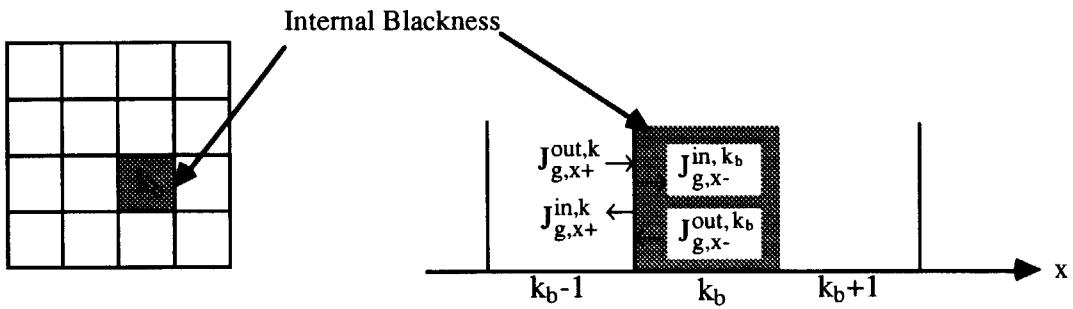
$$\begin{cases} L_{g,z}^{(i,j-1)} \Rightarrow L_{g,z}^{(j=1,i)} \\ \eta_y^{(i,j-1)} \Rightarrow \frac{\Delta_x^{(j=1,i)}}{\Delta_y^{(i,j)}} \end{cases} \quad [j=1\text{の場合}] \quad (2.8.27b)$$

このようにして得られる $\rho_{1g,x,y}^{(i,j)}$ 、 $\rho_{2g,x,y}^{(i,j)}$ 、 $\rho_{1g,z,y}^{(i,j)}$ 、 $\rho_{2g,z,y}^{(i,j)}$ を(2.6.23)式に適用する。

## 2.8.5 内部黒体

内部黒体(internal blackness)とは、炉心体系の外表面ではなく炉心内部に存在する一種の外挿境界\*である。研究炉などの特殊な炉心においては、強い熱中性子吸收物質を大きな塊(中性子の平均自由行程と比較して)として制御材に使用することがある。このような場合には、強吸收体の近くでは拡散理論が成立しなくなる。このため、強吸收体領域を直接断面積で扱わず、第2.8.1節の外挿境界と同じような扱いをする。ただし、外挿定数 $C_g$ は(2.8.4)式のように一定値とはせず、群に依存したユーザーの入力値を使用する。 $C_g$ の値は参照となる体系に対する実験結果や輸送計算との比較により決定される<sup>18)</sup>。更に、多群問題の場合、高速群においては黒体領域も強い中性子吸收材と見なされなくなるため、通常の断面積を使用した扱いとなる。すなわち内部黒体とは、いわば群に依存して炉心内部に現れる外挿境界と言える。現行PWRのクラスター型制御棒やBWRの十字型制御棒の場合は、集合体計算において燃料や減速材などと一緒に均質化され、炉心の拡散計算で直接扱われることはほとんどない。このため、プログラミングが複雑となり計算速度を遅くする原因となる内部黒体を扱えるようにしている拡散コードは数少ない<sup>19)</sup>。

\*2.8.1節では真空境界と呼んだが、ここでは外挿境界と呼ぶ方がふさわしいであろう。



内部黒体の扱いを行わないエネルギー群では、これまでに述べてきた通りの処理をすれば良い。内部黒体が現れるエネルギー群においては、例えば上図に示すように内部黒体( $k_b$ )に隣接するノード $k_b \pm 1$ に対して、 $x \pm$ 面が真空境界条件の場合と同様に扱う。

$$J_{g,x+}^{in,k_b-1} = \begin{cases} 1 - 2C_g^{k_b} f_{g,x+}^{k_b-1} \\ 1 + 2C_g^{k_b} f_{g,x+}^{k_b-1} \end{cases} J_{g,x+}^{out,k_b-1} \quad (x+面が内部黒体の場合) \quad (2.8.28a)$$

$$J_{g,x-}^{in,k_b+1} = \begin{cases} 1 - 2C_g^{k_b} f_{g,x-}^{k_b+1} \\ 1 + 2C_g^{k_b} f_{g,x-}^{k_b+1} \end{cases} J_{g,x-}^{out,k_b+1} \quad (x-面が内部黒体の場合) \quad (2.8.28b)$$

ただし、 $C_g^{k_b}$ は内部黒体に対して与えられる外挿定数である。また、ノード $k_b \pm 1$ においては、QTLAに対して、真空境界条件の場合と同様に、1次式近似またはバックリング近似で対応する。

内部黒体自体( $k_b$ )の部分中性流は物理的な意味を持たないが、プログラミング上は、未定義変数を避けるためと、内部黒体への正味中性子流入量の計算を容易にするために、次のような処理をしておく。

$$\begin{cases} J_{g,x-}^{out,k_b} = 0 \\ J_{g,x+}^{in,k_b} = J_{g,x+}^{out,k_b-1} - J_{g,x+}^{in,k_b-1} \end{cases} \quad (k_b \neq 1) \quad (2.8.29a)$$

$$\begin{cases} J_{g,x+}^{out,k_b} = 0 \\ J_{g,x-}^{in,k_b} = J_{g,x-}^{out,k_b+1} - J_{g,x-}^{in,k_b+1} \end{cases} \quad (k_b \neq N_x) \quad (2.8.29b)$$

また、内部黒体の中性子束も物理的な意味を持たないが、体系全体の反応率を計算し易いように便宜的にゼロを与えておく。

$$\phi_{g,0}^{k_b} = \phi_{g,x1}^{k_b} = \phi_{g,x2}^{k_b} = 0 \quad (2.8.30)$$

内部黒体が外部表面境界に接する場合には、以下に示すように、それぞれの境界条件に応じた例外処理をする。

(1) 内部黒体ノード( $k_b$ )が真空境界または反射境界に隣接する場合

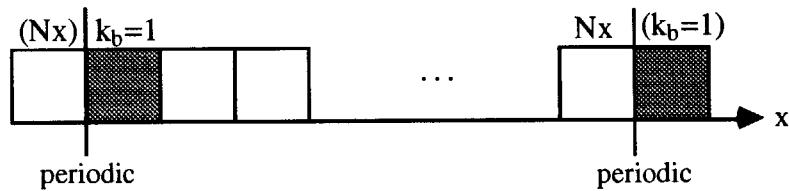


$$J_{g,x^-}^{in,k_b=1} = J_{g,x^+}^{out,k_b=1} = 0 \quad (k_b=1 \text{ の場合}) \quad (2.8.31a)$$

$$J_{g,x^+}^{in,k_b=Nx} = J_{g,x^+}^{out,k_b=Nx} = 0 \quad (k_b=Nx \text{ の場合}) \quad (2.8.31b)$$

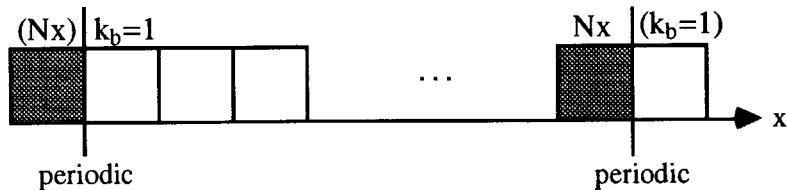
(2) 内部黒体ノード( $k_b$ )が周期境界に隣接する場合

- $k_b=1$ 、 $k=Nx$ が内部黒体でない場合



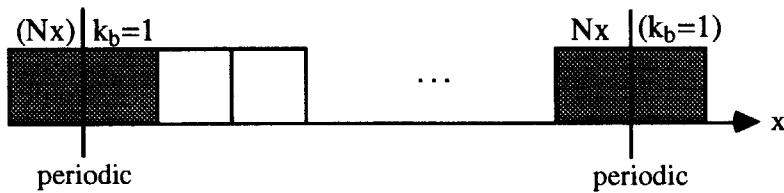
$$\begin{cases} J_{g,x^+}^{in,k=Nx} = \left( \frac{1-2C_g^{k_b=1}f_{g,x^+}^{k=Nx}}{1+2C_g^{k_b=1}f_{g,x^+}^{k=Nx}} \right) J_{g,x^+}^{out,k=Nx} \\ J_{g,x^-}^{out,k_b=1} = 0 \\ J_{g,x^-}^{in,k_b=1} = J_{g,x^+}^{out,k=Nx} - J_{g,x^+}^{in,k=Nx} \end{cases} \quad (2.8.32a)$$

- $k_b=Nx$ 、 $k=1$ が内部黒体でない場合



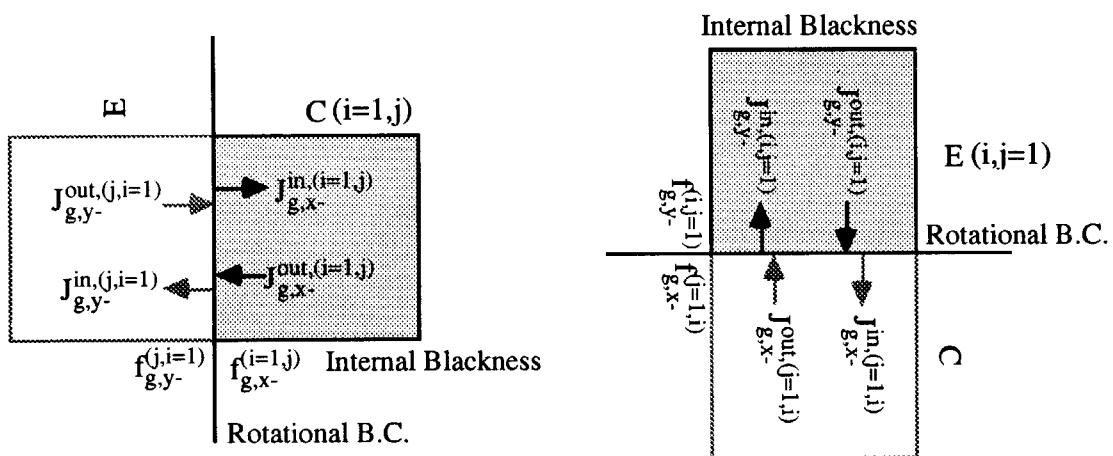
$$\begin{cases} J_{g,x^-}^{in,k=1} = \left( \frac{1-2C_g^{k_b=Nx}f_{g,x^-}^{k=1}}{1+2C_g^{k_b=Nx}f_{g,x^-}^{k=1}} \right) J_{g,x^-}^{out,k=1} \\ J_{g,x^+}^{out,k_b=Nx} = 0 \\ J_{g,x^+}^{in,k_b=Nx} = J_{g,x^-}^{out,k=1} - J_{g,x^-}^{in,k=1} \end{cases} \quad (2.8.32b)$$

- $k_b=1$ 、 $k=Nx$ が内部黒体の場合 (すなわち $k_b=Nx$ 、 $k=1$ が内部黒体の場合)



$$\begin{cases} J_{g,x^-}^{in,k_b=1} = J_{g,x^-}^{out,k_b=1} = 0 \\ J_{g,x^+}^{in,k_b=Nx} = J_{g,x^+}^{out,k_b=Nx} = 0 \end{cases} \quad (2.8.32c)$$

- (3) 内部黒体ノード(i, j)が90度回転境界に隣接する場合



(第2.8.4節の90度回転対称炉心参照)

- $(i=1,j)$ が内部黒体で、 $(j,i=1)$ が内部黒体ではない場合

$$\begin{cases} J_{g,y^-}^{in,(j,i=1)} = \frac{1 - 2C_g^{(i=1,j)}f_{g,y^-}^{(j,i=1)}}{1 + 2C_g^{(i=1,j)}f_{g,y^-}^{(j,i=1)}} J_{g,y^-}^{out,(j,i=1)} \\ J_{g,x^-}^{out,(i=1,j)} = 0 \\ J_{g,x^-}^{in,(i=1,j)} = J_{g,y^-}^{out,(j,i=1)} - J_{g,y^-}^{in,(j,i=1)} \end{cases} \quad (2.8.33a)$$

- $(i,j=1)$ が内部黒体で、 $(j=1,i)$ が内部黒体ではない場合

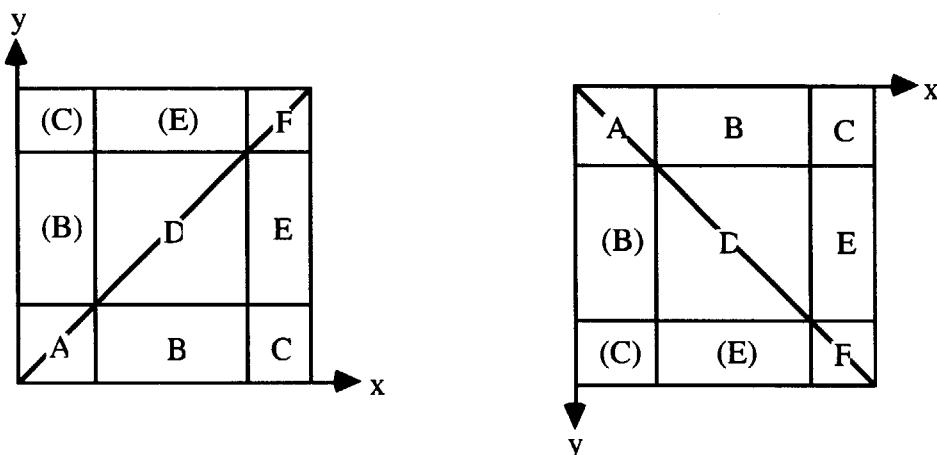
$$\begin{cases} J_{g,x^-}^{in,(j=1,i)} = \frac{1 - 2C_g^{(i,j=1)}f_{g,x^-}^{(j=1,i)}}{1 + 2C_g^{(i,j=1)}f_{g,x^-}^{(j=1,i)}} J_{g,x^-}^{out,(j=1,i)} \\ J_{g,y^-}^{out,(i,j=1)} = 0 \\ J_{g,y^-}^{in,(i,j=1)} = J_{g,x^-}^{out,(j=1,i)} - J_{g,x^-}^{in,(j=1,i)} \end{cases} \quad (2.8.33b)$$

- ・(i=1,j)が内部黒体で、(j,i=1)が内部黒体である場合 (すなわち(i,j=1)が内部黒体で、(j=1,i)が内部黒体である場合)

$$\begin{cases} J_{g,x^-}^{out,(i=1,j)} = J_{g,x^-}^{in,(i=1,j)} = 0 \\ J_{g,y^-}^{out,(i,j=1)} = J_{g,y^-}^{in,(i,j=1)} = 0 \end{cases} \quad (2.8.33c)$$

## 2.8.6 対角対称境界条件

対角対称は下図に示すように、右上がりの対角線について対称な場合と右下がりの対角線について対称な場合がある。

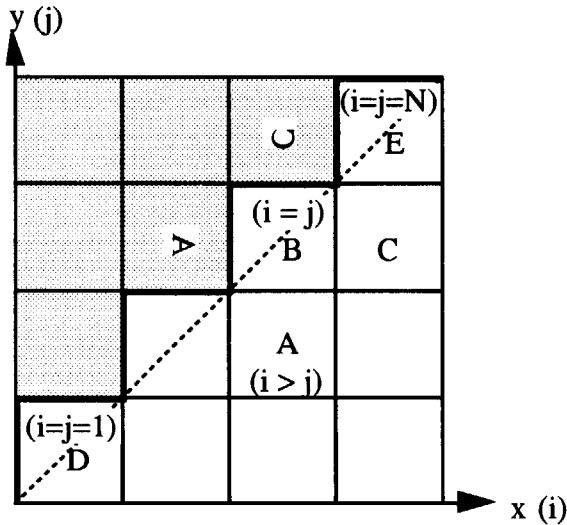


しかし、同図に示すように、右下がり対角対称な場合は、座標系の採り方により右上がり対角対称と同等に扱うことができるため、以下では右上がり対角対称の場合について述べる。対角対称境界条件が適用できるのは、(2.8.16)式と(2.8.17)式が満足され、更にx軸に沿った境界条件とy軸に沿った境界条件が等しい場合に限られる。対角対称境界条件を扱うにあたっては、z軸方向のノードは本質ではないため、水平方向ノード位置を(i,j)で表し、 $i \geq j$ のノードを計算対象として扱うものとする。以下では、QTLAへの対応、入射部分中性子流の更新、内部黒体への対応の順に話を進める。

### (1) QTLAへの対応

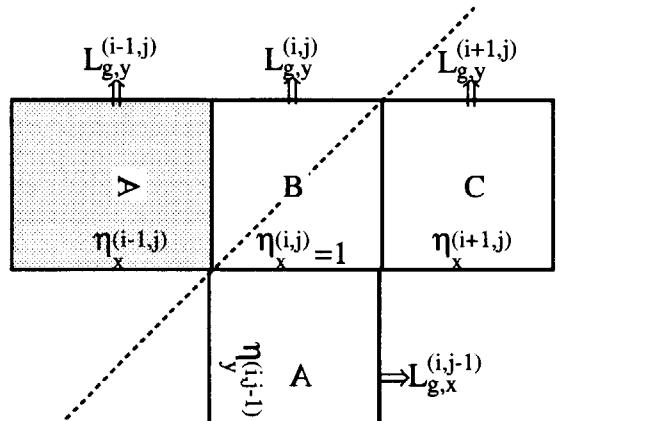
先ず*i > j*の位置にあるノード(例えば次ページの図のノードA)に着目すれば、左右上下にノードが存在するか、あるいは外部境界に接する。したがって、このようなノードについては、前節までの議論で対処可能である。対角対称境界条件を導入する際、例外処理が必要となるのは、対角線上( $i = j$ )にあるノードのみである。対角線上にあるノードは以下のように3通りに分類される。

- 外部境界に隣接しない対角線上ノード ( $i = j \neq 1, N$ 、例えば下図のノードB)
- $x$ -面と $y$ -面が外部境界に隣接するノード ( $i = j = 1$ 、例えばノードD)
- $x+$ 面と $y+$ 面が外部境界に隣接するノード ( $i = j = N$ 、例えばノードE)



### 1) $i = j \neq 1, N$ に位置するノードの処理

QTLAにおいて、 $y$ 方向Transverse Leakageの $x$ 方向分布の2次式係数は、一般に、(2.8.22)式で与えられる。右図のノードBに注目して、(2.8.22)式を使用する場合、 $L_{g,y}^{(i-1,j)}$ と $\eta_x^{(i-1,j)}$ が計算体系に存在していない。しかし、対角対称性を考慮すれば、



$$\begin{cases} L_{g,y}^{(i-1,j)} \Rightarrow L_{g,x}^{(i,j-1)} \\ \eta_x^{(i-1,j)} \Rightarrow \frac{\Delta_y^{(i,j-1)}}{\Delta_x^{(i,j)}} = \frac{\Delta_y^{(i,j-1)}}{\Delta_y^{(i,j)}} = \eta_y^{(i,j-1)} \end{cases} \quad (2.8.34a)$$

と置き換えることにより、対角線上ノードに対する $\rho_{1g,y,x}^{(i,j)}$ と $\rho_{2g,y,x}^{(i,j)}$ を得ることができる。一方、 $z$ 方向Transverse Leakageの $x$ 方向分布の係数 $\rho_{1g,z,x}^{(i,j)}$ 及び $\rho_{2g,z,x}^{(i,j)}$ についても、

$$\begin{cases} L_{g,z}^{(i-1,j)} \Rightarrow L_{g,z}^{(i,j-1)} \\ \eta_x^{(i-1,j)} \Rightarrow \eta_y^{(i,j-1)} \end{cases} \quad (2.8.34b)$$

と置き換えることにより得ることができ、得られた係数を(2.6.22)式に適用する。

次にy方向の分布を考える。x方向Transverse Leakageのy方向分布の係数は、一般に(2.8.25)式により与えられる。(2.8.25)式を対角線上ノードに適用する場合には、

$$\begin{cases} L_{g,x}^{(i,j+1)} \Rightarrow L_{g,y}^{(i+1,j)} \\ \eta_y^{(i,j+1)} \Rightarrow \frac{\Delta_x^{(i+1,j)}}{\Delta_y^{(i,j)}} = \frac{\Delta_x^{(i+1,j)}}{\Delta_x^{(i,j)}} = \eta_x^{(i+1,j)} \end{cases} \quad (2.8.35a)$$

と置き換えることにより、 $\rho_{1g,x,y}^{(i,j)}$ と $\rho_{2g,x,y}^{(i,j)}$ を得る。  
z方向Transverse Leakageのy方向分布の係数 $\rho_{1g,z,y}^{(i,j)}$ 及び $\rho_{2g,z,y}^{(i,j)}$ についても、

$$\begin{cases} L_{g,z}^{(i,j+1)} \Rightarrow L_{g,z}^{(i+1,j)} \\ \eta_y^{(i,j+1)} \Rightarrow \eta_x^{(i+1,j)} \end{cases} \quad (2.8.35b)$$

と置き換えることにより得ることができ、得られた係数を、(2.6.23)式に適用する。

x方向及びy方向Transverse Leakageのz方向分布については、通常の処理に任せることとする。以上により、(2.5.26)式で定義するベクトル $L_g^k$ の各成分が得られQTLAを適用することができる。

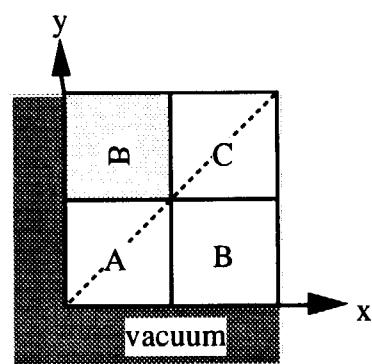
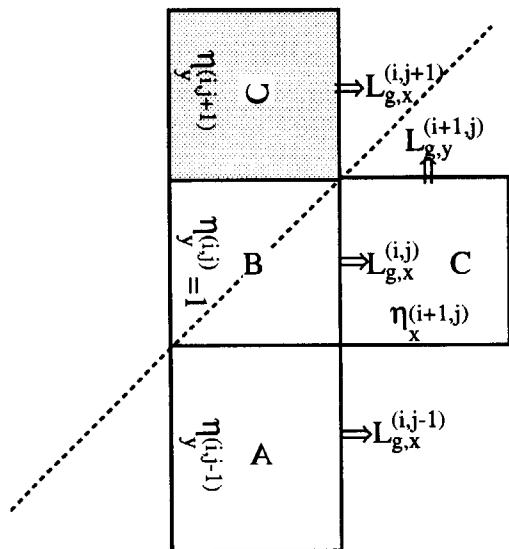
## 2) $i=j=1$ に位置するノードの処理

この場合、着目する対角線上ノードのx-面とy-面が外部境界に接するため、これらの境界条件に応じた例外処理が必要になる。基本的な考え方はこれまでと同じで、境界面の反対側にどのようなノードがどのような向きに配置されているか考えれば良い。

<真空境界に接する場合>

ノード(1,1)のx+面に接するノードが存在するならば1次式近似を、そうでなければ(x+面が真空または内部黒体に接する場合)、バックリング近似を適用する。1次式近似の場合には(2.8.10b)式より

$$\begin{cases} \rho_{1g,y,x}^{(i,j)} = \frac{2(L_{g,y}^{(i+1,j)} - L_{g,y}^{(i,j)})}{\eta_x^{(i+1,j)} + 1} \\ \rho_{2g,y,x}^{(i,j)} = 0 \end{cases} \quad (2.8.36a)$$



$$\begin{cases} \rho_{1g,z,x}^{(i,j)} = \frac{2(L_{g,z}^{(i+1,j)} - L_{g,z}^{(i,j)})}{\eta_x^{(i+1,j)} + 1} \\ \rho_{2g,z,x}^{(i,j)} = 0 \end{cases} \quad (2.8.36b)$$

y方向の分布については対角対称性を利用する。

$$\begin{cases} \rho_{1g,x,y}^{(i,j)} = \frac{2(L_{g,y}^{(i+1,j)} - L_{g,x}^{(i,j)})}{\eta_x^{(i+1,j)} + 1} \\ \rho_{2g,x,y}^{(i,j)} = 0 \end{cases} \quad (2.8.37a)$$

$$\begin{cases} \rho_{1g,z,y}^{(i,j)} = \frac{2(L_{g,z}^{(i+1,j)} - L_{g,z}^{(i,j)})}{\eta_x^{(i+1,j)} + 1} \\ \rho_{2g,z,y}^{(i,j)} = 0 \end{cases} \quad (2.8.37b)$$

<反射境界に接する場合>

(2.8.22)式及び(2.8.25)式を適用するにあたって、 $\rho_{1g,y,x}^{(i,j)}$  と  $\rho_{2g,y,x}^{(i,j)}$  の計算では、

$$\begin{cases} L_{g,y}^{(i-1,j)} \Rightarrow L_{g,y}^{(i,j)} \\ \eta_x^{(i-1,j)} \Rightarrow \eta_x^{(i,j)} = 1 \end{cases} \quad (2.8.38a)$$

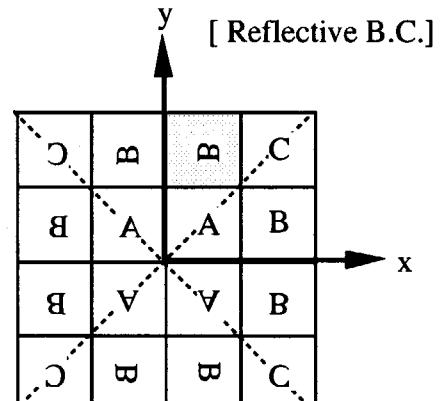
$\rho_{1g,z,x}^{(i,j)}$  と  $\rho_{2g,z,x}^{(i,j)}$  の計算では、

$$\begin{cases} L_{g,z}^{(i-1,j)} \Rightarrow L_{g,z}^{(i,j)} \\ \eta_x^{(i-1,j)} \Rightarrow \eta_x^{(i,j)} = 1 \end{cases} \quad (2.8.38b)$$

y方向の分布については対角対称性を利用する。

$$\begin{cases} L_{g,x}^{(i,j-1)} \Rightarrow L_{g,x}^{(i,j)} \\ \eta_y^{(i,j-1)} \Rightarrow \eta_y^{(i,j)} = 1 \\ L_{g,x}^{(i,j+1)} \Rightarrow L_{g,y}^{(i+1,j)} \\ \eta_y^{(i,j+1)} \Rightarrow \eta_x^{(i+1,j)} \end{cases} \quad (\rho_{1g,x,y}^{(i,j)} \text{ と } \rho_{2g,x,y}^{(i,j)} \text{ の計算}) \quad (2.8.39a)$$

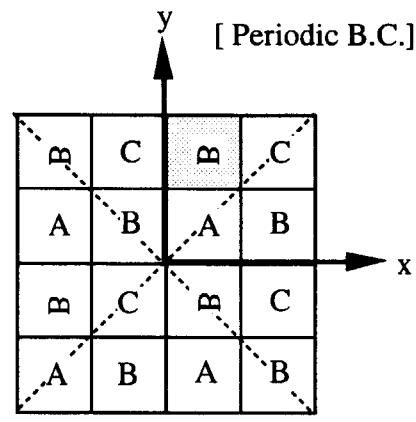
$$\begin{cases} L_{g,z}^{(i,j-1)} \Rightarrow L_{g,z}^{(i,j)} \\ \eta_y^{(i,j-1)} \Rightarrow \eta_y^{(i,j)} = 1 \\ L_{g,z}^{(i,j+1)} \Rightarrow L_{g,z}^{(i+1,j)} \\ \eta_y^{(i,j+1)} \Rightarrow \eta_x^{(i+1,j)} \end{cases} \quad (\rho_{1g,z,y}^{(i,j)} \text{ と } \rho_{2g,z,y}^{(i,j)} \text{ の計算}) \quad (2.8.39b)$$



&lt;周期境界に接する場合&gt;

$$\begin{cases} L_{g,y}^{(i-1,j)} \Rightarrow L_{g,y}^{(N,j)} \\ \eta_x^{(i-1,j)} \Rightarrow \eta_x^{(N,j)} = \frac{\Delta_x^{(N,j)}}{\Delta_x^{(i,j)}} \end{cases} \quad (\rho_{1g,y,x}^{(i,j)} \text{ と } \rho_{2g,y,x}^{(i,j)} \text{ の計算}) \quad (2.8.40a)$$

$$\begin{cases} L_{g,z}^{(i-1,j)} \Rightarrow L_{g,z}^{(N,j)} \\ \eta_x^{(i-1,j)} \Rightarrow \eta_x^{(N,j)} \end{cases} \quad (\rho_{1g,z,x}^{(i,j)} \text{ と } \rho_{2g,z,x}^{(i,j)} \text{ の計算}) \quad (2.8.40b)$$



$$\begin{cases} L_{g,x}^{(i,j-1)} \Rightarrow L_{g,y}^{(N,j)} \\ \eta_y^{(i,j-1)} \Rightarrow \eta_x^{(N,j)} \\ L_{g,x}^{(i,j+1)} \Rightarrow L_{g,y}^{(i+1,j)} \\ \eta_y^{(i,j+1)} \Rightarrow \eta_x^{(i+1,j)} \end{cases} \quad (\rho_{1g,x,y}^{(i,j)} \text{ と } \rho_{2g,x,y}^{(i,j)} \text{ の計算}) \quad (2.8.41a)$$

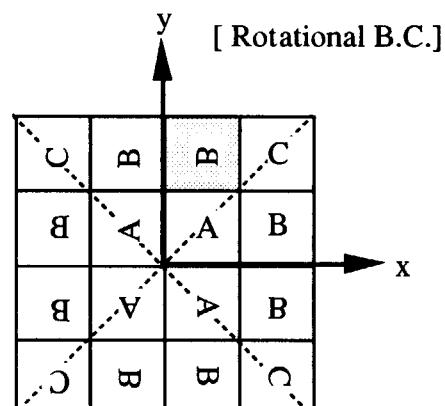
$$\begin{cases} L_{g,z}^{(i,j-1)} \Rightarrow L_{g,z}^{(N,j)} \\ \eta_y^{(i,j-1)} \Rightarrow \eta_x^{(N,j)} \\ L_{g,z}^{(i,j+1)} \Rightarrow L_{g,z}^{(i+1,j)} \\ \eta_y^{(i,j+1)} \Rightarrow \eta_x^{(i+1,j)} \end{cases} \quad (\rho_{1g,z,y}^{(i,j)} \text{ と } \rho_{2g,z,y}^{(i,j)} \text{ の計算}) \quad (2.8.41b)$$

&lt;90度回転境界に接する場合&gt;

$$\begin{cases} L_{g,y}^{(i-1,j)} \Rightarrow L_{g,x}^{(i,j)} \\ \eta_x^{(i-1,j)} \Rightarrow \eta_y^{(i,j)} = 1 \end{cases} \quad (\rho_{1g,y,x}^{(i,j)} \text{ と } \rho_{2g,y,x}^{(i,j)} \text{ の計算}) \quad (2.8.42a)$$

$$\begin{cases} L_{g,z}^{(i-1,j)} \Rightarrow L_{g,z}^{(i,j)} \\ \eta_x^{(i-1,j)} \Rightarrow \eta_y^{(i,j)} = 1 \end{cases} \quad (\rho_{1g,z,x}^{(i,j)} \text{ と } \rho_{2g,z,x}^{(i,j)} \text{ の計算}) \quad (2.8.42b)$$

$$\begin{cases} L_{g,x}^{(i,j-1)} \Rightarrow L_{g,y}^{(i,j)} \\ \eta_y^{(i,j-1)} \Rightarrow \eta_x^{(i,j)} = 1 \\ L_{g,x}^{(i,j+1)} \Rightarrow L_{g,y}^{(i+1,j)} \\ \eta_y^{(i,j+1)} \Rightarrow \eta_x^{(i+1,j)} \end{cases} \quad (\rho_{1g,x,y}^{(i,j)} \text{ と } \rho_{2g,x,y}^{(i,j)} \text{ の計算}) \quad (2.8.43a)$$



$$\begin{cases} L_{g,z}^{(i,j-1)} \Rightarrow L_{g,z}^{(i,j)} \\ \eta_y^{(i,j-1)} \Rightarrow \eta_x^{(i,j)} = 1 & (\rho_{1g,z,y}^{(i,j)} \text{ と } \rho_{2g,z,y}^{(i,j)} \text{ の計算}) \\ L_{g,z}^{(i,j+1)} \Rightarrow L_{g,z}^{(i+1,j)} \\ \eta_y^{(i,j+1)} \Rightarrow \eta_x^{(i+1,j)} \end{cases} \quad (2.8.43b)$$

### 3) $i = j = N$ に位置するノードの処理

この場合も、着目する対角線上ノードの  $x+$ 面と  $y+$ 面が外部境界に接するため、これらの境界条件に応じた例外処理が必要になる。右側及び上側境界では90度回転境界条件は考えない。また、対角対称性からこれらの境界には同じ境界条件が与えられる。特に、周期境界条件の場合には必然的に4方向とも周期境界となる。

#### < 真空境界に接する場合 >

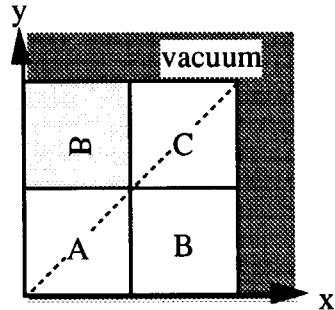
ノード  $(N, N)$  の  $x$ -面に接するノードと対角対称なノードが存在するならば1次式近似を、そうでなければ(真空または内部黒体に接する場合)、バックリング近似を適用する。Transverse Leakage の  $x$  方向分布に対して1次式近似を適用する場合には、(2.8.10a)式に対角対称性を考慮することにより、

$$\begin{cases} \rho_{1g,y,x}^{(i,j)} = \frac{2(L_{g,y}^{(i,j)} - L_{g,y}^{(i-1,j)})}{\eta_x^{(i-1,j)} + 1} = \frac{2(L_{g,y}^{(i,j)} - L_{g,x}^{(i,j-1)})}{\eta_y^{(i,j-1)} + 1} \\ \rho_{2g,y,x}^{(i,j)} = 0 \end{cases} \quad (2.8.44a)$$

$$\begin{cases} \rho_{1g,z,x}^{(i,j)} = \frac{2(L_{g,z}^{(i,j)} - L_{g,z}^{(i-1,j)})}{\eta_x^{(i-1,j)} + 1} = \frac{2(L_{g,z}^{(i,j)} - L_{g,z}^{(i,j-1)})}{\eta_y^{(i,j-1)} + 1} \\ \rho_{2g,z,x}^{(i,j)} = 0 \end{cases} \quad (2.8.44b)$$

$y$  方向の分布については、

$$\begin{cases} \rho_{1g,x,y}^{(i,j)} = \frac{2(L_{g,x}^{(i,j)} - L_{g,x}^{(i,j-1)})}{\eta_y^{(i,j-1)} + 1} \\ \rho_{2g,x,y}^{(i,j)} = 0 \end{cases} \quad (2.8.45a)$$

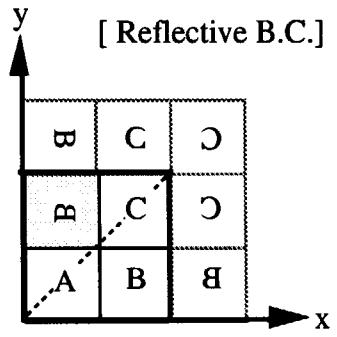


$$\begin{cases} \rho_{1g,z,y}^{(i,j)} = \frac{2(L_{g,z}^{(i,j)} - L_{g,z}^{(i,j-1)})}{\eta_y^{(i,j-1)} + 1} \\ \rho_{2g,z,y}^{(i,j)} = 0 \end{cases} \quad (2.8.45b)$$

&lt;反射境界に接する場合&gt;

(2.8.22)式及び(2.8.25)式を適用するにあたって、  
 $\rho_{1g,y,x}^{(i,j)}$  と  $\rho_{2g,y,x}^{(i,j)}$  の計算では、

$$\begin{cases} L_{g,y}^{(i-1,j)} \Rightarrow L_{g,x}^{(i,j-1)} \\ \eta_x^{(i-1,j)} \Rightarrow \eta_y^{(i,j-1)} \\ L_{g,y}^{(i+1,j)} \Rightarrow L_{g,y}^{(i,j)} \\ \eta_x^{(i+1,j)} \Rightarrow \eta_x^{(i,j)} = 1 \end{cases} \quad (2.8.46a)$$



$\rho_{1g,z,x}^{(i,j)}$  と  $\rho_{2g,z,x}^{(i,j)}$  の計算では、

$$\begin{cases} L_{g,z}^{(i-1,j)} \Rightarrow L_{g,z}^{(i,j-1)} \\ \eta_x^{(i-1,j)} \Rightarrow \eta_y^{(i,j-1)} \\ L_{g,z}^{(i+1,j)} \Rightarrow L_{g,z}^{(i,j)} \\ \eta_x^{(i+1,j)} \Rightarrow \eta_x^{(i,j)} = 1 \end{cases} \quad (2.8.46b)$$

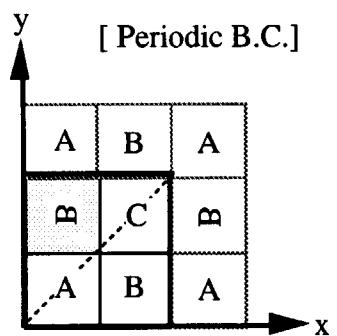
y方向の分布について、

$$\begin{cases} L_{g,x}^{(i,j+1)} \Rightarrow L_{g,x}^{(i,j)} \\ \eta_y^{(i,j+1)} \Rightarrow \eta_y^{(i,j)} = 1 \end{cases} \quad (\rho_{1g,x,y}^{(i,j)} \text{ と } \rho_{2g,x,y}^{(i,j)} \text{ の計算}) \quad (2.8.47a)$$

$$\begin{cases} L_{g,z}^{(i,j+1)} \Rightarrow L_{g,z}^{(i,j)} \\ \eta_y^{(i,j+1)} \Rightarrow \eta_y^{(i,j)} = 1 \end{cases} \quad (\rho_{1g,z,y}^{(i,j)} \text{ と } \rho_{2g,z,y}^{(i,j)} \text{ の計算}) \quad (2.8.47b)$$

&lt;周期境界に接する場合&gt;

$$\begin{cases} L_{g,y}^{(i-1,j)} \Rightarrow L_{g,x}^{(i,j-1)} \\ \eta_x^{(i-1,j)} \Rightarrow \eta_y^{(i,j-1)} \\ L_{g,y}^{(i+1,j)} \Rightarrow L_{g,x}^{(j,1)} \\ \eta_x^{(i+1,j)} \Rightarrow \frac{\Delta_y^{(j,1)}}{\Delta_x^{(i,j)}} \end{cases} \quad (\rho_{1g,y,x}^{(i,j)} \text{ と } \rho_{2g,y,x}^{(i,j)} \text{ の計算}) \quad (2.8.48a)$$



$$\begin{cases} L_{g,z}^{(i-1,j)} \Rightarrow L_{g,z}^{(i,j-1)} \\ \eta_x^{(i-1,j)} \Rightarrow \eta_y^{(i,j-1)} & (\rho_{1g,z,x}^{(i,j)} \text{ と } \rho_{2g,z,x}^{(i,j)} \text{ の計算}) \\ L_{g,z}^{(i+1,j)} \Rightarrow L_{g,z}^{(j,1)} \\ \eta_x^{(i+1,j)} \Rightarrow \frac{\Delta_y^{(j,1)}}{\Delta_x^{(i,j)}} \end{cases} \quad (2.8.48b)$$

$$\begin{cases} L_{g,x}^{(i,j+1)} \Rightarrow L_{g,x}^{(i,1)} & (\rho_{1g,x,y}^{(i,j)} \text{ と } \rho_{2g,x,y}^{(i,j)} \text{ の計算}) \\ \eta_y^{(i,j+1)} \Rightarrow \frac{\Delta_y^{(i,1)}}{\Delta_y^{(i,j)}} = \eta_y^{(i,1)} \end{cases} \quad (2.8.49a)$$

$$\begin{cases} L_{g,z}^{(i,j+1)} \Rightarrow L_{g,z}^{(i,1)} & (\rho_{1g,z,y}^{(i,j)} \text{ と } \rho_{2g,z,y}^{(i,j)} \text{ の計算}) \\ \eta_y^{(i,j+1)} \Rightarrow \eta_y^{(i,1)} \end{cases} \quad (2.8.49b)$$

## (2) 対角対称体系における入射部分中性子流の更新

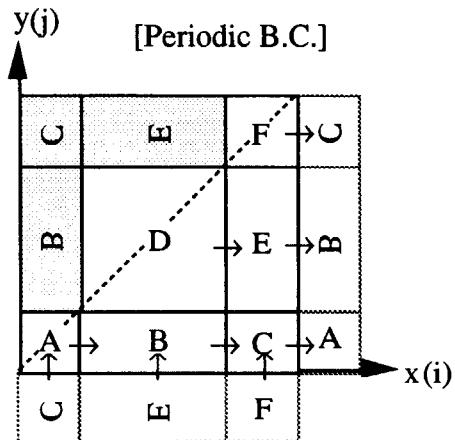
(2.5.21)式により各ノードの  $J_g^{\text{out},k}$  が得られたら、隣接ノードに対する入射部分中性子流  $J_g^{\text{out},k\pm 1}$  の更新を一般には(2.7.12)式と(2.7.13)式により行う。この際、対角対称境界条件を使用する場合には、例外処理が必要になる。1つのノードには6つの表面( $x\pm$ 、 $y\pm$ 、 $z\pm$ )がある。これらのうち、 $z\pm$ 面については対角対称境界条件に関係しないため、以下では $x\pm$ 面、 $y\pm$ 面における部分中性子流の更新を考える。

< $x+$ 面における処理>

着目するノード( $i, j$ )の $x+$ 面における処理は以下のように場合分けすることができる。

- ・ノード( $i+1, j$ )が存在する場合 : (2.7.12)式による処理
- ・真空境界に接する場合 : (2.8.7a)もしくは(2.8.9a)式による処理
- ・反射境界に接する場合 : (2.8.11a)式による処理
- ・内部黒体に接する場合 : (2.8.28a)式による処理
- ・周期境界に接する場合 : 対角対称性を利用した例外処理

$x+$ 面における処理において対角対称境界条件を考慮する必要があるのは、 $x+$ 面が周期境界に接する場合のみである。この場合、例えば右図において、ノードC,E,Fの $x+$ 面における部分中性子流を使用して、ノードA,B,Cの $y$ -面からの入射部分中性子流を更新する。すなわち、周期境界条件でも対角対称境界条件と併用する場合には、(2.8.13a)式の代わりに以下の式を使用する。



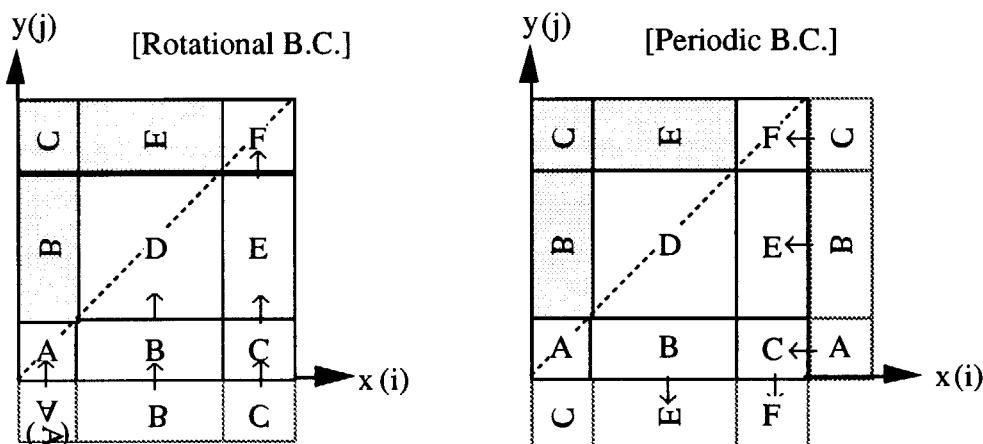
$$J_{g,y^-}^{in,(j,1)} = (1 - \alpha_{g,x+}^{(N,j)}) J_{g,x+}^{out,(N,j)} - \alpha_{g,x+}^{(N,j)} J_{g,x+}^{in,(N,j)} \quad (2.8.50a)$$

$$\text{ただし、 } \alpha_{g,x+}^{(N,j)} \equiv \frac{1}{2} \left( 1 - \frac{f_{g,x+}^{(N,j)}}{f_{g,y^-}^{(j,1)}} \right) \quad (2.8.50b)$$

#### < $y$ -面における処理 >

着目するノード( $i, j$ )の $y$ -面における処理は以下のように場合分けすることができる。

- ・ノード( $i, j-1$ )が存在する場合 : (2.7.13)式と同様な処理
- ・真空境界に接する場合 : (2.8.7b)もしくは(2.8.9b)式と同様な処理
- ・反射境界に接する場合 : (2.8.11b)式と同様な処理
- ・内部黒体に接する場合 : (2.8.28b)式と同様な処理
- ・90度回転境界に接する場合 : 対角対称性を利用した例外処理(反射境界と見なす)
- ・周期境界に接する場合 : 対角対称性を利用した例外処理



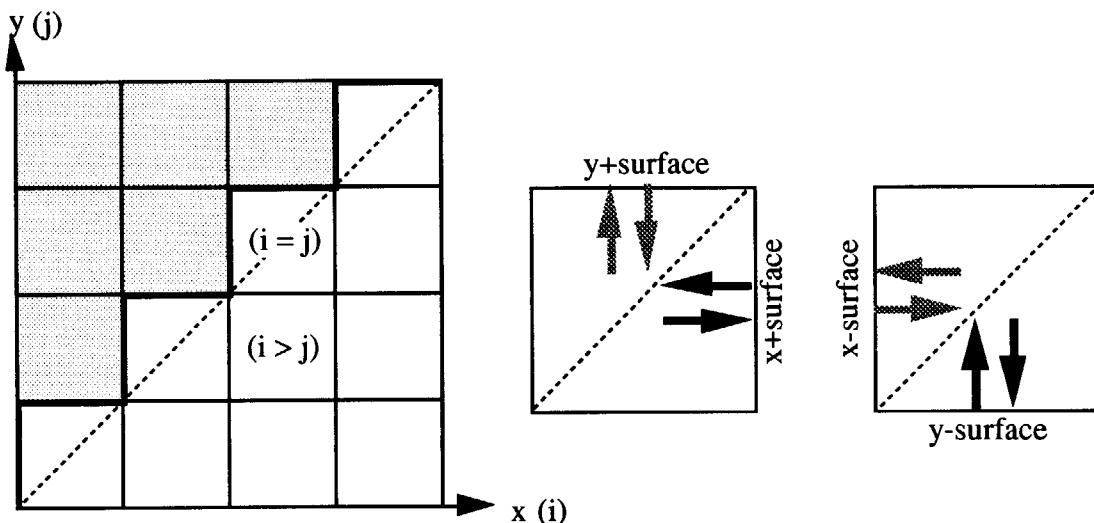
$y$ -面における処理において対角対称境界条件を考慮する必要があるのは、 $y$ -面が90度回転

対称境界もしくは周期境界に接する場合である。90度回転対称境界に接する場合は、上図(左)に示すように、y-面の処理は反射境界に接する場合と同じ扱いをすれば良い。一方、上図(右)に示すように、周期境界に接する場合には、例えばノードA,B,Cのy-面における部分中性子流を使用して、ノードC,E,Fのx+面からの入射部分中性子流を以下の式により更新する。

$$J_{g,x+}^{in,(N,i)} = (1 - \alpha_{g,y-}^{(i,1)}) J_{g,y-}^{out,(i,1)} - \alpha_{g,y-}^{(i,1)} J_{g,y-}^{in,(i,1)} \quad (2.8.51a)$$

$$\text{ただし、} \alpha_{g,y-}^{(i,1)} = \frac{1}{2} \left( 1 - \frac{f_{g,y-}^{(i,1)}}{f_{g,x+}^{(N,i)}} \right) \quad (2.8.51b)$$

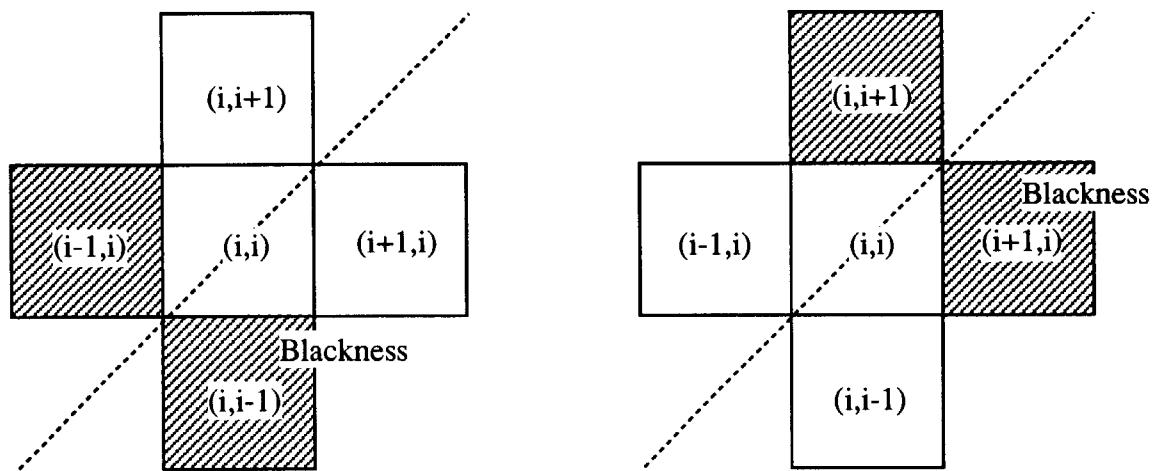
<x-面とy+面における処理>



上図において、 $(i>j)$ の位置にあるノードについては、x-面とy+面に隣接する部分中性子流の更新処理において対角対称性を考慮する必要性はない。例外処理が必要となるのは対角線上にあるノード $(i=j)$ についてのみである。一方、先に述べたx+面とy-面における例外処理が既に済んでいるのであれば、x+面とy-面の部分中性子流を対角対称性を利用して、y+面とx-面の部分中性子流として複写すればよい。

## (2) 対角対称体系において内部黒体を扱うときの注意点

QTLAの扱いは、既に述べたように、対角線上のノードに対して対称性を考慮した例外処理を要求する。この際、下図に示すように対角線上ノード $(i,i)$ に内部黒体が隣接する場合には注意が必要である。



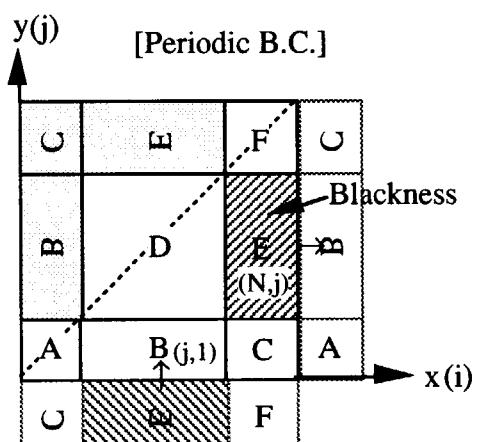
ノード(i,i)におけるTransverse Leakageのx方向分布を計算する場合には、(i-1,i), (i,i), (i+1,i)の位置にあるノードの情報が必要であり、対角対称体系では、(i-1,i)ノードの代わりに(i,i-1)ノードの情報を利用する(2.8.34式参照)。(i,i-1)ノードが黒体である場合には、QTLAは利用できず、(i,i)と(i+1,i)の2ノードの情報から1次式近似またはバックリング近似を適用する。内部黒体は一般にはエネルギー群に依存して現れるため、このような例外処理の判断を群依存で行う必要がある。

ノード(i,i)におけるTransverse Leakageのy方向分布を計算する場合には、(i,i-1), (i,i), (i,i+1)の位置にあるノードの情報が必要であり、対角対称体系では、(i,i+1)ノードの代わりに(i+1,i)ノードの情報を利用する(2.8.35式参照)。(i+1,i)ノードが黒体である場合には、(i,i)と(i,i-1)の2ノードの情報から1次式近似またはバックリング近似を適用する。

次に、入射部分中性子流を更新する際の注意点を考える。内部黒体に対する扱いは第2.8.5節で述べた通りである。しかし、対角対称境界条件を使用する場合で、内部黒体ノードのx+面またはy-面が90度回転対称境界または周期境界に接する場合には特別な注意が必要となる。x+面が回転対称境界に接することはありえない。y-面が回転対称境界に接する場合には、対角対称性から反射境界に接するものとして内部黒体に対する処理(2.8.31a式参照)をすればよい。問題となるのは、下図に示すように、黒体ノードのx+面またはy-面が周期境界に接する場合である。

右図において、(N,j)の位置にあるノードEが内部黒体であるとする。対角対称性と周期境界条件を考慮して、(j,1)の位置にあるノードBのy-面からの入射部分中性子流は以下の式により更新する。

$$J_{g,y^-}^{in,(j,1)} = \frac{1 - 2C_g^{(N,j)} f_{g,y^-}^{(j,1)}}{1 + 2C_g^{(N,j)} f_{g,y^-}^{(j,1)}} J_{g,y^-}^{out,(j,1)} \quad (2.8.52)$$



また、黒体自体のx+面における部分中性子流は、

便宜的に正味中性子流を保存するように決める。

$$\begin{cases} J_{g,x+}^{out,(N,j)} = 0 \\ J_{g,x+}^{in,(N,j)} = J_{g,y-}^{out,(j,1)} - J_{g,y-}^{in,(j,1)} \end{cases} \quad (2.8.53)$$

次に(i,1)の位置にあるノードBが内部黒体である場合を考える。対角対称性と周期境界条件を考慮して、(N,i)の位置にあるノードEのx+面からの入射部分中性子流は以下の式により更新する。

$$J_{g,x+}^{in,(N,i)} = \begin{cases} \frac{1-2C_g^{(i,1)}f_{g,x+}^{(N,i)}}{1+2C_g^{(i,1)}f_{g,x+}^{(N,i)}} \end{cases} J_{g,x+}^{out,(N,i)} \quad (2.8.54)$$

また、黒体自体のy-面における部分中性子流は、

$$\begin{cases} J_{g,y-}^{out,(i,1)} = 0 \\ J_{g,y-}^{in,(i,1)} = J_{g,x+}^{out,(N,i)} - J_{g,x+}^{in,(N,i)} \end{cases} \quad (2.8.55)$$

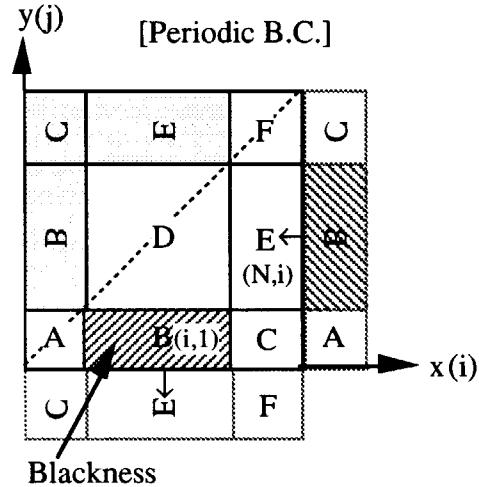
として与える。

## 2.8.7 アルベド境界条件

アルベドは反射係数 (reflection coefficient)とも呼ばれ、例えばx方向境界面において、以下の式により定義される。

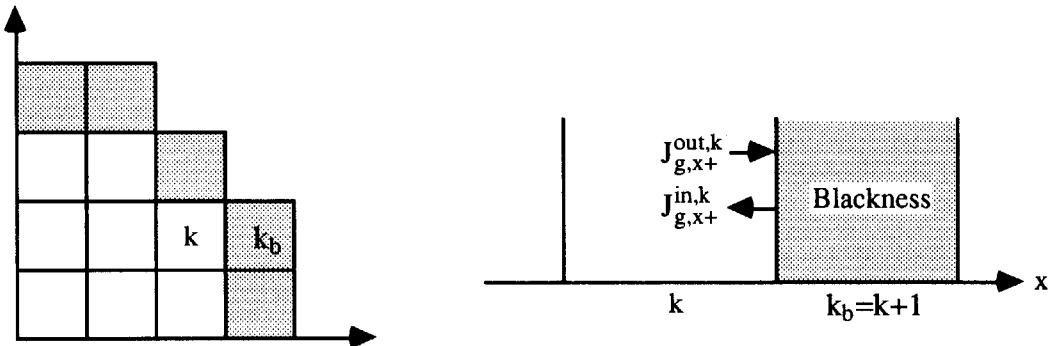
$$\alpha_{g,x+}^k \equiv \frac{J_{g,x+}^{in,k}}{J_{g,x+}^{out,k}} \quad (2.8.56)$$

外部反復計算において、反射体領域の中性子束の収束は炉心部の収束に比べて遅いことがある。また、特に有限差分法による3次元計算では、反射体領域のメッシュ数は無視できないほど多くなり、計算時間に多くの負担を掛けることがある。このため、予め適切なアルベドを何らかの方法で求めておいて、計算体系に反射体領域を含めず裸の原子炉を扱い、外表面においてアルベド境界条件を適用する。正確なアルベドの値は、反射体付きの体系



を3次元問題として解いて与えられるため、アルベド境界条件は、炉心の特性があまり変わらない問題を数多く解くことが必要な場合に使用される。例えば、アルベドの値は、基準となる体系を反射体を付けて解き、これにより得られる反射体に隣接するノードの部分中性子流を利用して、(2.8.56)式により与えることができる。

アルベド境界条件は、外挿境界条件または内部黒体と同様に扱うことができる。



例えば、上図に示すように炉心周辺部に黒体ノードを配置する。黒体ノードに隣接するノード $k$ の $x+$ 面に注目すれば、(2.8.28a)式より

$$J_{g,x+}^{in,k} = \frac{1 - 2C_g^{k_b} f_{g,x+}^k}{1 + 2C_g^{k_b} f_{g,x+}^k} J_{g,x+}^{out,k}, \text{ただし } k_b = k+1 \quad (2.8.57)$$

これと、(2.8.56)式との比較から、黒体ノードの外挿定数とアルベドの間には次の関係式が得られる。

$$\alpha_{g,x+}^k = \frac{1 - 2C_g^{k_b} f_{g,x+}^k}{1 + 2C_g^{k_b} f_{g,x+}^k} \quad (2.8.58a)$$

または、

$$C_g^{k_b} = \frac{f_{g,x+}^k (1 + \alpha_{g,x+}^k)}{2(1 - \alpha_{g,x+}^k)} \quad (2.8.58b)$$

反射体境界で不連続因子を扱わない場合には、 $f_{g,x+}^k = 1$ として、

$$C_g^{k_b} = \frac{(1 + \alpha_{g,x+}^k)}{2(1 - \alpha_{g,x+}^k)} \quad (2.8.58c)$$

すなわち、基準体系に対する計算結果から(2.8.56)式を使用して得られるアルベド値を(2.8.58b)または(2.8.58c)式に適用して、アルベドを内部黒体または外挿境界の外挿定数として扱うことができる。

## 2.9 計算フローの概要

4次の多項式展開拡散ノード法の一般的な計算の流れをべき乗法による固有値計算を例に考えると、以下のような計算手順となる。

- (a) (2.5.27)式～(2.5.30)式により、ノード毎に応答行列 $[P_g^k]$ 及び $[R_g^k]$ を作成する。
- (b)  $k_{eff}$ 及び全ノード( $k=1, 2, \dots, N$ )、全群( $g=1, 2, \dots, G$ )に対する $J_g^{out,k}$ (2.5.13式参照)、 $J_g^{in,k}$ (2.5.14式参照)、 $\Phi_g^k$ (2.5.25式参照)の初期値を与える。これらの初期値は体系全体の中性子生成率が $k_{eff}$ と一致するように規格化しておく。

$$\sum_{k=1}^N \sum_{g=1}^G v \Sigma_{f,g}^k \Phi_{g,0}^k V^k = k_{eff} \quad (2.9.1)$$

具体的な初期値の与え方については第4.2.6節で記述する。

- (c)  $\Phi_g^k$ の初期値を使用して、(2.9.2)式により全ノード、全群に対する核分裂源 $F_g^k$ の初期値を計算する。

$$F_g^k \equiv \frac{\chi_g^k}{k_{eff}} \sum_{g'=1}^G v \Sigma_{f,g}^k \Phi_{g'}^k \quad (2.9.2)$$

(2.9.2)式より、体系全体の中性子生成率は1.0に規格化される。

- (d) (2.9.3)式により第 $g$ 群に対する $Q_g^k$ (2.5.24式参照)を作成/更新する。

$$Q_g^k = F_g^k + S_g^k \quad (2.9.3)$$

$$\text{ただし、散乱源 } S_g^k \equiv \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{s,g' \rightarrow g}^k \Phi_{g'}^k \quad (2.9.4)$$

- (e) 第 $g$ 群の部分中性子流を使用して、(2.6.19)式により全ノードに対する $L_{g,x}^k$ 、 $L_{g,y}^k$ 、 $L_{g,z}^k$ を計算する。
- (f) QTLAに基づき(2.6.22)～(2.6.24)式により、第 $g$ 群に対する $L_g^k$ (2.5.26式参照)を全ノードに対して作成/更新する。この際、外部境界または内部黒体に隣接するノードについては、第2.8節で述べた例外処理を行う。
- (g) (2.5.27)式により第 $g$ 群の $J_g^{out,k}$ を更新する。

- (h) (2.7.12)式及び(2.7.13)式により、第 $g$ 群の $\mathbf{J}_g^{in,k}$ を更新する。この際、外部境界または内部黒体に隣接するノードについては、第2.8節で述べた例外処理を行う。
  - (i) 新しい $\mathbf{J}_g^{in,k}$ と $\mathbf{J}_g^{out,k}$ を使用して $L_{g,x}^k$ 、 $L_{g,y}^k$ 、 $L_{g,z}^k$ を更新する。また、(2.5.7)、(2.5.19)、(2.5.20)式により $\phi_g^k$ を更新する。
  - (j)  $\phi_{g,0}^k$ の収束判定をする。 $\phi_{g,0}^k$ が収束しているか若しくは内部反復計算の上限回数に至つていれば(k)に進む。そうでなければ(f)に戻り、第 $g$ 群に対する内部反復計算を繰り返す。
  - (k)  $g \Rightarrow g+1$ として(d)に戻り(ただし散乱源のみ更新)、次のエネルギー群に対する計算を続行する。全群に対する計算が終了したら(l)に進む。ただし、熱群の群数が多く上方散乱が顕著な場合には、熱群反復計算(thermal iteration)を行うことが収束加速の観点から望ましい(第4.2.3節参照)。
- 

- (l) 外部反復計算毎に体系全体の中性子生成率は1.0に規格化されるため、得られたノード平均中性子束から(2.9.1)式により体系全体の中性子生成率を計算すれば $k_{eff}$ が得られる。
  - (m)  $k_{eff}$ と $\phi_{g,0}^k$ の収束判定を行う。収束していれば計算を終了する。収束していない場合には、(2.9.2)式により $\mathbf{F}_g^k$ を更新(規格化)して(d)に戻る。
- 

(a)～(c)は前処理であり、(d)～(j)は特定のエネルギー群に対する内部反復計算である。また、(d)～(k)はエネルギー群に関するループであり、(d)～(m)が外部反復計算である。

### 3. スーパーコンピュータのための 高速計算アルゴリズム

#### 3.1 様々なスーパーコンピュータ

近年では、コンピュータの進歩がめざましく、従来のスカラー型と呼ばれるコンピュータに対し、ベクトル処理または並列処理を行う高速計算が可能なスーパーコンピュータが開発されている。Fig.3.1.1は、これらの計算機の処理方法を模式的に示したものである。ここでは、24匹のペンギン(データ)をエスカレータ(プロセッサ)により1階から2階へあげる作業(演算)を行っている。従来のスカラー計算機では、台座は1つしかなく、前のペンギンが2階に到着するまで次のペンギンはエスカレータに乗ることができなかつた。一方、ベクトル計算機では、次々とエスカレータに乗ることができる。ただし、あらかじめペンギンを整列させておく必要がある(ベクトル計算のための前処理)。ここで、ペンギンの数をベクトル長と呼ぶこととする。ベクトル長が長いほど、スカラー計算機に比べて高速な計算が可能となる。逆に、ベクトル長が短い場合には、ベクトル計算機の最高性能を引き出すことができなくなる。

並列計算機は、計算を行うプロセッサを複数個持つものである。ペンギンの例では、エスカレータ数を多くすることに対応する。エスカレータのタイプがスカラー型であるかベクトル型であるかにより、スカラー型の並列計算機とベクトル型の並列計算機に分類される。Fig.3.1.1(c)は、4並列のスカラー計算機の処理を示している。理想的には、1機のエスカレータを使うよりも、4倍の高速化が期待される。しかし、各エスカレータに乗るペンギンの数が均等でない場合(負荷の不均一)には、全体の処理速度は最もペンギンの数が多いエスカレータの処理により制限され、全てのペンギンが2階に上がるまで、処理を終了したエスカレータは休止する。また、並列計算機はペンギンを2階の別々の場所に運ぶため、ペンギンどうしが相談をして次の行動を決定するためには、相談相手がいる場所へ移動する手間が掛かる(プロセッサ間のデータ通信待ち)。この手間は、並列化が進むほど多くなる。Fig.3.1.1(d)は、ベクトル型の並列計算機の処理を示している。一見、最も処理が速そうに見受けられるが、24匹のペンギンは分散されるため、並列化度(エスカレータ数の増加)が進むほど、ベクトル長が短くなり計算効率を落としてしまう。このため、かなりの高並列化を目指す計算機では、スカラー型のものが主流となる。

スカラー及びベクトル計算機共に、並列計算機の場合には、共有メモリー型のものと分散メモリー型のものとに分類される(Fig.3.1.2)。また、これらの複合型も存在する。共有メモリー型では、データを収納するメモリー空間は1つであり、複数のプロセッサ(PE)から直接アクセスすることができる。一方、分散メモリー型の並列計算機では、複数のプロセッサが各個にメモリーを分散して所有しており、他のプロセッサが所有するメモリー内に収納されるデータが必要な場合には、プロセッサ間でデータの通信をする必要がある。共有メモリー型の欠点は、並列化度が進むと、分散されたPEが同じメモリアドレスにあるデータ

タをアクセスする頻度が多くなり、そのための待ち時間が計算速度を落とすことである。一方、分散型の並列計算機では、並列化度が進むとメモリーアクセスの競合は減るが、データの通信量が多くなり計算速度を落とすことになる。メモリーアクセスの競合は、データの処理が一括して行われる非並列型のベクトル計算機の場合にも起る。

ハードウェアは、高速計算が可能な環境(エスカレータの速度や数の増加)を提供するが、それだけでは高速な解析が実現するわけではない。ベクトル計算機に対しては、ベクトル長を長くする工夫や前処理をする必要がある。並列計算機では、負荷を均等にしたり通信を少なくしたりする必要がある。また、両計算機共に、メモリーアクセスの競合を可能な限り避けなければならない。これらを行うのはソフトウェアの役割であり、その考慮を怠ったソフトウェアは、いかに優れたスーパーコンピュータを導入しても、その性能を発揮することができないばかりか、かえって計算速度を落とすことになる。

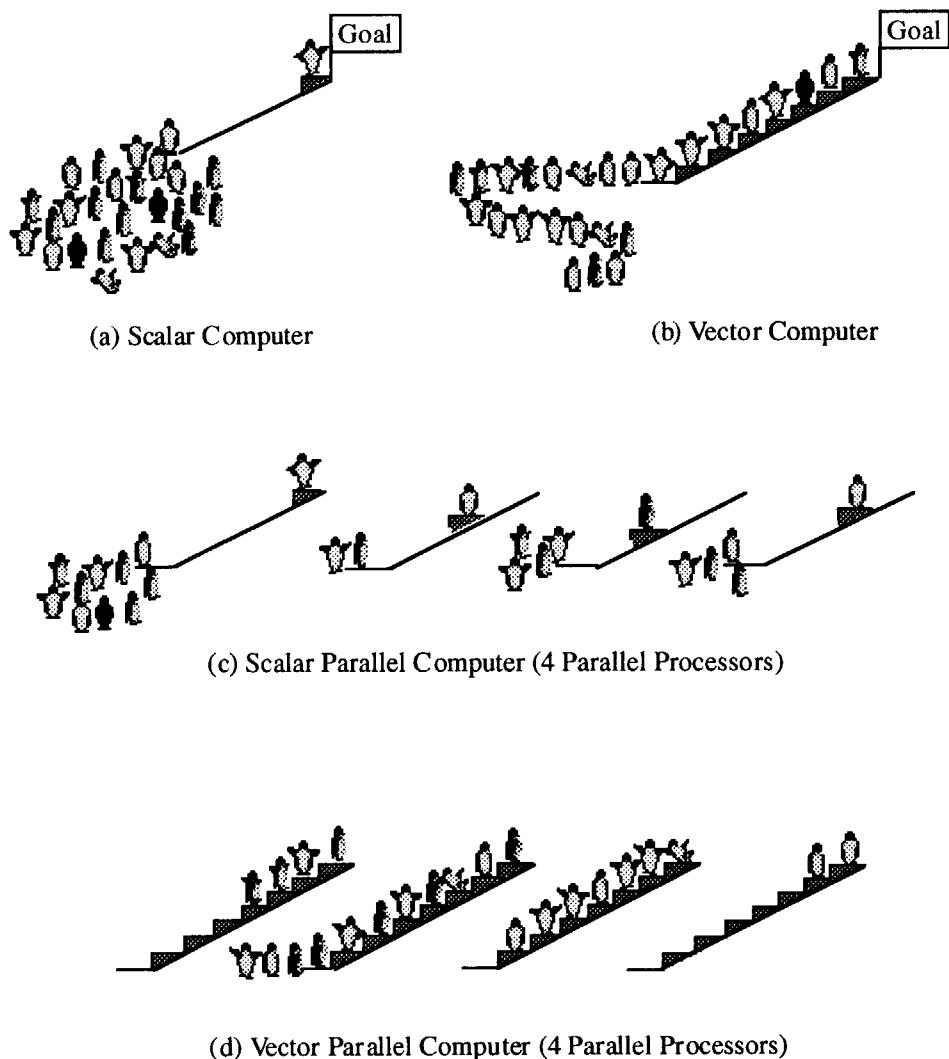
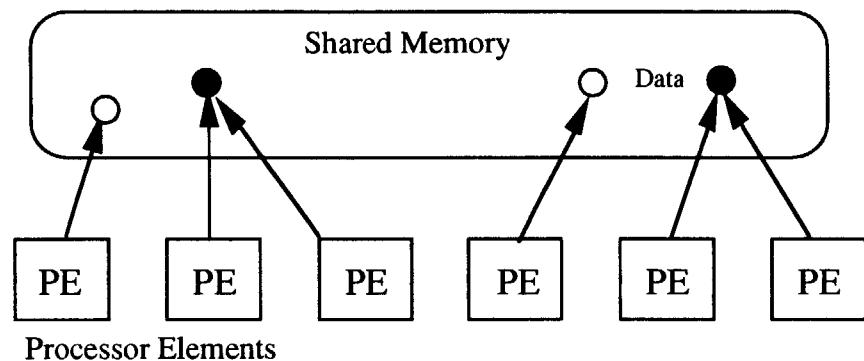
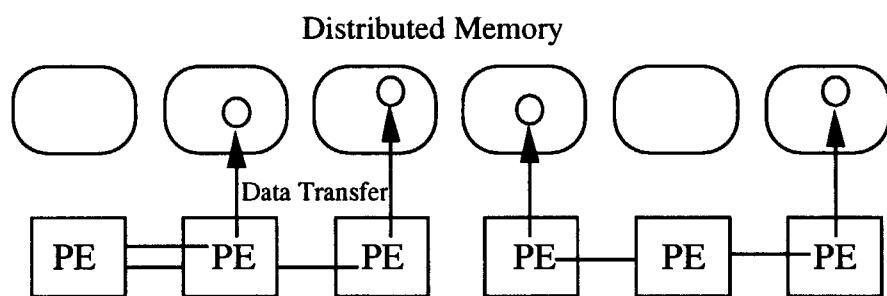


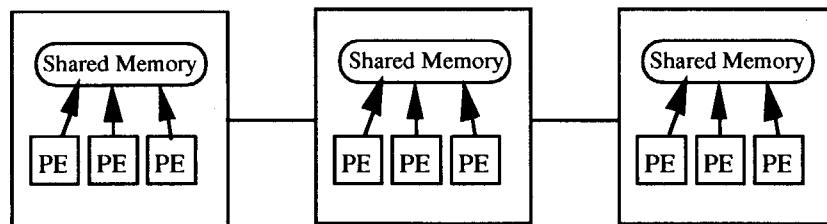
Fig.3.1.1 Image of data processing on various computers



(a) Shared memory type parallel computer



(b) Distributed memory type parallel computer



(c) Comprex type parallel computer

Fig.3.1.2 Classification of parallel computers by memory type

## 3.2 スーパーコンピュータのためのプログラミング技法

MOSRA-Lightコードにおける高速計算手法の理解を促すため、本節ではベクトル計算や並列計算の基本的なプログラミング技法について触れておく。

### 3.2.1 ベクトル計算機

ベクトル処理は、FORTRANのDO文の内部に書かれた演算式に対して行われる。例えば、配列Bと配列Cの1000個のデータを加算して配列Aに収納し、更に配列Eと配列Fのデータを加算して配列Dに収納する構文を考える。

```
DO 100 i=1,1000
    A(i)= B(i)+C(i)
    D(i)= E(i)+F(i)
100 CONTINUE
```

この場合、スカラー計算とベクトル計算では、次のような処理の違いがある。

<<スカラー処理>>

```
A(1)=B(1)+C(1)
D(1)=E(1)+F(1)
A(2)=B(2)+C(2)
D(2)=E(2)+F(2)
A(3)=B(3)+C(3)
D(3)=E(3)+F(3)
.
.
.
A(1000)=B(1000)+C(1000)
D(1000)=E(1000)+F(1000)
```

<<ベクトル処理>>

```
A(i)=B(i)+C(i) i=1～1000を一括処理
D(i)=E(i)+F(i) i=1～1000を一括処理
```

すなわち、スカラー演算では配列要素の一つ一つを逐次処理するのに対し、ベクトル演算では、配列全体を同時に一括処理する。注意すべきことは、DO文内の演算の順序が、スカラ処理とベクトル処理で異なる点であるが、これについては後に述べることにする。このような配列単位の一括処理をすることにより、ベクトル計算機では高速な計算が可能となる。この際、ベクトル長(上記の例では1000)が長いほど、スカラ計算に比べた計算速度は速くなる。逆にベクトル長が極端に短い場合には、スカラ計算より計算速度が遅くなることがある。これは、簡単に言えば、ベクトル計算を実施するために計算機内部で行われる僅かな準備時間が、ベクトル計算による計算時間の短縮効果を上回るためである。結局、ベクトル計算機を有効に利用する基本方針は、プログラム全体をベクトル長が長いDO文で構成させることである。しかし、全てのDOループ内の演算に対してこのようなベクトル処理が行われるわけではなく、一般に以下のようないくつかの規則が存在する(Fig.3.2.1参照)。

- ・多重のDOループにおいては、最内ループのみがベクトル化の対象となる。
- ・原則としてIF文やGOTO文などの分岐を含むDOループはベクトル化されない。
- ・READ/WRITEなどのI/O関連の文を含むDOループはベクトル化されない。
- ・CALL文を含むDOループはベクトル化されない。

・回帰演算を含むDOループはベクトル化されない。  
 これらの原則を踏まえ、具体的には以下のような点に留意してプログラムを作成する必要がある。

### (1) 多重のDOループ

反応率の計算などにおいて、しばしば配列変数をエネルギーと空間に関して積分する必要が生ずる。この場合、エネルギー積分と空間積分は、FORTRANのプログラムでは2重のDOループとして表現される。エネルギー積分と空間積分の順序を問わない場合には、ベクトル長を長くできる積分を優先させて最内DOループにすることが好ましい。例えば、炉心燃焼計算や空間依存の動特性解析計算では、ほとんどの場合が少数群(2~4群程度)で行われる。このような場合に、エネルギー群に関する積分を最内DOループにしても、ベクトル計算による高速化は期待できないため、空間積分を最内DOループとして行うべきである。

3次元の空間積分を実行する場合に、配列変数を例えば $X(i,j,k)$ のように3次元的に表現するのは賢明ではない。この場合、x、y、z各方向のメッシュ分割数をNX、NY、NZとすると、空間積分は $1 \leq i \leq NX$ 、 $1 \leq j \leq NY$ 、 $1 \leq k \leq NZ$ に関する3重のDOループとして計算される。したがって、ベクトル長はせいぜい何れかの方向のメッシュ分割数( $\max(NX, NY, NZ)$ )にしかならず、特に粗メッシュを扱うノード法コードでは不利となる。このような場合には、空間変数を $(X(m), m=NX*NY*NZ)$ のように1次元配列で表現することにより、ベクトル長を長くすることができます。

<<ベクトル長が短い>>

```
TOTAL=0.0
DO 100 i=1,NX
  DO 100 j=1,NY
    DO 100 k=1,NZ
      TOTAL=TOTAL+X(k, j, i)
    100 CONTINUE
  100 CONTINUE
```

<<ベクトル長が長い>>

```
TOTAL=0.0
DO 100 m=1, NX*NY*NZ
  TOTAL=TOTAL+X(m)
  100 CONTINUE
```

### (2) 分岐を含むDOループ

境界条件などに対応して、DOループ内に例外処理が必要になることがある。例外処理は、FORTRANにおいてIF文やGOTO文として表現される。このような条件付き分岐がDOループ内に現れる場合には、配列単位の一括処理が困難となり、該当するDOループ内の演算は原則としてベクトル化されない。IF文の構文が単純な場合には、コンパイラの機能(マスク機能・収集/拡散機能・リストベクトルアクセス機能)によって特別な処理を行いベクトル化を行うシステムも存在する。例えば、リストベクトルアクセス機能では、例外処理が必要な配列要素のインデックスリストをシステムが内部生成し、これに基づきベクトル処理を行う。このような場合、インデックスリストをシステムが内部生成する時間が余分に掛かり、IF分岐を含まない単純DOループの場合と比べて冗長な処理となる。ベクトル処理による計算時間の短縮よりも、インデックスリスト生成などの内部的な準備時間が大きい場合には、スカラー計算よりもかえって計算時間を多く消費することになる。スカラー計算機

用に作成されたプログラムを無理にベクトルコンパイルすると、かえって計算時間が遅くなることがあるのは、このような冗長処理が生じるためである。また、IF分岐を含むDOループの自動ベクトル化機能は、システムのコンパイラに強く依存するため、他機種への移植を考える場合には、これに頼らずプログラムレベルで対処した方が良い。

テーブル内挿法は多くのコードで使用されるが、ほとんどの場合がIF文を使用したプログラムになりベクトル化ができない。可能であるならテーブル内挿法の代わりに関数フィッティングに変更するとベクトル処理ができるようになる。DOループ内の分岐を避ける最も有効な方法は、何よりもプログラムの設計段階で、条件付き分岐を可能な限り減らす計算アルゴリズムを開発することである。MOSRA-Lightが採用する境界分離チェックボード・スウェーピ(第3.4.2節参照)は、例外処理を分離して扱うベクトル化手法の一例である。

### (3) I/Oを含むDOループ

READ文やWRITE文に代表されるI/O関連の文を含むDOループはベクトル化されない。DOループを使用して、入力データの読み込みや計算結果のプリント出力を行う場合には、やむを得ないことが多いが、同じDOループ内にベクトル処理が可能な大きな演算を含む場合には、演算部分とI/O部分を別のDOループとして分離することにより高速化が可能となる。

<< ベクトル計算不可 >>

```
DO 100 i=1,1000
    A(i)=B(i)+C(i)
    WRITE(6,*)
100 CONTINUE
```

<< 演算部を分離してベクトル計算 >>

```
DO 100 i=1,1000
    A(i)=B(i)+C(i)
100 CONTINUE
DO 200 i=1,1000
    WRITE(6,*)
200 CONTINUE
```

### (4) CALL文を含むDOループ

DOループの中でサブルーチンをCALLしている場合、コンパイラはこのサブルーチン内部で行われる演算の内容を判断できないため、原則としてベクトル化は行われない。システムに依っては、コンパイラがCALLされるサブルーチン内部の内容をDOループ内でインライン展開してベクトル化を行うものもある。しかし、これもCALLされるサブルーチンがごく単純な場合に限られるし、必ずしもコンパイラがこのような自動ベクトル化の機能を持っているとは限らない。したがって、ベクトル化による高速化を図るために、ベクトル演算が可能なDOループ内にCALL文を書かないような工夫をする必要がある。例えば、CALL文はIF分岐と併用して異なる処理をさせたい場合に良く使用される。この場合、DOループをサブルーチン内部に持ち込むことによりベクトル化が有効になることがある。以下は配列A(i)にiが500以下であれば0.0を、そうでなければ1.0を代入するプログラムである。

<< ベクトル計算不可 >>

```
DIMENSION A(1000)
DO 100 i=1,1000
    IF(i.LE.500) THEN
```

<< サブルーチン内でベクトル計算可 >>

```
DIMENSION A(1000)
NS=1
NE=500
```

```

        CALL SET0(A(i))
        ELSE
          CALL SET1(A(i))
        ENDIF
100 CONTINUE                               VALUE=0.0
                                              CALL SET01(A, NS, NE, VALUE)
                                              NS=501
                                              NE=1000
                                              VALUE=1.0
                                              CALL SET01(A, NS, NE, VALUE)

SUBROUTINE SET0(A)                         SUBROUTINE SET01(A, NS, NE, VALUE)
A=0.0                                         DIMENSION A(1000)
RETURN                                         DO 100 i=NS, NE
END                                           A(i)=VALUE
                                              100 CONTINUE
SUBROUTINE SET1(A)                         RETURN
A=1.0                                         END
RETURN
END

```

## (5) 回帰演算を含むDOループ

典型的な回帰演算の例を以下に示す。

```

C   B(i)には初期値として全て1.0が設定されているものとする。
      DO 100 i=1,3
          A(i)=B(i)
          B(i+1)=A(i)+1
100 CONTINUE

```

上記のような回帰演算を含むDOループを強制的にベクトル演算処理を行うものとすると、スカラー演算を行う場合と次のように計算結果が異なってくる。

## &lt;&lt;スカラー演算の処理&gt;&gt;

```

A(1)=B(1)      ! A(1)=1
B(2)=A(1)+1    ! B(2)=1+1=2
A(2)=B(2)      ! A(2)=2
B(3)=A(2)+1    ! B(3)=2+1=3
A(3)=B(3)      ! A(3)=3
B(4)=A(3)+1    ! B(4)=3+1=4

```

## &lt;&lt;ベクトル演算の処理&gt;&gt;

A(1)=B(1)	! A(1)=1	一括処理
A(2)=B(2)	! A(2)=1	
A(3)=B(3)	! A(3)=1	

B(2)=A(1)+1	! B(2)=2	一括処理
B(3)=A(2)+1	! B(3)=2	
B(4)=A(3)+1	! B(4)=2	

実際には、コンパイラがDOループ内に回帰演算が含まれることを認識してベクトル化コンパイルを実施せず、誤動作を生じることは無い。逆に言えば、回帰演算を含むDOループはスカラー処理が行われ、高速化ができない。また、プログラムが複雑化すると、表層的な構文からコンパイラが回帰演算であるかどうか判断できない場合がある。このような場合、実際には回帰演算となっていなくても、コンパイラは安全上、スカラー処理を行うようにコンパイルをする。

回帰演算は、実際のプログラミングにおいてしばしば現れる。例えば、ノード法の場合、 $k$ 番目ノードの部分中性子流を使用して、隣接するノード( $k \pm 1$ )の入射部分中性子流を更新する場合に回帰演算となる(2.7.12式、2.7.13式参照)。要は、配列をベクトル化により一括処理する際(並列計算の場合も同じ)、配列要素のデータが相互に干渉する場合に回帰演算となる。これを避けるためには、DOループ内の演算を配列要素に関して完全に独立した演算とする

ことが肝要である。

回帰演算を回避する方法として、チェックカードスウェープ (Red-and-Black Sweep、または Odd-Even Sweepとも呼ばれる)は有効な手段の一つである。通常、内部反復計算のように体系スウェープを行う場合には、隣接するメッシュのデータを利用するが多く、空間メッシュに対して同時処理を行うと、演算の干渉が生ずる。このような演算の干渉を避けるために、Fig.3.2.2に示すようにメッシュ分割された空間をチェックカード状に2色(例えば赤と黒)に分類し、先ず赤色のメッシュに対する処理を一括処理(ベクトル処理)する。引き続き黒色メッシュに対する処理を一括処理する。一つの赤色メッシュに対するデータが更新されている時、別の赤色メッシュのデータが同時に更新されていても、これらは隣接していないため、互いの計算になんら影響を及ぼさない。よって、回帰演算にならないためベクトル化による高速な処理が可能となる。3次元問題の場合には、軸方向について色を反転させたボードを交互に配置させれば、チェックカードスウェープを利用することができます。

ところで、このようなチェックカードスウェープも、プログラム上では単なるDOループにすぎない。それが、チェックカードスウェープになっているかどうかは、プログラマーのみぞ知ることであり、コンパイラには判断できない。したがって、コンパイラは回帰演算の可能性があることを認識し、安全上スカラーコンパイルを行おうとする。これを避けるため、プログラマーがコンパイラにDOループ内の演算が回帰演算にならないことを知らせてやる必要がある。このために、各計算機システムが定める形式に従って、ソースプログラムの内部にベクトル制御指示行を挿入する。例えば、FACOMのベクトル計算機(VP-2600、VPP-500など)では、直下のDOループが回帰演算を含んでいないことを明示するため、以下のようなベクトル制御指示行を挿入する。

```
*VOCL LOOP, NOVREC
  DO 100 I=1,1000
  .
  100 CONTINUE
```

このようなベクトル制御指示行は、いわば計算機システムに依存した方言であり、プログラムを異種計算機に移植する場合には、それに応じたベクトル制御指示行に変更する必要がある。ベクトル制御指示行は、1カラム目の記号"\*\*"により、FORTRANではコメント行と見なされるため、そのままスカラー計算機などに移植することができる。

#### (6) メモリーアクセスの競合

メモリーアクセス競合の原理については、第3.1節で述べたとおりである。ベクトル計算機や並列計算機など、同時処理を行うシステムではメモリーアクセスの競合に十分注意しなければならない。プログラミング上はうまくベクトル化や並列化がされていても、メモリーアクセスの競合が頻繁にあると、実行上は高速な演算が達成できない。具体例として、ノード法コードにおける断面積データへのメモリーアクセスの方法を考えてみる。臨界集合体をイメージすると、炉心体系中にはせいぜい数種類のタイプの材料しか無く、それは

ど多くの均質化断面積セットは必要ない。したがって、ノード単位に断面積セットを与えるなくても、断面積セットの認識番号を与えて間接的に断面積データにアクセスすれば良い。例えば、反応率の計算などは、以下のように行う。

```
C  REACT(i) : Reaction Rate of the i-th Node
C  MAT(i)   : Material Identification Number of the i-th Node
C  XS(k)    : XS Data of the k-th Material
C  FLUX(i)  : Flux of the i-th Node
C
      DO 100 i=1,N
          REACT(i)= XS(MAT(i)) *FLUX(i)
100 CONTINUE
```

このようにすると、確かに断面積セットを一つ一つのノードに割り当てるよりも、データ量は遥かに少なくてすみ、使用メモリーを節約することができる。しかし、ベクトル計算機で上記DOループ内の演算を一括処理する場合、断面積データのアクセスにおいて、明らかにメモリーアクセスの競合が生じる。これを避けるためには、ノード毎に断面積データを割り当てて、以下のようなプログラムにすると良い。

```
DO 100 i=1,N
    REACT(i)= XS(i) *FLUX(i)
100 CONTINUE
```

メモリー節約の観点からは一見不合理に思われるが、計算コストが高い核熱結合炉心燃焼計算や空間依存の動特性解析計算においては、燃焼度の分布や熱水力フィードバックなどにより結局ノード単位に断面積セットを与えることになる。これらの解析に比べれば、臨界集合体の解析コストは遥かに安価であり、むしろノード単位に断面積セットを割り当てて、メモリーアクセスの競合を避ける方が賢明と言える。

#### (7) 順次メモリーアクセス

メモリーアクセスの順序を考慮することは、ベクトル計算機やFORTRANに限らず一般的なプログラミングの基本と言われる。例えば、次のような2次元配列A及び3次元配列Bを考える。

```
DIMENSION A(2,3), B(2,2,3)
```

計算機のメモリー上では、Fig.3.2.3に示すように配列A及び配列Bの記憶場所(アドレス)が1次元的に割り当てられる。配列Aの全ての要素に1.0を代入するプログラムは、以下に示すように2通りに書くことができる。

<pre>&lt;&lt; Program1 &gt;&gt;</pre> <pre>DO 100 j=1,3     DO 100 i=1,2         A(i,j)=1.0 100 CONTINUE</pre>	<pre>&lt;&lt; Program2 &gt;&gt;</pre> <pre>DO 100 i=1,2     DO 100 j=1,3         A(i,j)=1.0 100 CONTINUE</pre>
--	--

同じ演算ではあるが、Program1の場合は、Fig.3.2.3の配列Aの記憶場所を連続的にアクセス

するのに対し、Program2の場合は不連続的なアクセスとなる。このようなメモリーアクセスの順番の違いにより計算速度の違いが現れる。例えば、以下のような2つのプログラムをベクトル計算機とスカラー計算機(EWS)で行い、計算時間を計測してみた。

<< 順次メモリーアクセス >>      << 不連続メモリーアクセス >>

```
DIMENSION A(2000,2000)                  DIMENSION A(2000,2000)
DO 100 j=1,2000                          DO 100 i=1,2000
DO 100 i=1,2000                          DO 100 j=1,2000
      A(i,j)=1.0                          A(i,j)=1.0
100 CONTINUE                                  100 CONTINUE
```

予想通り、順次メモリーアクセスをする方が計算時間は速く、ベクトル計算機で約1.4倍、スカラー計算機で約1.7倍の速度差が観測された。速度比自体は計算機種、配列サイズ、配列の次元等に強く依存するであろうが、順次メモリーアクセスの奨励はまんざら嘘ではないようである。

ベクトル計算機を利用する場合には、既に述べたように、ベクトル長が長くできるDOループを最内ループとするプログラミングが望ましい。これと、順次メモリーアクセスの推奨から、多次元配列の宣言の仕方も自ずと決まってくる。例えば、エネルギー群(NG=2)と空間メッシュ(MESH=2000)に依存する反応率を計算するプログラムを考える。

<< Program3 >>

```
DIMENSION XS(MESH, NG), FLUX(MESH, NG), REACT(MESH, NG)
DO 100 ig=NG
DO 100 m=1,MESH
      REACT(m, ig)=XS(m, ig)*FLUX(m, ig)
100 CONTINUE
```

<< Program4 >>

```
DIMENSION XS(NG, MESH), FLUX(NG, MESH), REACT(NG, MESH)
DO 100 ig=NG
DO 100 m=1,MESH
      REACT(ig, m)=XS(ig, m)*FLUX(ig, m)
100 CONTINUE
```

どちらのプログラムも、ベクトル長を長くする観点から空間メッシュに関するDOループを内側に置いている。しかし、Program4の方は、反応率、断面積、中性子束に関するメモリーアクセスが不連続になっている。これは、反応率、断面積、中性子束に対する2次元配列をエネルギー群とメッシュに関して逆に宣言しているためである。散乱マトリックスの配列を宣言する場合でも、In-ScatteringとOut-Scatteringのどちらの計算頻度が多いかを考慮して、2次元配列の宣言を決めることが好ましい。プログラムの作成において、物理量に対応した多次元配列の設計はごく初期に行うものであるが、プログラム全体が完成してからの修正労力は大変なものとなるため、全体の計算アルゴリズムに照らして、最初の配列設計を十分に考慮して行う必要がある。

ベクトル計算機用のプログラム作成における留意点を以上7項目あげた。その他にも、機種やコンパイラのサービス機能に応じた細かい留意点もあるが、基本的には上述の基本方針を徹底してプログラムを作成することにより、ベクトル計算機の性能を十分に発揮させることができる。

既存コードをベクトル化する場合(ベクトルチューニング)には、典型的な計算を実行させ、サブルーチン単位の動的な計算コスト(CPU時間)の分布を求め、コストの高いものから順にベクトル計算機に適したプログラムに変更する。計算コストが数個のサブルーチンに集中している場合にはチューニングは容易であるが、特に大型コードなどで計算コストが分散している場合には労力の割に計算速度はあまり向上しない。また、既存コードのチューニングは、配列設計やアルゴリズムの根本的な変更まで踏み込むことが困難なため、格段の速度向上は望めないことが多い。

### 3.2.2 並列計算機

現在の並列計算機には、既に述べたように、スカラー並列計算機とベクトル並列計算機とがあり、更にメモリータイプにより、共有メモリー型と分散メモリー型に大別され、計4種類(複合メモリー型を入れれば6種類)の並列計算機が存在する。プログラム作成上の留意点は、これら個々の計算機によって異なるが、以下では一般論のみを挙げておく。

#### (1) 演算の独立化

並列計算機では複数のプロセッサ(PE)により演算が同時処理されるため、モンテカルロ法の粒子追跡のように、演算が独立であることが重要である。演算が独立していない場合には、同期待ちや通信が頻繁に生じて高速な演算ができなくなる。同時処理の対象となる演算を独立させる点においてベクトル計算の場合と共通であり、ベクトル計算のアルゴリズムを並列計算にも利用することができる。例えば、前節のベクトル計算機において回帰演算を避けるチェックボードスワイプ(Fig.3.2.2)は、並列計算においても非常に有効な手法となる。ベクトル計算との違いは、最内DO文の中の演算に限定されず、また演算が独立であれば、IF分岐やCALL文などが含まれても並列計算が可能な点である。

#### (2) 計算負荷の均等化

プロセッサが複数ある場合には、ペンギンの例で示したように、計算負荷を等しくするよう心がけなければならない(Fig.3.2.4(a)参照)。モンテカルロ計算の場合には、各PEに割り振る粒子の数を均等にすることにより、おおよその計算負荷を等しくすることができる。一方、決定論的手法で複数の方程式を解く必要がある場合は、負荷の均等化を難しくする。例えば、核熱結合炉心計算の場合を考えてみる。核計算では、メッシュ数(またはノード数)を各PEに等しく配分(領域分割法)することにより負荷を均等にすることができる。しかし、同じメッシュ分割を熱水力計算に適用した場合、単相流の部分と二相流の部分では、二相

流の方が計算内容は複雑となり、負荷にばらつきを生ずる。

物理的な考慮と無関係に計算負荷を等しくする方法には、DO文の分割処理がある。

```
DO 100 i=1,1000
    A(i)=B(i)+C(i)
100 CONTINUE
```

例えば、上記のプログラムで、iが1から500までの演算を1つのPEに任せ、iが501から1000までの演算を別のPEに任せれば、完全に負荷を均等にすることができる。しかし、これがベクトル計算機の場合には、ベクトル長を短くし、ベクトル効率を落とすことになる。

### (3) メモリーアクセスの競合

ベクトル計算機の場合と同様、並列計算機の場合にもメモリーアクセスの競合ができるかぎり避けなければならない。特に共有メモリー型の並列計算機で並列化による速度向上率を落とすのは、これが原因であることが多い。例えば、並列計算に適したモンテカルロ法であっても、並列処理されている粒子が同じ断面積ライブラリーの参照を求めるのであれば、Fig.3.2.4(b)に示すようにメモリーアクセスの競合が頻繁に生じ、アクセス待ちにより計算効率を落としてしまう。Fig.3.2.4(c)の分散メモリー型では、このようなメモリーアクセスの競合が生じにくい利点がある。しかし、断面積ライブラリーを全ての分散メモリー上に配置しておく必要があり、一般に高並列なマシンほど分散メモリーのサイズが小さくなる傾向があるため、巨大なライブラリーがメモリーに載らないことがある。熱水力計算における蒸気表や物性値のテーブルなどに関しても同様である。

### (4) 通信の削減

分散メモリー型の並列計算機では、分散されたPEに接続するメモリー内のデータに、他のPEから直接アクセスできない。このため、PE間でデータの通信が必要になるが(Fig.3.2.4(c)参照)、現在の計算機は演算速度に比べて通信速度は遙かに遅い。通信待ちによる計算速度の律速を避けるには、何よりも通信を必要としない計算アルゴリズムを構築することである。モンテカルロ法が並列計算に適しているのは、粒子の追跡を全く独立して行うことができるためである。バッチ毎に各PEの計算結果を編集して統計処理を行うまで通信をほとんど必要としない。一方、決定論的手法の多くは空間をメッシュ分割し、メッシュ間の物理量の情報伝達を"流れ"により扱うことが多い。このため、領域分割法では高並列になるほど"流れ"による通信が頻繁に生じ並列化による速度向上率を低下させることが多い。

### (5) ベクトル化との両立

ベクトル並列計算機の場合、ベクトル化と並列化の効率を両立させることは極めて難しい。多くのプログラムは、並列化を行うとベクトル長を短くしてベクトル効率を落してしまうことが多い。ベクトル計算機では、最内DOループに対してのみベクトル化が行われ、また、IF分岐やCALL文を含むDOループはベクトル化できない。幸い並列計算にはこのよ

うな制限がないため、ベクトル計算と並列計算をうまく両立させることにより一層高速な計算ができる可能性がある。

```

DO 100  i=1, 4
DO 200  j=1, 1000
    A(j,i)=B(j,i)*C(j,i)
200 CONTINUE
100 CONTINUE

```

例えば、上記のプログラムで、外側のDOループ(100)を4並列に分割したとする。ベクトル計算は最内ループ(200)に対して有効で、各PEは次のようなイメージで計算を行う。

PE-1 :  $A(j,1)=B(j,1)+C(j,1)$   $j=1 \sim 1000$ を一括処理  
 PE-2 :  $A(j,2)=B(j,2)+C(j,2)$   $j=1 \sim 1000$ を一括処理  
 PE-3 :  $A(j,3)=B(j,3)+C(j,3)$   $j=1 \sim 1000$ を一括処理  
 PE-4 :  $A(j,4)=B(j,4)+C(j,4)$   $j=1 \sim 1000$ を一括処理

このような場合には、並列計算とベクトル計算が両立しており、DOループの縦割り分割のようにベクトル効率を落とすことなく、高速計算が可能である。しかし、実際に当面する問題では、このようなプログラミングは難しい。例えば、 $i$ をエネルギー群、 $j$ を空間メッシュと置き換えて以下のような中性子拡散方程式を解くことを考えてみる。

$$-\nabla D_g(\mathbf{r}) \cdot \nabla \phi_g(\mathbf{r}) + \Sigma_{r,g}(\mathbf{r}) \phi_g(\mathbf{r}) = Q_g(\mathbf{r}) \quad (3.2.1)$$

$$\text{ただし、 } Q_g(\mathbf{r}) = \frac{\chi_g(\mathbf{r})}{k_{\text{eff}}} \sum_{g'=1} \nu \Sigma_{f,g'}(\mathbf{r}) \phi_{g'}(\mathbf{r}) + \sum_{g' \neq g} \Sigma_{s,g' \rightarrow g}(\mathbf{r}) \phi_{g'}(\mathbf{r}) \quad (3.2.2)$$

予め中性子源の項 $Q_g(\mathbf{r})$ が与えられれば(例えば前の外部反復計算の解を使用する)、(3.2.1)式をエネルギー( $g$ )と空間( $r$ )に分離して解くことは可能である。しかし、中性子源の項には減速散乱源が含まれており、エネルギー群を独立して解くと、上群の解を下群の散乱源に反映することができなくなり、収束を遅くすることになる。また、分散メモリー型の場合にはエネルギー群に関する積分で常に通信を必要とする。熱群反復計算(Thermal-Iteration)をすると、高速群を担当するPEが休止する。何よりも、並列化による高速化の限度が群数で決定される。

ベクトル効率を落とすことなく高並列化を達成することは、決定論的手法に限らず一般的に難しい問題である。ベクトル並列計算機では、高並列を狙いすぎて返って実質的な速度を落とすこともあるため、ベクトル化と並列化のバランスを考慮して全体の速度向上を図る必要がある。

## (6) 並列化の対象

並列計算において最も重要な点は、何に対して並列化するか？であろう。例えば、少数

群で計算されることが多い炉心計算で、エネルギー群に関して並列化しても、高並列計算機の恩恵をあまり得られない。ノード法において、炉心の軸方向ノード分割数に関して並列化したとする。現在の軽水炉の炉心長は約4mであり、ノード幅20cmとして計算する場合には、最高20倍の並列計算しかできない。扁平な炉心では更に並列度の上限が小さくなる。一般的に言えば、実際の解析において並列度に対し制限とならないものを並列化の対象とすべきである。しかし、この際、上記の(1)～(5)も考慮して決めなければならない。

Only most inner DO-loop can be vectorized.  
Vector length should be as long as possible.

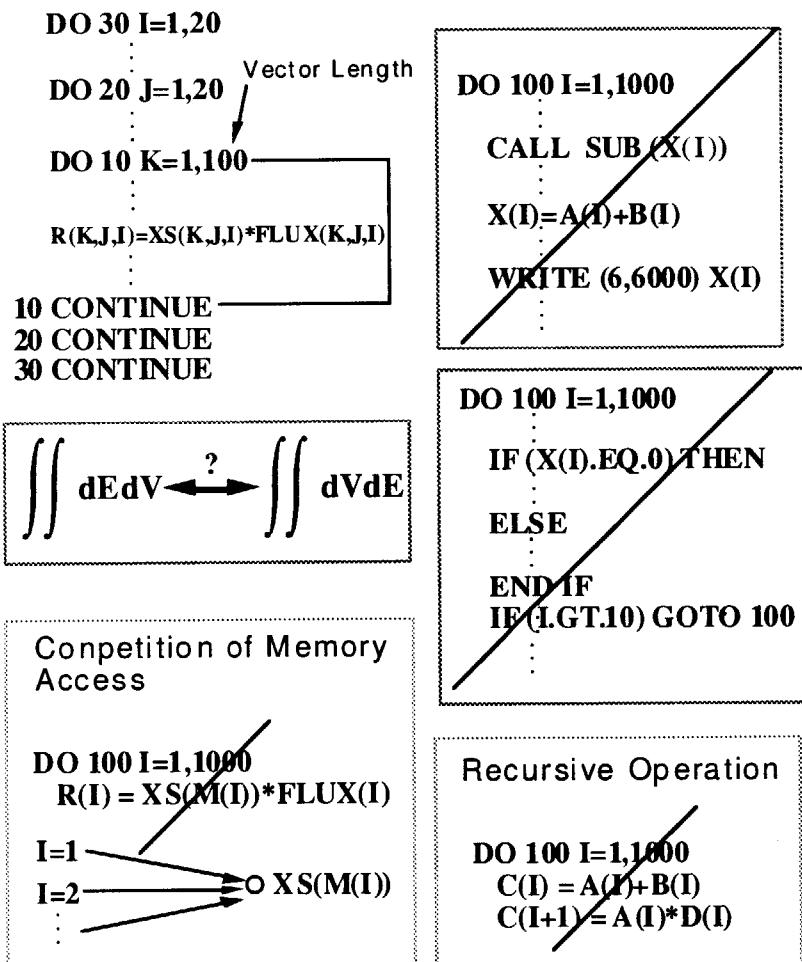


Fig.3.2.1 Programming techniques for supercomputers (Vector Computers).

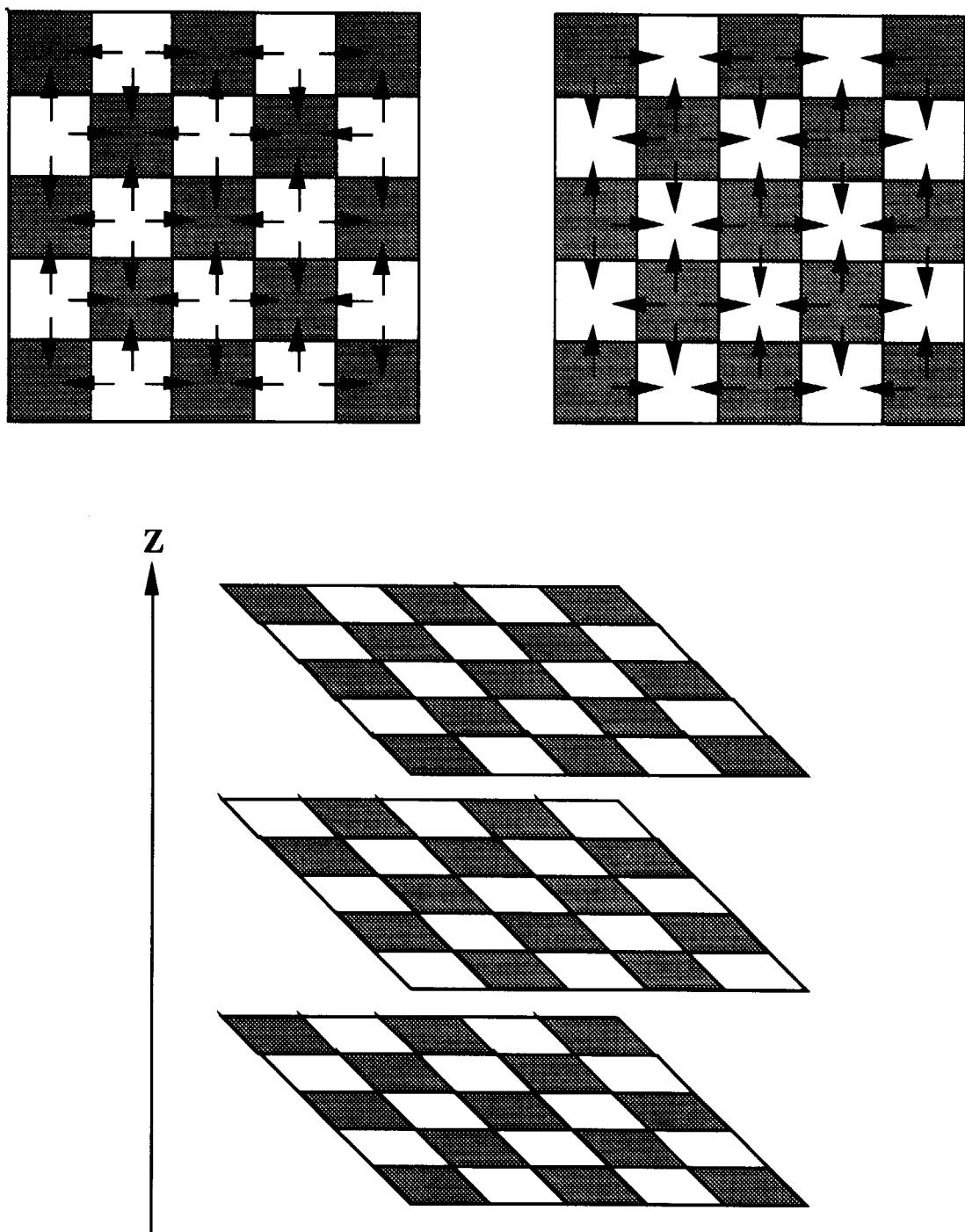


Fig.3.2.2 Checkerboard sweep to avoid recursive operation in vector or parallel computers.

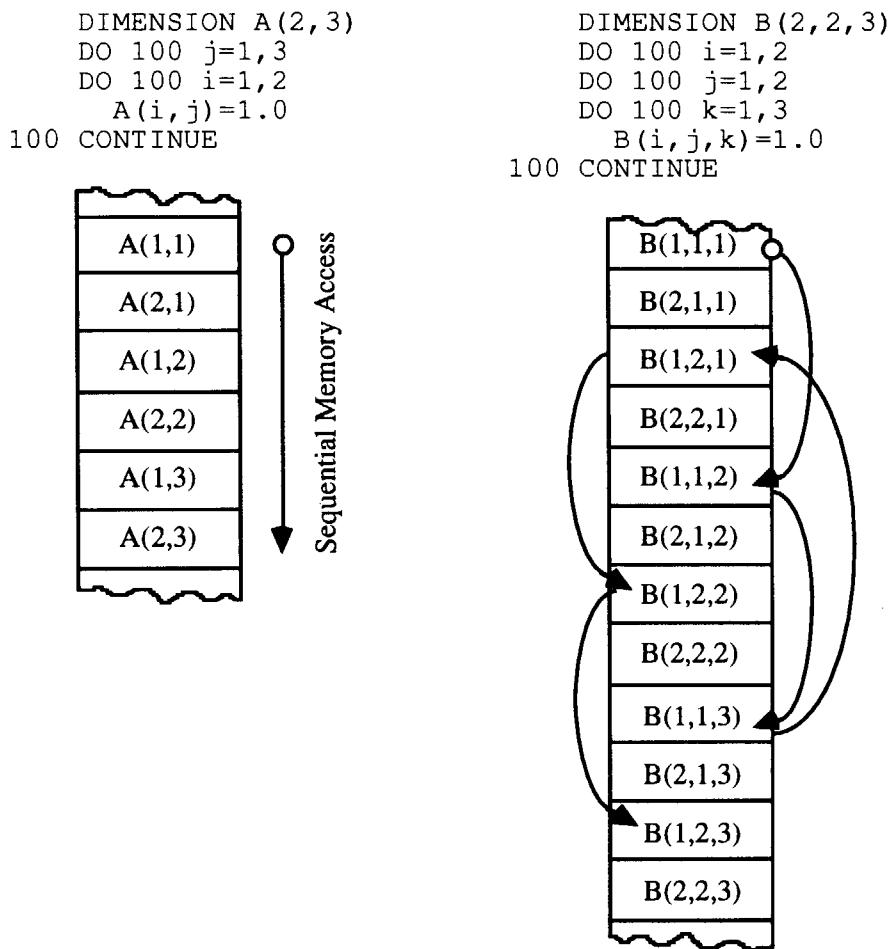
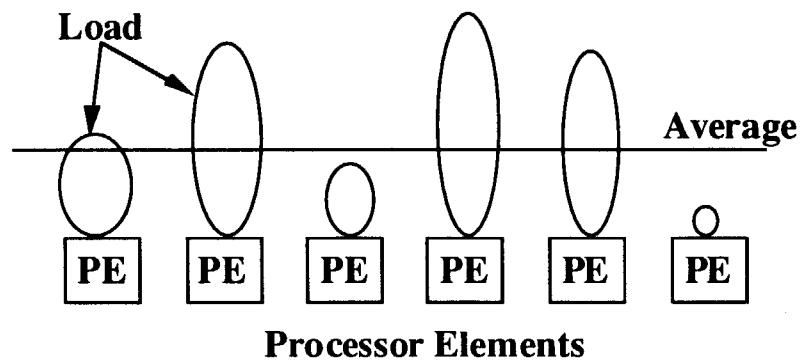
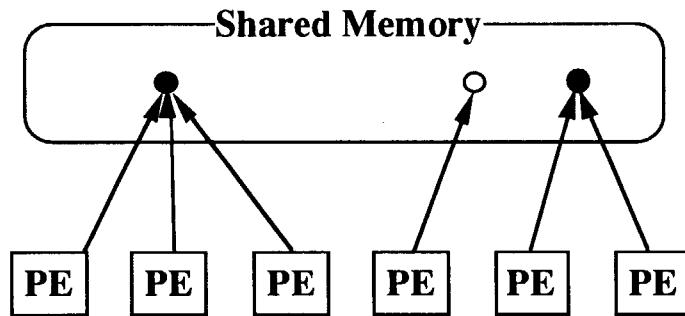


Fig.3.2.3 Memory address of multi-dimensional array and order of memory access depending on programs. Sequential memory access is recommended for speedup.

**(a) Computation load should be equally distributed.**



**(b) Competition of memory access should be avoided.**



**(c) Data communication among PEs should be reduced if possible.**

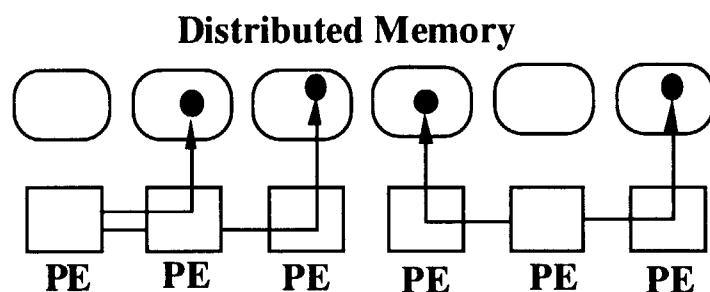


Fig.3.2.4 Programming techniques for supercomputers (Parallel Computers).

### 3.3 高速計算の実現と問題点

#### 3.3.1 ソフトウェアの重要性

コンピュータの計算性能は年々伸びていると言われる。ピーク性能に関して言えば確かにその通りかもしれない。しかし、実際のアプリケーションソフトの実行速度は、ベクトルまたは並列計算機になってから、それほど伸びているとは思われない。その原因の1つには、計算機の本来性能を発揮できるプログラム作成の努力が行われていないことによる。

ベクトル計算機のスカラー計算に対する速度向上率を $P$ とすると、 $P$ は次の式で表される。

$$P = \frac{1}{(1-V) + V/\alpha} \quad (3.3.1)$$

ただし、 $V$ はベクトル化率、 $\alpha$ はベクトル-スカラー速度比と呼ばれ、次のように定義される。

$$V = \frac{\text{ベクトル処理可能な部分のスカラーCPU時間}}{\text{スカラーによる全体のCPU時間}} \quad (3.3.2)$$

$$\alpha = \frac{\text{ベクトル処理可能な部分のスカラーCPU時間}}{\text{ベクトル処理可能な部分のベクトルCPU時間}} \quad (3.3.3)$$

これらの関係を図示すると、Fig.3.3.1のようになる。プログラム作成上留意すべきことは、先ず、 $\alpha$ の値を大きくすべき事である。これは、

- ・ベクトル長を長くする。
- ・データの引用と定義が連續的
- ・演算子(+,-,\*,/.)が多い
- ・IF文が少ない
- ・メモリアクセスが少ない

といった事で実現され、具体的には第3.2.1節の内容と一致する。最も重要な点は、何よりもベクトル長を長くすることである。一方、ベクトル化率は、ベクトル計算により速度向上できる部分が、動的に見て(実際計算の反復などを考慮して)、プログラム全体の何割あるかを意味する。これは、プログラム全体(特に反復計算などでコストが大きい部分)を、徹底してベクトル化可能なDOループの集まりとして書くことにより実現される。

結局、ベクトル長をできる限り長くして、更にベクトル化率が少なくとも95%以上になるようなプログラムを作成しないかぎり、ベクトル計算機の本来性能を発揮させることができない。多くのプログラムは、これらの努力が払われていないため、ベクトル計算機の恩恵を得ていないことが多い。しかし、問題によっては、ベクトル化に適さないプログラムがあることも確かではある。

並列計算機は、これまでの技術で高速化に成功したプログラムの一層の高速化やベクト

ル化の恩恵を得られなかったプログラムの高速化に光明を与える技術として注目されている。並列計算機は、内蔵するプロセッサ(PE)の数にはほぼ比例して計算の速度が向上するようと思われるが、実際には、そのような速度向上率が達成されるのは、第3.2.2節の並列計算のためのプログラミング技法を徹底して遵守したコード、あるいはそれが可能な恩恵された手法を利用するコードのみである。Fig.3.3.2は、PEの数と並列化による速度向上率の関係を、最近のモンテカルロコードと決定論的手法コードの現状として概略示したものである(分散メモリー型スカラー並列計算機)。ちなみに、同様な図は多くの論文で見られるが、座標をLinear-LinearかLog-Logで表すかによりかなり印象が異なるため、デジタル値で判断することが望ましい。モンテカルロ法のように独立した演算は、PE数が100以上になつても高い並列化効率が得られるが、決定論的手法の多くは約20PEが1つの壁となっており、これ以上の並列化はかなりの技術を要する。最近では、近代拡散ノード法を含め決定論的手法に対する並列化の試みが数多く見られるが<sup>19)-23)</sup>、50以上の高並列マシンで高い並列化効率を実現することはかなり困難なようである。

並列化プログラムの作成は、特に分散メモリー型の並列計算機において、プログラミング上の制約が多く難しい。現在の並列計算機のピーク性能にソフトウェアが追従していない現状は、単にプログラマーの努力不足だけではなく、並列計算技術そのものが十分に熟成していないせいもある(第3.3.2節参照)。いずれにせよ、近年のスーパーコンピューターは、高速計算が可能な環境を提供しているのであり、その実現はプログラム次第であることを強調しておきたい。

### 3.3.2 並列計算の問題点

ベクトル計算機用のプログラミングは、高速な計算アルゴリズムの設計とそのためのプログラム表現であり、FORTRANのプログラムソース自体は、従来のスカラー計算機のプログラムと何ら差異はない。強いて言えば、ベクトル制御指示行をコメント文として挿入するだけである。ベクトル制御指示行には、機種により様々なもののが存在するが、良くできたベクトルプログラムであれば、回帰演算の有無を指定するもの(第3.2.1節参照)だけで十分である。一方、並列計算機の場合には、並列計算を実現するための独特なプログラム表現が必要となる。現在、並列計算を行うためのプログラム表現方法には、以下のようなものがある。

- 1) コンパイラの機能により並列制御指示行で記述する方法
- 2) MPI (Message Passing Interface)ライブラリーにより記述する方法
- 3) PVM (Parallel Virtual Machine)ライブラリーにより記述する方法
- 4) HPF(High Performance Fortran)により記述する方法
- 5) その他のシステム固有ライブラリーで記述する方法

例えば、原研に主力計算機として導入されているFACOM/VPP-500では、並列制御指示行により並列プログラムを作成する。以下に簡単なプログラム例を示す。

## &lt;&lt; 並列制御指示行による並列化の記述 &gt;&gt;

```

DIMENSION A(100, 400)
!xocl processor p(4) ← PE数の指定
!xocl ip=(p, index=1:400, part=band) ← 分割法の指定
!xocl local A(:, /ip) ← 配列の分割宣言
C
!xocl parallel region ← 並列プロセス生成
TOTAL=0.0
!xocl spread do
DO 200 I=1, 400
DO 100 J=1, 100
    TOTAL=TOTAL+A(J, I)
100 CONTINUE
200 CONTINUE
!xocl end spread sum (TOTAL) ← 分割積算値の総和
!xocl end parallel ← 並列プロセス終了
C
!仮想的な共有メモリーを提供し、通信を明示的に記述する必要は無いが、原理的には分散メモリー
!間の通信であるため、乱用すると計算が遅くなる。

```

この方法の利点は、並列制御指示行はFORTRANのコメント行(1カラム目の記号"!"はコメント行の指示)として挿入されており、これを除けば一般的なFORTRANプログラムソースとして通用し、非並列のスカラーまたはベクトル計算機でもソースプログラムを利用できる点である。逆に欠点は、並列制御指示行はコンパイラが異なる他の並列計算機では通用せず、異機種で並列計算ができなくなることである。

次に、MPIライブラリーを使用するプログラム例を以下に示す。

## &lt;&lt; MPIライブラリーによる記述 &gt;&gt;

```

Include 'mpif.h' ← MPIのためのヘッダ
call mpi_init(ierr) ← MPI環境の初期化
...
call mpi_send(FLUX, 1000, ...) ← データの送信
...
call mpi_recv(XS, 1000, ...) ← データの受信
...
call mpi_finalize(ierr) ← MPI操作終了

```

MPIの特徴はデータの通信を明示的にソースプログラムに書き込むことである。MPIライブラリーは、並列プログラム記述法の標準化における有力候補の1つであり、これをサポートするシステム間で並列計算プログラムを移植することが可能である。しかし、これをサポートしていない並列計算機や従来のEWSなどの非並列計算機では、開発したプログラムを利用できない点が問題となる。

並列計算の技術は、ハード、基本ソフト共に現在は発展途上かつ熾烈な競争時代にあり、マシンと並列化の記述法が統一・標準化されていない。このため、これに対応できるアプリケーションプログラムを安心して開発できる環境とは言いがたい。既に述べてきたように、近年のスーパーコンピュータの本来性能を発揮できるかどうかは、プログラムの最初の配列設計や計算アルゴリズムの設計でほとんど決まってしまうことが多く、今後の並列計算技術の進展には十分留意が必要である。

本報告書の本筋とは関係ないが、並列計算の問題点を以下に列挙しておく。並列計算機用のプログラムを開発する場合には、以下の問題点を十分に意識しておく必要がある。

- 1) プログラムの移植性が悪い。
  - ・並列計算機間の移植性(将来は淘汰、標準化の方向)
  - ・従来スカラーアルゴリズムとの移植性
- 2) ハード、基本ソフト(コンパイラや並列ライブラリー)共に発展途上&戦国時代である。
  - ・現時点での大型コードの開発リスクが大きい。
- 3) プログラム作成が難解である。
  - ・高性能マシンになるほど、本来性能を発揮させるための制約条件が多くなり、アルゴリズム、配列設計などに関してプログラミングが難しい。
  - ・マシンの特性、並列言語/ライブラリー、物理の何れの知識が欠如しても高速なプログラム作成は困難である。
  - ・その他(リストア/リストアファイル作成、I/O関連)
- 4) 共用計算機でのメリットが少ない。
  - ・並列化効率が低いプログラムが横行すると、全体としての効率を落とす(待ち時間が長くなる)。
  - ・要求するPE数の空き待ち時間が長くなる。
- 5) 既存大型コードの並列化が困難
  - ・チューニングレベルで対応ができない(配列、アルゴリズムの再設計が必要)。仮にできても十分な高速化は期待できない。
- 6) プログラム管理が困難となる。
  - ・プログラムが読みにくい。
  - ・制約が多いため、後からの修正や機能追加が困難となる。

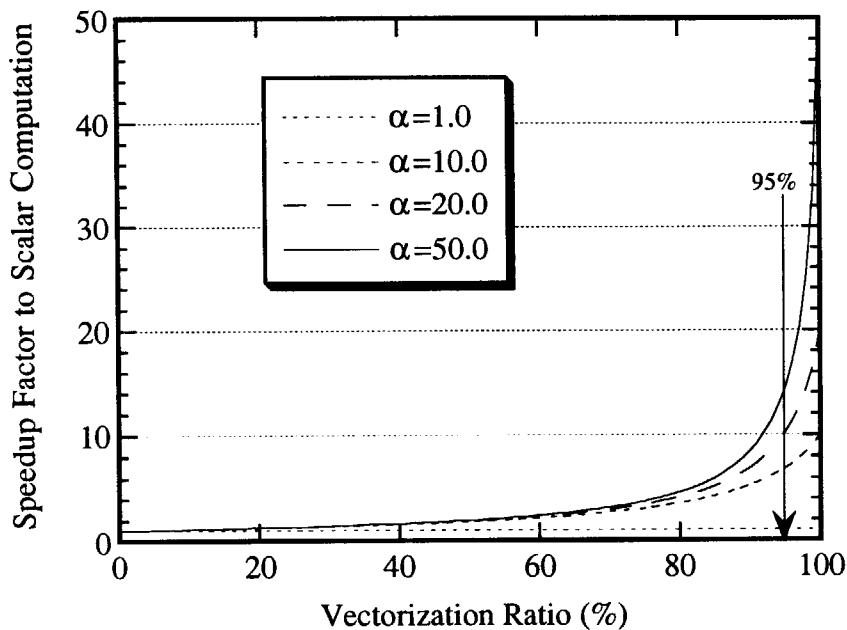


Fig. 3.3.1 Speedup factor depending on vectorization ratio and speed ratio of vector and scalar calculation in vectorized parts ( $\alpha$ ).

Vectorization ratio = ((total CPU time in scalar mode calculation) - (total CPU time in vector mode calculation)) / (total CPU time in scalar mode calculation)

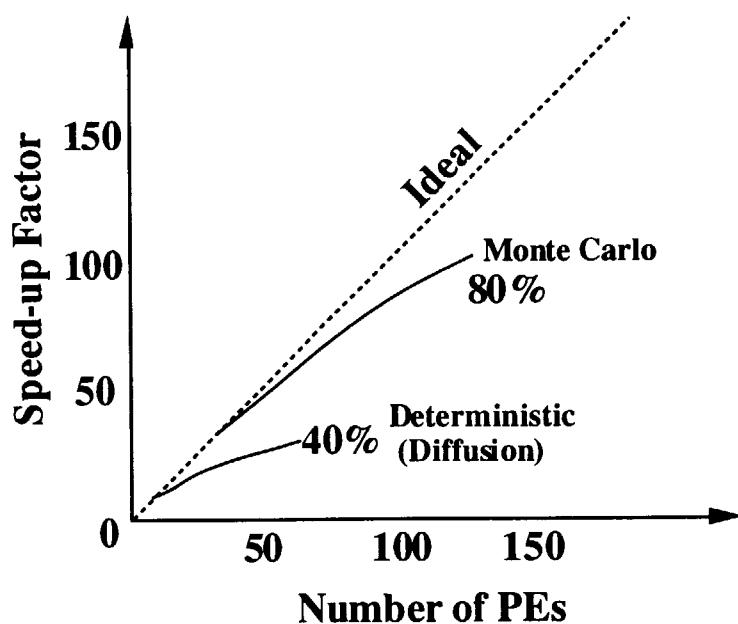


Fig. 3.3.2 The present state of parallel computations in the field of reactor analyses.

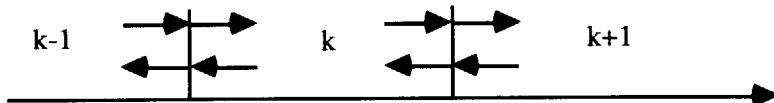
### 3.4 MOSRA-Lightコードの高速計算アルゴリズム

MOSRA-Lightコードは、ベクトル計算機用に開発した高速3次元拡散ノード法コードである。現在のバージョンでは、具体的な並列計算機で動作できるようにしていかないが、アルゴリズムレベルにおいて、共有メモリー型のスカラー/ベクトル並列計算機に対応している。具体的な並列化については、第3.3.2節で述べたように、現状では導入上の問題点が多く、当面は並列計算技術の成熟と標準化を待たなければならない。本節では、MOSRA-Lightコードのために新しく開発したベクトル(並列)計算機のための高速計算アルゴリズムを説明する。

#### 3.4.1 独立した演算のための準備

ベクトル計算と並列計算に共通することは、複数データの同時処理であり、これを実現するためには、独立した演算が可能な条件を与えることである。MOSRA-Lightでは、以下のようにして演算の独立性を実現している。

- 1) 前処理として、全てのノードにノード幅や断面積データの情報を持たせる。これは、以降の演算を独立して行える利点の他、メモリーアクセスの競合を避けるためにも有効である。
- 2) 全てのノードに対して6面の部分中性子流を隣接ノードと独立して与える。不連続因子を扱う場合には、必然的に部分中性子流を隣接ノードと別に与える必要があるが、このことはノード単位に部分中性子流を利用する計算を他のノードと独立して行える利点がある。



- 3) 各ノード毎に、隣接するノードの有無または境界条件の種類に関する情報を持たせる。幾何形状や境界条件に関する判定文を反復計算の最深部において繰り返し行うこと为了避免するため、反復計算に入る前に各ノードに情報を与えておく。具体的には、X+, X-, Y-, Y+, Z-, Z+の各面に隣接する境界条件を以下のように6桁の整数値(N)で表している。

$$N = I_{z-} * 10^5 + I_{z+} * 10^4 + I_{y-} * 10^3 + I_{y+} * 10^2 + I_{x-} * 10^1 + I_{x+} \quad (3.4.1)$$

ここで、各桁の整数値は以下に示す規則に従って、添字で示すノード表面に隣接する境界条件の種類または隣接ノードの存在を表す。

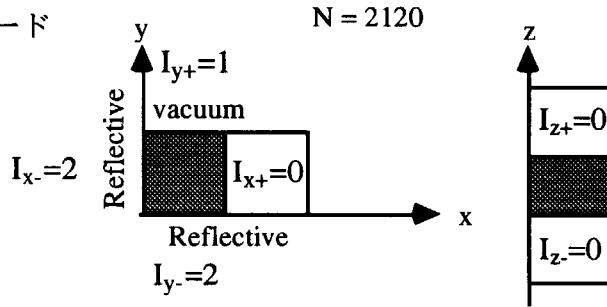
$I=0$  : 黒体ではないノードが存在

= 1 : 真空境界条件または黒体ノード

= 2 : 反射境界条件

= 3 : 周期境界条件

= 4 : 90度回転境界条件



例えば右図のハッチングノードに  
対しては、 $N=2120$ が与えられる。

ただし、黒体ノードに対しては、エネルギー群毎に  $I=0/1$  を変更する。また、対角対称境界条件を使用する際には、対角対称性を考慮して隣接ノードの有無と境界条件を与える。

- 4) ノード内のTransverse Leakage分布を表現するのに2次式近似(QTLA: 第2.6.3節参照)を使用する。QTLAは、着目するノード内のTransverse Leakageの分布を2次式で表現する際、隣接するノードの部分中性子流を利用する。チェックカーボードスウェープでQTLAを使用すると回帰演算とならないため、ベクトル計算や並列計算に適している。ただし、QTLAが利用できない真空境界や黒体に接するノードに対しては、片側にノードが存在する場合には1次式で近似し、両側にノードが存在しない場合にはバックリング近似(第2.6.2節参照)を使用する。

以上に述べた他、MOSRA-Lightではベクトル計算機の性能を最大限に発揮させるためのプログラミング技法(第3.2.1節参照)を駆使している。

### 3.4.2 境界分離チェックカーボードスウェープ

MOSRA-Lightの最大の特徴は、ベクトル計算機のために開発した独自の高速計算アルゴリズム(境界分離チェックカーボードスウェープ)を採用したことである。以下では、このアルゴリズムについて説明する。Fig.3.4.1は、多項式展開ノード法の計算フローの概略を示したものである。全体の計算プロセスは、以下のようないくつかの段階に分割される。

#### (a) 応答行列の作成

ノード毎に与えられた断面積とノードサイズから、ノード毎に応答行列  $[P_g^k]$  及び  $[R_g^k]$  を作成する。

#### (b) 初期値の作成

全ノードに対する  $J_g^{out,k}$ 、 $J_g^{in,k}$ 、 $\Phi_g^k$  及び  $k_{eff}$  の初期値を与える。

#### (c) ソース項の作成または更新

全ノードに対するソース項  $Q_g^k$ (核分裂中性子源+減速中性子源)を作成または更新する。

#### (d) $L_g^k$ の作成または更新

QTLAに基づき  $\mathbf{L}_g^k$  を全ノードに対して作成または更新する。この際、境界条件に隣接するノードに対しては、例外処理(第2.8節参照)が必要となる。

(e)  $\mathbf{J}_g^{out,k}$  の更新

応答行列式により、与えられた  $\mathbf{J}_g^{in,k}$  から  $\mathbf{J}_g^{out,k}$  を更新する。

(f)  $\mathbf{J}_g^{in,k}$  の更新

更新された  $\mathbf{J}_g^{out,k}$  を使用して、隣接ノードの  $\mathbf{J}_g^{in,k}$  を更新する。この際、境界条件に隣接するノードに対しては、例外処理(第2.8節参照)が必要となる。

(d)～(f)のプロセスでは、ノードに関して演算を独立にするため、チェックカード(red-black)スウェーブを行う。すなわち、先ず全てのblackノードに対して  $\mathbf{L}_g^k$  を求める。次に、応答行列式により全blackノードの  $\mathbf{J}_g^{out,k}$  を更新し、これを用いて、隣接するredノードの  $\mathbf{J}_g^{in,k}$  を更新する。次に、redノードを対象にして同様の計算を行う。

(g)  $\Phi_g^k$  の計算

内部反復計算毎に得られる  $\mathbf{J}_g^{in,k}$ 、 $\mathbf{J}_g^{out,k}$ 、 $\mathbf{Q}_g^k$  から  $\Phi_g^k$  を計算する。 $\Phi_g^k$  が収束するか、内部反復回数の上限値に至るまで、(d)～(g)を繰り返す。

(h) 外部反復収束判定

全エネルギー群について内部反復計算が終了したら、 $k_{eff}$  と  $\Phi_g^k$  の収束判定を行う。収束していない場合は(c)に戻る(外部反復計算)。

以上の(a)～(h)の計算プロセスは、何れも複数のノードに対して独立して処理することができ、全般的にはベクトル計算や並列計算に適したアルゴリズムと言える。ベクトル計算において問題になるのは、(d)と(f)の処理に含まれる、境界条件に隣接するノードに対する例外処理である。例外処理は、FORTRANプログラムではIF分岐として表現するため、ノード数をベクトル長とするDO文が有効にベクトル処理できなくなる。多様な境界条件を扱うほどIF分岐は複雑なものになり、ベクトル化が難しくなる。

境界分離チェックカードスウェーブは、通常の方法ではベクトル化が困難な(d)と(f)の処理の高速化を狙ったアルゴリズムである。Fig.3.4.2にその考え方を示す。同図では、計算体系を2次元のチェックカードで示してあるが、実際には3次元のチェックカード(Fig.3.2.2)を使用する。

先ず、無駄な空間スウェーブを避けるため、計算に必要がないノード(Fig.3.4.2において右上の分離したノード)を計算対象から除外する。以降ではこれを真空ノードと呼ぶことにする。実際、原子炉は円筒形をしていることが多く、このような扱いは合理的である。逆に真空ノードが扱えないコードでは、反射体物質などで全体系を埋める必要が生じ、余分な計算労力と収束の遅れを生じる。MOSRA-Lightでは、前節で述べたように、全てのノードが隣接ノードの有無や境界条件を知っているため、このような扱いが可能である。

空間スウェーブは3次元で行うが、中性子束や部分中性子流などのノードパラメータの位置は、プログラム上は全て1次元配列として与える。例えば、

$\phi(i, j, k, n)$ 、 $i$  : x方向座標、 $j$  : y方向座標、 $k$  : z方向座標、 $n$  : エネルギー群とするのではなく、

$\phi(m, n)$ 、 $m$  : 3次元空間の位置を表すノード番号、 $n$  : エネルギー群とする。このようにすることにより、ベクトル長を極めて長くすることができる。ノードの空間位置を1次元的に表現するため、予めノードに連続した番号を付ける。この際、Fig.3.2.2に示すように以下の順に番号を付ける。

- (a) blackノードで境界に接しないノード(例外処理が必要ないノード)
- (b) blackノードで境界に接するノード(例外処理が必要なノード)
- (c) redノードで境界に接しないノード(例外処理が必要ないノード)
- (d) redノードで境界に接するノード(例外処理が必要なノード)

空間スウェープは、このノード番号に基づいて、チェックカーボードスウェープ(black->red->black->red...)により行う。ただし、(a)と(b)及び(c)と(d)をそれぞれ分離したDOループとして扱うことにより、外部境界に接しない体系中の大部分のノードである(a)と(c)に対する処理は、IF分岐を含まないため、極めて高速なベクトル計算が可能になる。一方、(b)と(d)のノードについては、ベクトル処理が有効でない部分として残る。しかし、例外処理が必要なノード数の割合は、体系体積に対する表面積の比に比例するため、体系が大きい問題であるほど小さくなる。逆に言えば、計算時間が掛かるノード総数が多い大規模な問題になるほど、この方法によるベクトル計算の効率は高くなり、計算時間を短縮することができる。

体系内に黒体ノードが存在する場合には、黒体部分を境界条件と同様に扱い、これに接するノードは例外処理ノードに分類して扱う。

### 3.4.3 並列計算機への適用性

現在のMOSRA-Lightコードは、具体的な並列計算機には対応していないが、アルゴリズムレベルにおいて、共有メモリータイプの並列計算機に対応している。これは、ベクトル計算と並列計算は、独立した演算を一括処理するという点において共通しているためである。MOSRA-Lightコードのプログラミングでは、DOループ内部はノードに関して独立した演算で構成されており、ノードに関するDO文を単純にプロセッサ数で均等に分割することにより、優れた並列計算効率が期待できる。なぜなら、MOSRA-Lightは、断面積などの計算に必要な情報をノード毎に独立して与えており、メモリーアクセスの競合が頻繁に生じないこと、また、同じ演算内容を含むDO文の分割は各プロセッサの計算負荷を均等にすることができるためである。並列計算機では、前節で述べたようなベクトル計算で高速処理の対象とならないIF分岐を含む例外処理部分も高速化できる点が魅力である。しかし、ベクトル並列計算機の場合には、DO文の分割によりベクトル長が短くなるため、ベクトル計算効率の低下を招くことになる。したがって、並列化による高速化とベクトル効率の低下

との競合により最適なプロセッサ数が問題毎に異なってくると考えられる。残念ながら原研には共有メモリータイプの高並列マシンが導入されていないことと、並列計算自体の問題点(第3.3.2節参照)のため、MOSRA-Lightの並列計算機版は現在のところ作成していない。

分散メモリータイプの並列計算機の場合には、MOSRA-Lightの手法をそのまま利用することは効率的ではない。このタイプの並列計算機で最も重要視すべきことは、プロセッサ間のデータ通信を極力避けることである。現在のMOSRA-Lightの手法で問題となるのは、ノードの番号付けが空間的な配置とは関係なくベクトル計算上の都合で付けられていることである。このため、QTLAの計算と隣接ノードの $J_g^{in,k}$ を更新する際(すなわち隣接ノードの情報を参照したり書き換えたりする場合)に、プロセッサ間の通信が頻繁に生じることが予想される。これを避けるためには、むしろ同一プロセッサのメモリー内で処理できるノードをグループ化するような番号付けが望ましい。また、ベクトル並列計算機の場合には、ベクトル計算と並列計算のどちらを優先させるかも問題となる。例えば、水平方向の2次元平面について境界分離チェックボードスウェーブによるベクトル計算を行い、軸方向に対して分散並列処理を行う(軸方向ノード間で通信が必要)ことも考えられる。いずれにせよ、ノード法に限らず隣接メッシュとのデータの交換を反復計算の深部で行う計算手法は、分散メモリータイプの高並列計算機にはあまり向いておらず、新しい並列計算アルゴリズムの開発が望まれる<sup>23)</sup>。

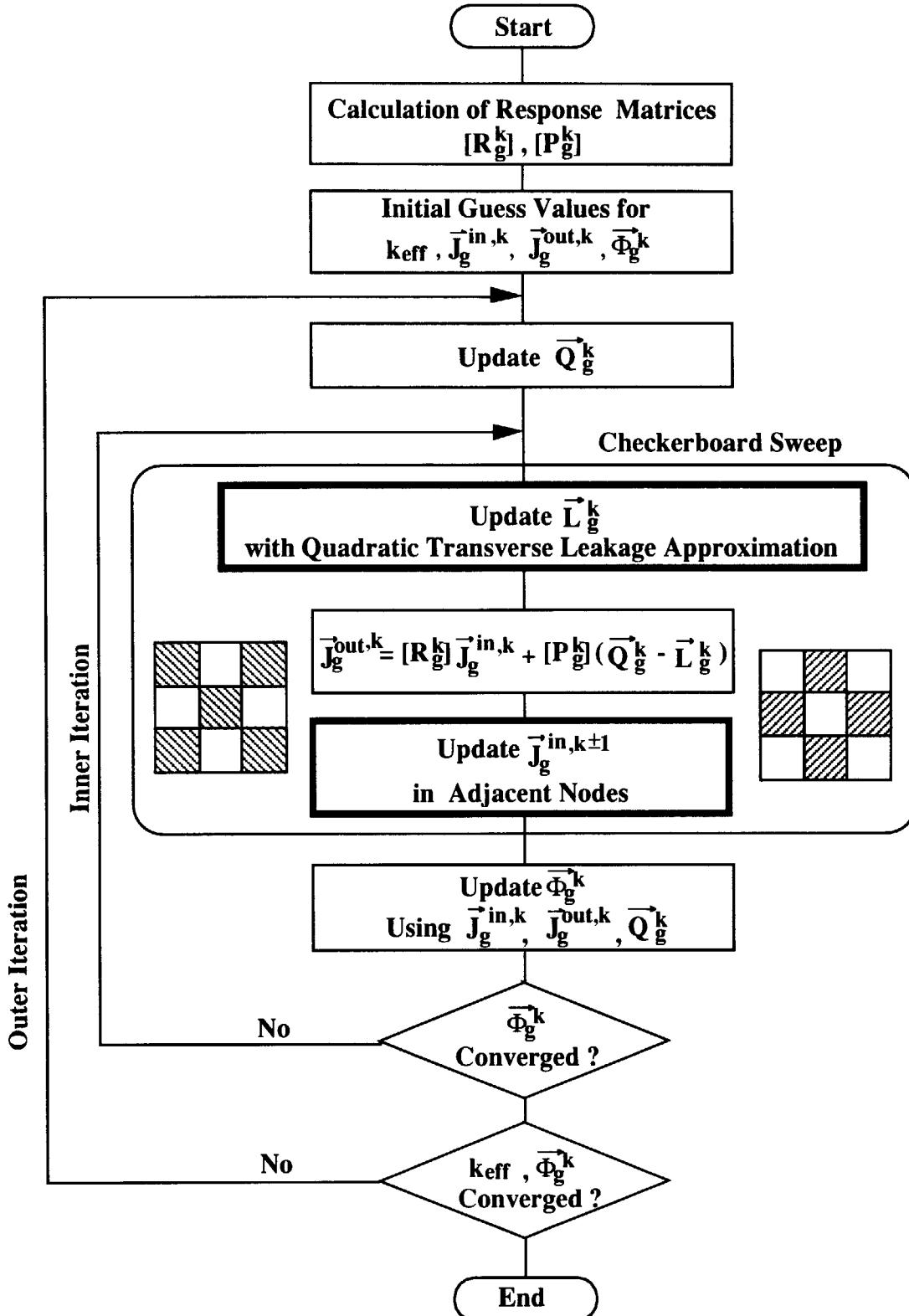
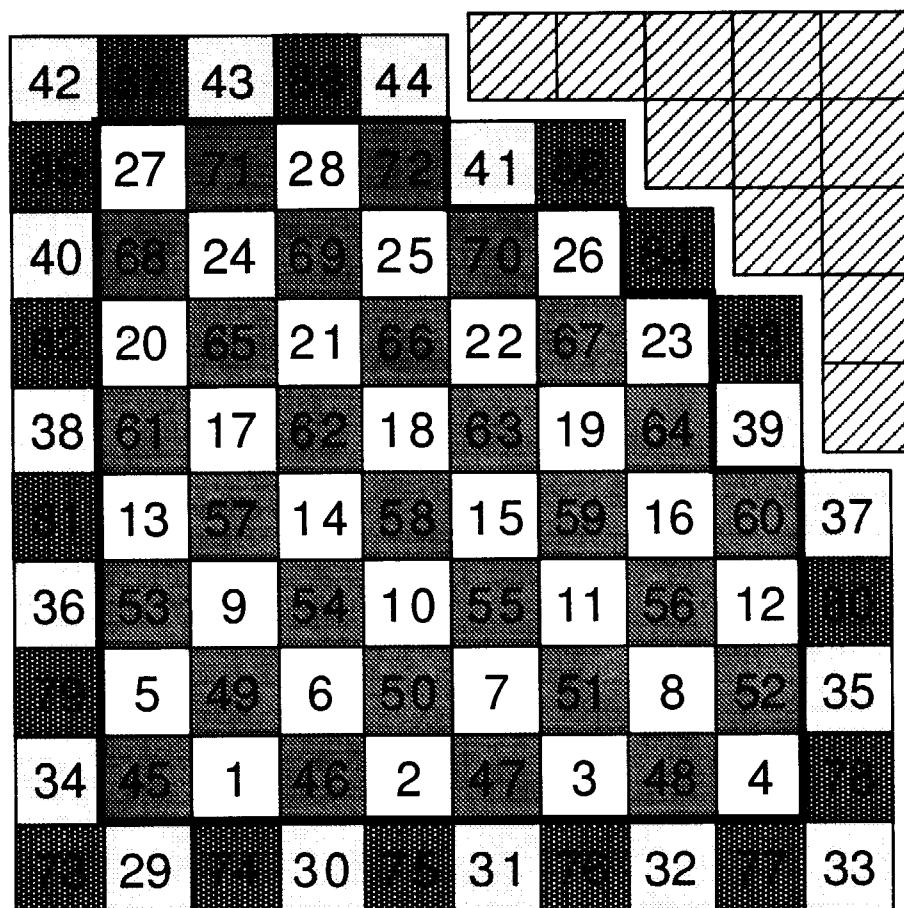
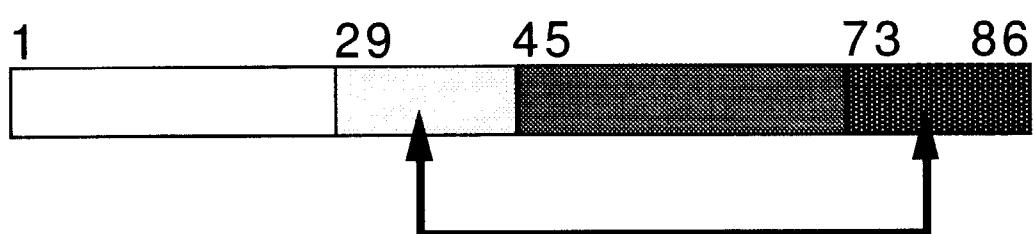


Fig.3.4.1 Outline of NEM (Nodal Expansion Method) calculation



**checker-board sweep with partial neutron currents**



**Exceptional treatments are necessary  
according to boundary conditions.  
(not vectorized because of if -branch)**

Fig.3.4.2 Boundary separated checkerboard sweep for vector computers

## 3.5 MOSRA-Lightコードの高速計算性能

### 3.5.1 IAEA3次元ベンチマーク問題

MOSRA-Lightコードの高速計算性能とその精度を確認するため、IAEAの3次元ベンチマーク問題<sup>24)</sup>の解析を行った。このベンチマーク問題は、拡散ノード法コードの検証計算に世界的にも広く利用されているものである。計算体系をFig.3.5.1に、2群断面積データをTable 3.5.1に示す。炉心は2種類の燃料タイプで構成され、内側燃料領域には、4箇所に制御棒が挿入される。その内1領域は炉心上部から80cmの所に制御棒先端が位置する部分挿入状態である。参照解としては、ベンチマークブック<sup>24)</sup>に提供されているVENTUREコードの計算結果を利用することができる。VENTUREコードは有限差分法による拡散コード(CITATIONの改訂版)であり、参照解はメッシュ分割を(9×9×10)、(17×17×19)、(34×34×38)、(68×68×76)、(102×102×114)の5通りに変えたVENTUREの計算結果を基に、メッシュ幅ゼロの点に外挿して得られた信頼性の高いものである。最もメッシュ幅が小さいケースでは、1メッシュの大きさは1.67cm×1.67cm×3.33cmであり、当時の計算機(IBM-360/91)で約6時間の計算時間(CPU)を要したと報告されている。

MOSRA-Lightコードによる計算は、全方向共にメッシュ幅を20cm(1メッシュ/集合体)、10cm(4メッシュ/集合体)、5cm(16メッシュ/集合体)に変えた3通りの計算を行い、メッシュ幅に対する計算精度の依存性を調べた。ここでの計算は、富士通のベクトル計算機であるFACOM VP-2600を使用して行った。この計算機は既に原研からは撤去されベクトル並列計算機であるFACOM VPP-500に置き換えられているが、単一プロセッサの実効性能としてはVPP-500の約2倍の速度性能を有している。

実効中性子増倍率と集合体平均出力分布について、MOSRA-Lightコードによる計算結果とVENTUREコードによる参照解との比較をFig.3.5.2に示す。同図では、メッシュ幅が20cmのケースと10cmのケースの計算結果を示してあるが、どちらのケースにおいても中性子増倍率と出力分布の計算結果は参照解とよく一致しており、MOSRA-Lightコードの粗メッシュ計算法が妥当であると判断することができる。更にベクトル計算性能を測定するため、MOSRA-Lightのソースプログラムをスカラーモードでコンパイルし、同様な計算を同じ計算機で行わせることにより、ベクトル化率とベクトル化による速度向上率(=スカラーロードモジュールによる計算時間/ベクトルロードモジュールによる計算時間)を測定した。メッシュ幅が20cm、10cm、5cmの各ケースにおける計算精度と計算速度をTable 3.5.2にまとめてある。中性子増倍率の計算精度については、参照解との差異は何れの場合も0.006%以下でありメッシュ効果は無視できる。出力分布の差異については、メッシュ幅が20cmの場合において平均で0.6%、最大でも1.3%程度であり実用上十分に小さい範囲である。メッシュ幅10cm以下では、出力分布の最大誤差は約0.5%以下と小さく、計算精度はほとんど変わっていない。すなわち、与えられた収束判定条件内で精度飽和状態にある。所要計算時間(CPU時間)は20cmメッシュの場合で2秒程度であり、対角対称境界条件を使用すれば更に短縮さ

れる。メッシュ幅が5cmの場合でも数分以内で計算をすることができる。ベクトル効率は極めて高く、20cmノードを使用した場合でも99%以上であり、ノード数が更に多くなると100%近くに達する。このため、スカラー計算に対して20倍～40倍以上の高速計算が可能である。MOSRA-Lightコードの最大の特長は、ノード数が多い場合ほど、すなわち計算コストが高い問題になるほど、ベクトル長が長くなるとともに境界分離チェックボードスウェーブが有効に働くため、高速計算性能が一層発揮されることである。ちなみに、ここでの計算では、収束制御パラメータ(内部反復回数上限値など)を安定収束を優先させるデフォルト値に設定しており(第4.2.7節参照)、この制御パラメータを適切に調整することにより更に計算時間を数分の1に短縮させることができる。

### 3.5.2 MOSRA-Lightコードにおける計算時間の分布

コードの計算時間を一層短縮するためには、先ず、サブルーチンなどの単位で、典型的な問題に対する動的な計算時間の分布を測定することが有効である。これにより、コストの高いサブルーチンを同定し、優先的にチューニングを行うわけである。MOSRA-Lightコードの一層の高速化の可能性を探るため、前節のIAEA3次元ベンチマーク問題における計算コストの分布を評価してみた。計算は、ノード幅を20cmとして、FACOM VP2600により行った。

NEMによる計算は、以下のように大別される(第2.9節及びFig.3.4.1参照)。

- (a) 応答行列 $[P_g^k]$ 、 $[R_g^k]$ の作成(第2.5節参照)
- (b) 散乱源を含む源項 $Q_g^k$ の更新(2.9.3式参照)
- (c) トランスバース漏洩項 $L_g^k$ の更新(第2.6節参照)
- (d)  $J_g^{out,k}$ の更新(2.5.27式参照)
- (e)  $J_g^{in,k}$ の更新(第2.7節、第2.8節参照)
- (f)  $\Phi_g^k$ の更新(2.5.7、2.5.19、2.5.20式参照)
- (g) 収束判定、核分裂源の更新(2.9.2式)、加速パラメータ計算など
- (h) その他(主として入出力)

サブルーチン単位で計測された計算時間を、上記の8つの区分に集計した。スカラーモードで計算した場合とベクトルモードで計算した場合のコスト分布を比較して、Fig.3.5.3に示す。スカラー計算では、(c)～(f)の部分が大きなコストを占めるが、これらは内部反復計算に相当する部分であり当然の帰結である。これらの中でも、特にトランスバース漏洩項の計算部分が大きなコストとなっている。一般に、単純な加減乗除に比べ、論理IF文の処理の方がコスト高となり、MOSRA-Lightが扱える境界条件の種類の豊富さにより、トランスバース漏洩項の計算に複雑な分岐処理が多く存在するためと考えられる。

ベクトルモードによる計算では、最も計算コストが高い内部反復の部分を、ベクトル処

理により極めて高速化していることが、Fig.3.5.3により分かる。一般に、ノード幅をより小さくすると、安定に計算を収束させるためにはより多くの内部反復回数が必要となる。MOSRA-Lightでは、ノード幅を小さくするほど、ベクトル長が長くなるとともに境界分離チェックボードスウェーブが効力を発し、内部反復回数の増加による計算コストの増大が抑制される。

Fig.3.5.4に、ベクトル計算モードによるコスト分布を示す。これは、Fig.3.5.3におけるベクトル計算部分の拡大図である。ベクトル計算では、総計算時間のおよそ半分をトランスマルス漏洩項の計算に費やしている。また、そのおよそ75%が境界に隣接したノード、即ちベクトル処理できないノードの処理に費やされている。逆に言えば、ベクトル処理できる部分の高速化がうまく働き、できない部分だけが残されたわけである。これ以上の高速化を図るためにには、最大コストとなっている境界隣接ノードのトランスマルス漏洩項の計算を高速化する必要がある。しかし、それはベクトル化できない部分であり、これらがベクトル化による高速化の限界といえる。

共有メモリタイプのベクトル並列計算機では、更なる高速化が期待できる。トランスマルス漏洩項の計算は、DOループ中のIF分岐の存在によりベクトル化できないのであり、チェックボードスウェーブの採用により回帰演算にはなっていない。従って、DO文の分割により、容易に効率的な並列化が可能である。スカラー並列計算機の場合には、高い並列化効率が期待できるが、ベクトル並列計算機では、DO文の分割によりベクトル長が短くなる。したがって、並列化によるトランスマルス漏洩項計算の高速化とベクトル長が短くなる事による全体計算のベクトル効率の低下との兼ね合いにより、最適な並列数が決まる。これは、ノード総数や境界隣接ノード数など、対象とする問題に依存する。どのような手法が良いかは、マシンの性能とユーザーに与えられたマシンの使用環境に強く依存する。

### 3.5.3 有限差分コード(CITATION)との比較

IAEAの3次元ベンチマーク問題には、VENTUREコードによる信頼性の高い参照解が与えられている。しかし、計算自体は10年以上も前の計算機で行われており、計算速度に関する議論にはあまり役立たない。そこで、MOSRA-Lightコードの計算に使用したものと同じ計算機(FACOM VP-2600)を使用して、有限差分コードCITATIONによるベンチマーク計算を行った。実際に利用したコードは、SRAC95コードに組み込まれたCITATIONである。SRAC-CITATIONは、チェックボードスウェーブによるベクトル化チューニングが行われているが、ここでは、オリジナルのCITATIONと同じアルゴリズムを使用するスカラー版を使用した。SRAC-CITATIONは、メッシュ数などに関わる配列数の制限が取り払われており、メッシュ効果を調べるの便利である。

CITATIONによる計算は、メッシュ幅を各方向で同じとし、20cm(1×1/集合体)、10cm(2×2/集合体)、6.66cm(3×3/集合体)、5cm(4×4/集合体)、2cm(10×10/集合体)の5ケースについて行った。収束判定は、MOSRA-Lightによる計算と同じ条件(固有値について1.0E-5以下、

ポイントフラックスについては $1.0E-4$ 以下)とした。参照解を提供しているVENTUREの計算とは、メッシュ分割の仕方、収束条件(ポイントフラックスで $1.0E-5$ 以下)が若干異なっている。

Fig.3.5.5は、VENTUREによる参照解が得られた方法と同様にして、CITATIONによる中性子増倍率の計算結果をメッシュ幅ゼロに外挿したものである。両コードによる外挿点での中性子増倍率の差異は、僅かに0.002%であり、参照解の妥当性が改めて確認された。Fig.3.5.6(実線)は、CIATTIONにより得られた中性子増倍率の誤差のメッシュ幅依存性を示したものである。メッシュ幅が0~10cmの範囲で、中性子増倍率の誤差は比較的小さく、0.05%以内である。しかし、カーブの形からも分かるように、これは空間積分における誤差の相殺によるものである。一般的には、メッシュ幅が大きくなるほど誤差は単調に増加する。この問題の場合には、制御棒の挿入により、誤差の相殺が生じていると考えられる。実際、Fig.3.5.7(実線)に見られるように、集合体平均出力分布のメッシュ効果は非常に大きく、誤差はメッシュ幅とともに単調に増大している。有限差分法コードでメッシュ幅を10cmとする計算では、水平方向出力分布を最大で23%、平均(R.M.S)で9%もの誤差を生じている。

Fig.3.5.6及びFig.3.5.7には、CITATIONによる所要計算時間(破線)も示してある。MOSRA-Lightによる計算では、メッシュ幅が20cmでの出力分布の最大誤差は、僅かに1.3%であり、これに要する計算時間は約2秒である(Table 3.5.2)。一方、これと同等な精度をCITATIONにより得るためにには、約1cmのメッシュ幅を使用し、およそ2万秒(5.6時間)もの計算時間を要することになる(Fig.3.5.7)。即ち、CITATIONに比べてMOSRA-Lightは、数千~数万倍の高速化を実現していると言える。これは、近代ノード法の効果として数百倍、更にベクトル化による効果として数十倍の高速化が達成されるためである。もちろん、高速炉のように、中性子の平均自由行程が長く、比較的粗いメッシュ分割でも十分な精度が得られる場合には、有限差分法に対する優位性は減退し、これほどの高速化は期待できない。

現実問題では、たった1回の炉心計算に数時間もの計算時間を掛けることはほとんど無い。粗いメッシュで計算し、Fig.3.5.6のようなメッシュ効果の補正を行う必要がある。炉心燃焼計算や空間依存動特性計算のように、メッシュ効果の補正が事実上できない場合には、別の近似や経験を導入した手法を導入するか、ある程度の誤差を許容することになる。有限差分法は、原理的には正確な解を与えるが、計算資源の制約を考えると、近代ノード法の方がはるかに正確な解を与えてくれる。1970年代に、原研においてCITATIONを利用した空間依存動特性コードが開発された。当時としては画期的であるが、現在の要求精度に耐えられるものとは考えがたい。制御棒半挿入を模擬したIAEAの3次元ベンチマーク問題は、PWRにおける制御棒引き抜き事故の一過程と考えられる。メッシュ幅を10cmとして、1つのタイムステップに20秒の計算時間を掛けたとしても、集合体出力分布に23%もの誤差が入ることになる。出力分布の誤差は、そのまま燃料温度の誤差として反映される。モンテカルロ法と同様に、計算速度の向上は、作業効率を上げるだけではなく、計算精度の向上と等価である。

Table 3.5.1 Two group constants for IAEA benchmark problem<sup>24)</sup>

Region	D <sub>1</sub>	D <sub>2</sub>	$\Sigma_{1 \rightarrow 2}$	$\Sigma_{a1}$	$\Sigma_{a2}$	$v\Sigma_{f2}$	Material
1	1.5	0.4	0.02	0.01	0.08	0.135	Fuel 1
2	1.5	0.4	0.02	0.01	0.085	0.135	Fuel 2
3	1.5	0.4	0.02	0.01	0.13	0.135	Fuel 2+Rod
4	2.0	0.3	0.04	0.0	0.01	0.0	Reflector
5	2.0	0.3	0.04	0.0	0.055	0.0	Refl.+Rod

Table 3.5.2 Mesh Effect in IAEA 3D benchmark calculation by MOSRA-Light code  
(convergence criteria ;  $\Delta\Phi \leq 10^{-4}$ )

Size of node (cm)	20	10	5
Number of node	1311	9158	73264
Error* of k-eff (%)	-0.006	+0.006	+0.004
Error* of assembly power (%)			
Maximum	-1.3	+0.51	+0.41
Average	0.59	0.23	0.28
CPU time by FACOM/VP-2600 (Diagonal symmetric option)	2sec.12 (1sec.37)	14sec.09 (8sec.21)	3min.6sec.75 (1min.57sec.27)
Vectorization Ratio	99.3	99.7	----**
Speed-up factor to scalar calculation	23	36	----**

\* Deviation from the reference solution by VENTURE code

\*\* Scalar calculation was not done because of too much CPU time.

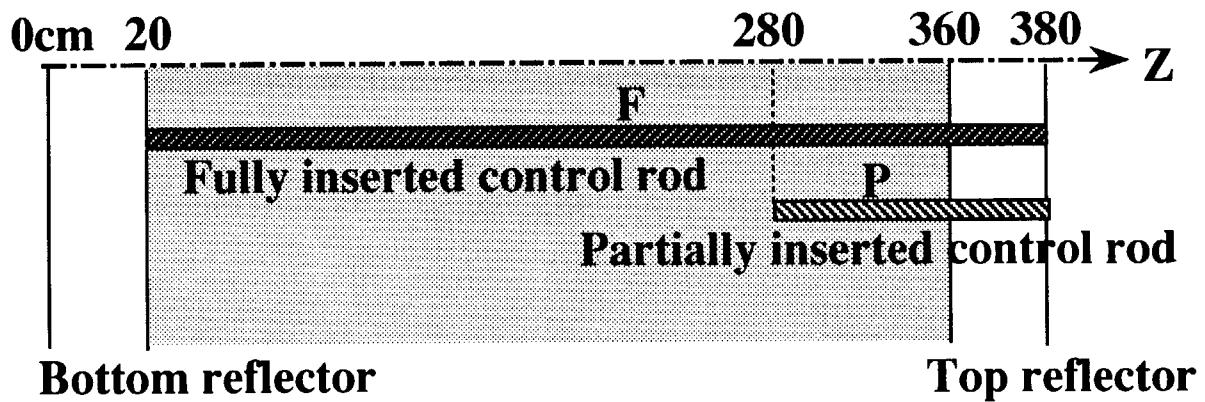
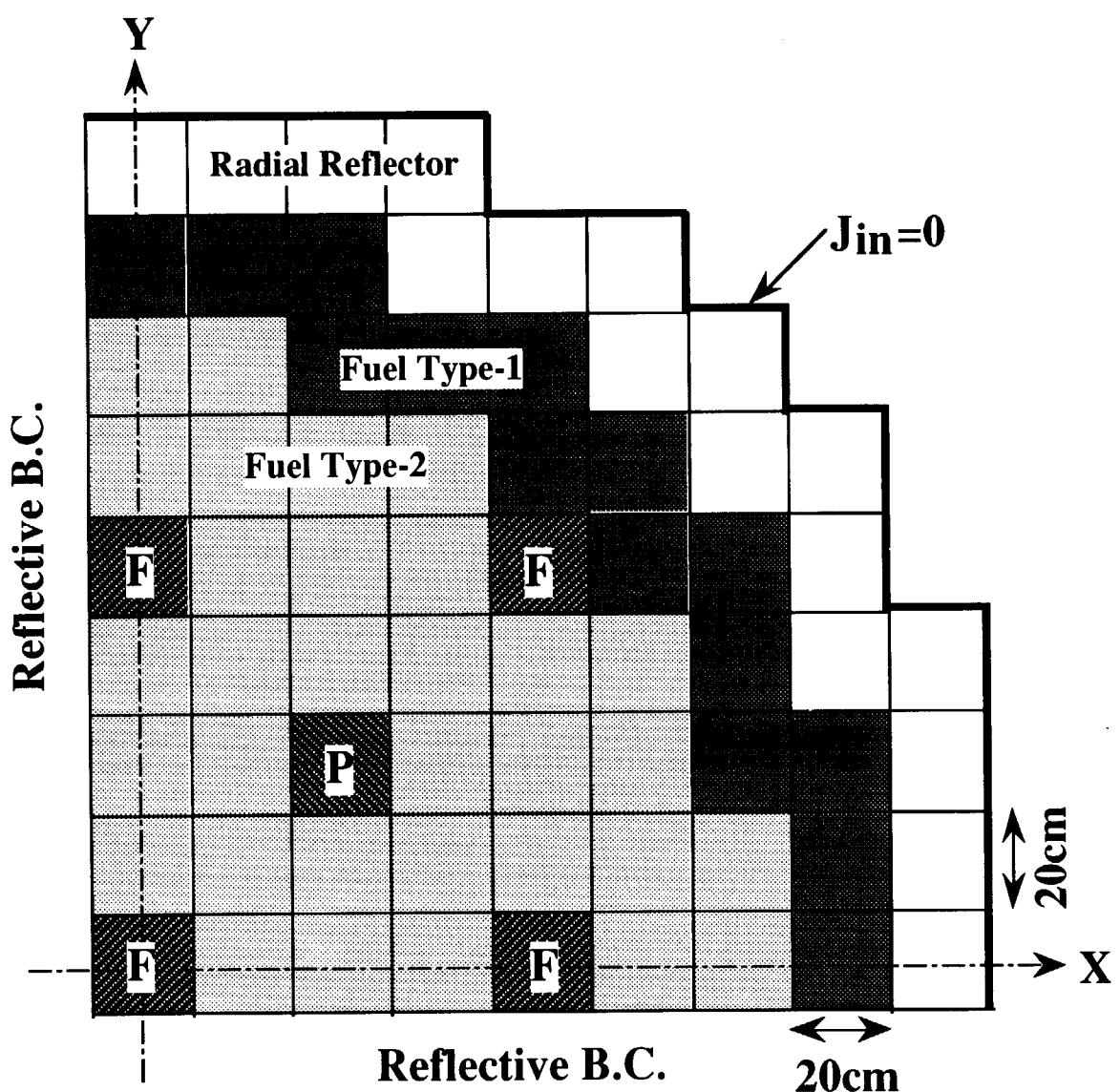


Fig.3.5.1 Core geometry of IAEA 3-dimensional benchmark problem

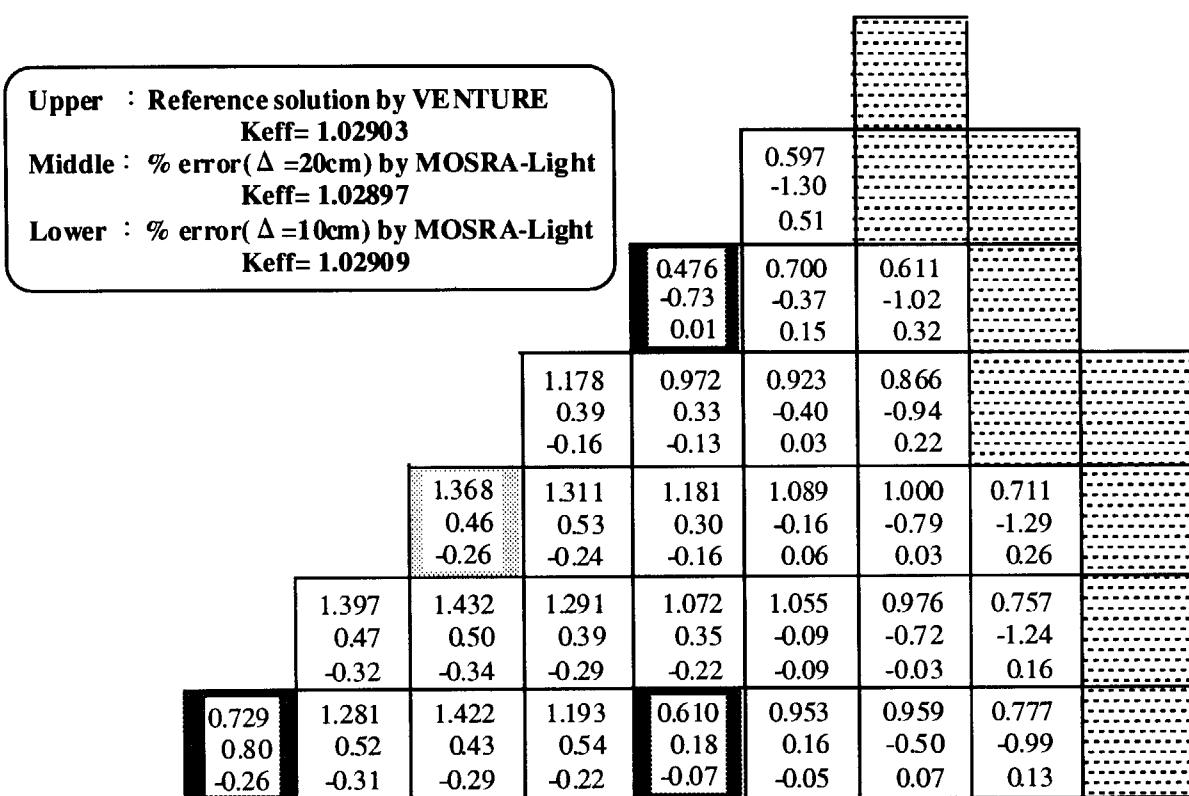


Fig.3.5.2 Comparisons of k-eff and assembly averaged power distribution in IAEA 3D benchmark

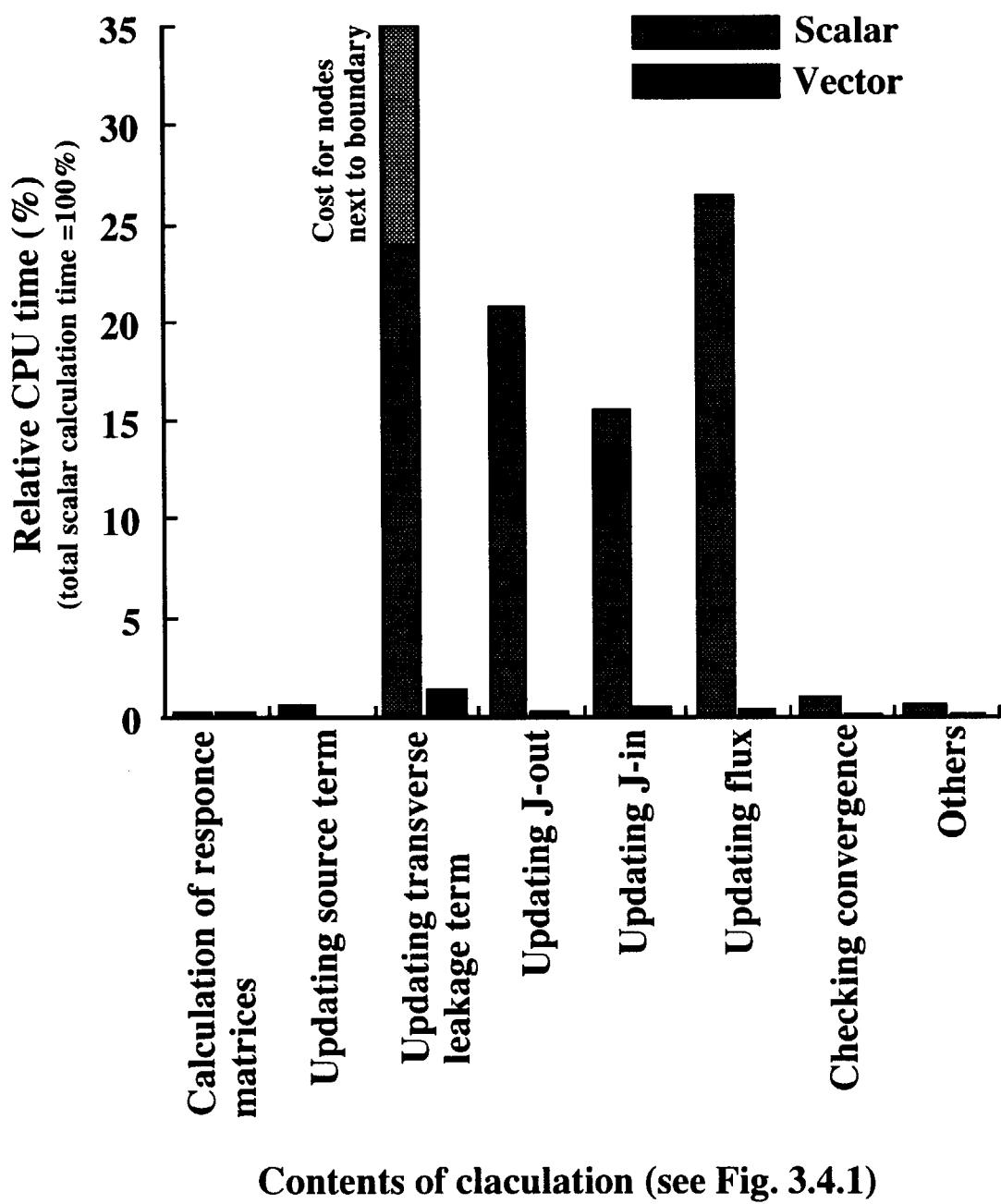


Fig.3.5.3 Distribution of computation time in IAEA 3D benchmark calculation by MOSRA-Light

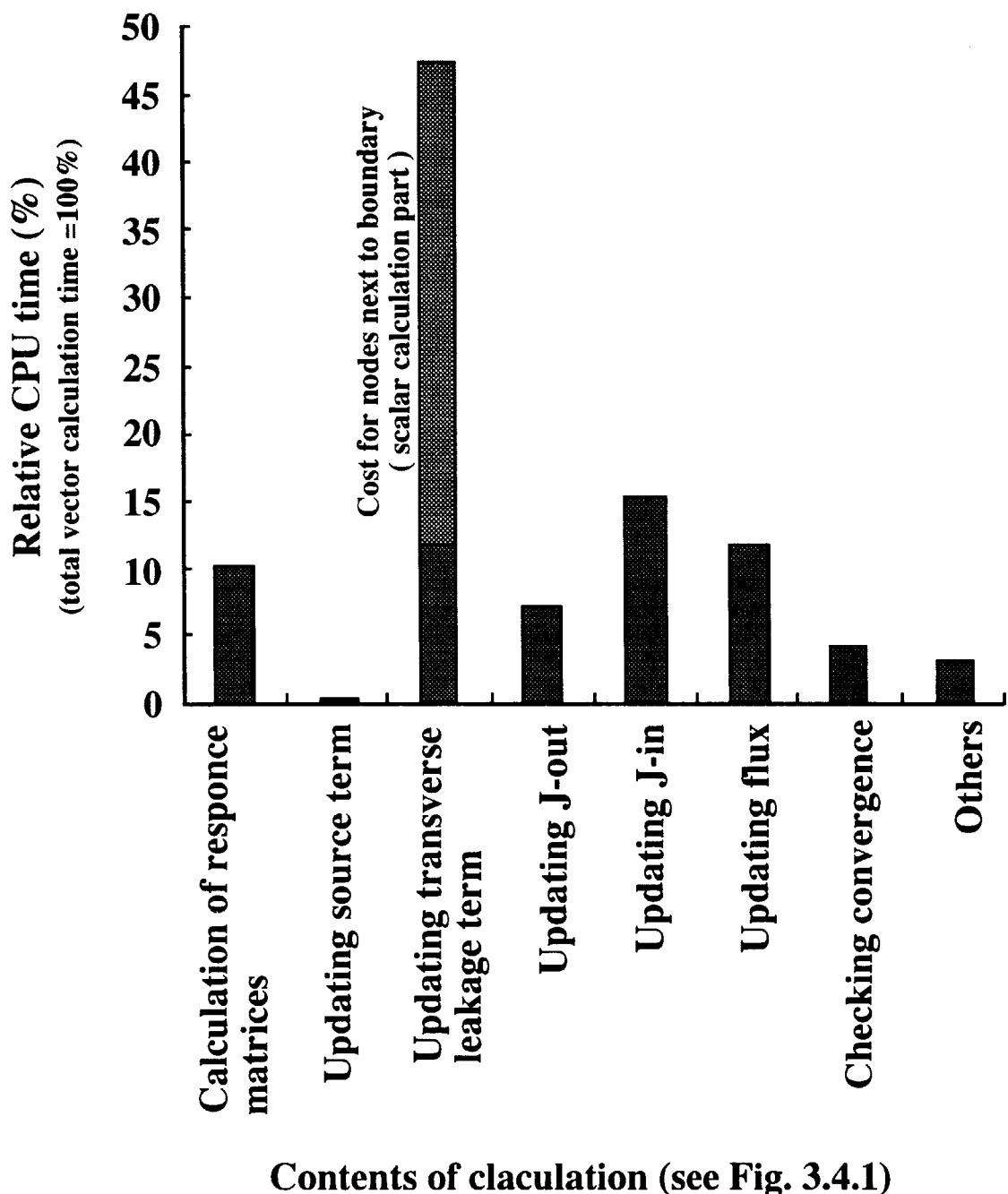


Fig.3.5.4 Distribution of computation time in IAEA 3D benchmark calculation by MOSRA-Light (vector mode)

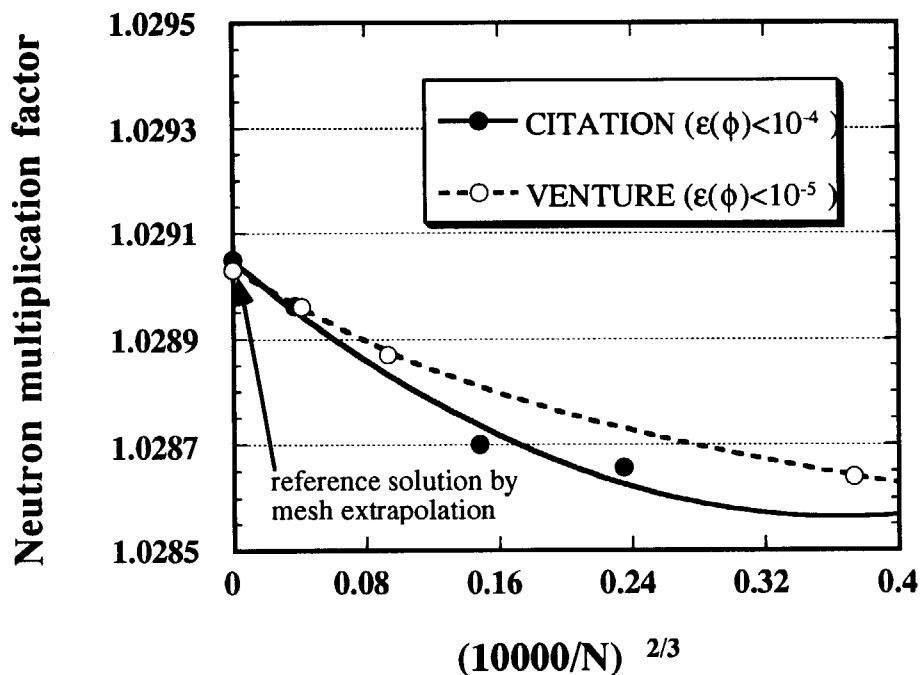


Fig.3.5.5 k-eff dependence on number of mesh points (N) in IAEA 3D benchmark calculation by CITATION and VENTURE

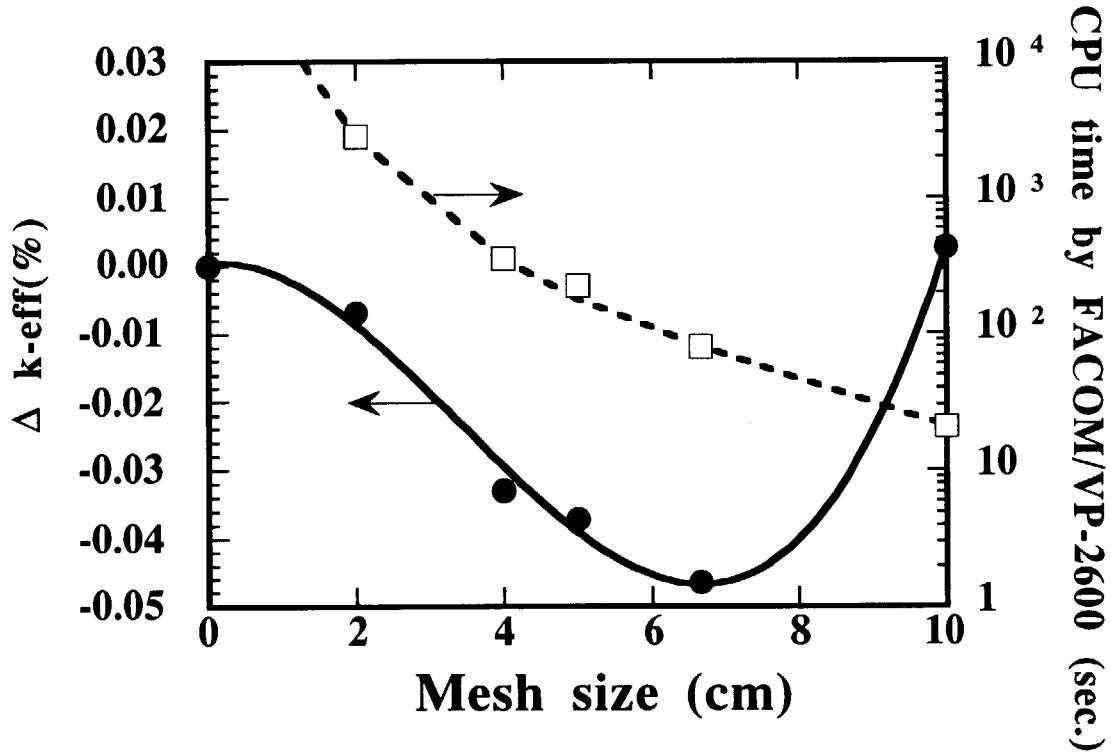


Fig.3.5.6 Mesh edffect on k-eff error and required computation time in IAEA 3D benchmark calculation by CITATION

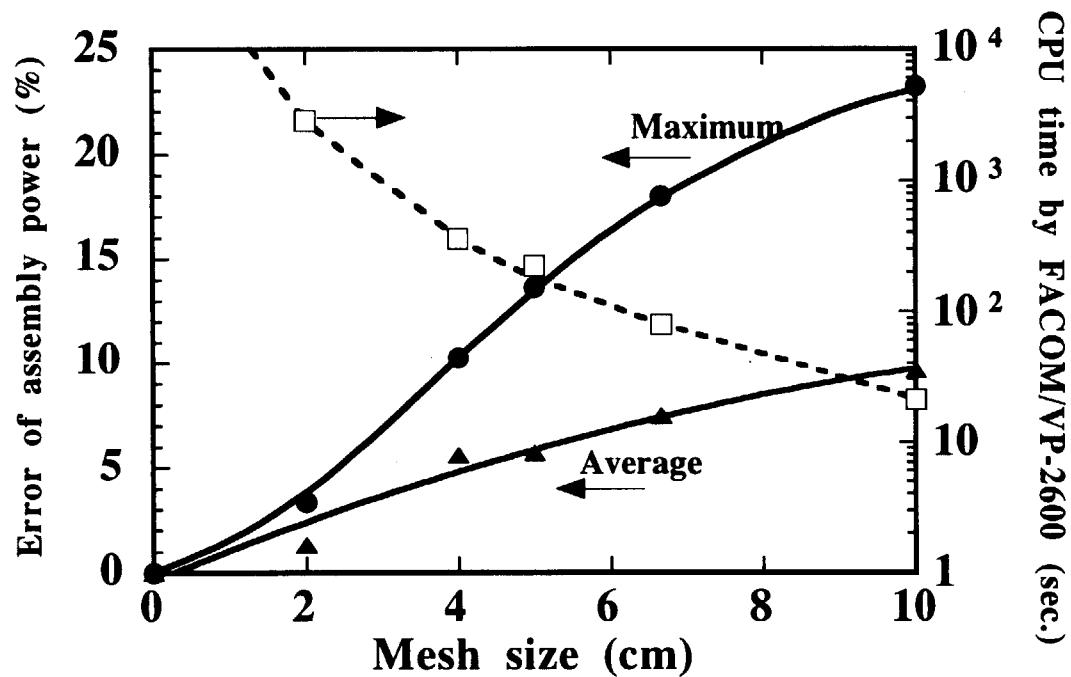


Fig.3.5.7 Mesh effect on assembly power error and required computation time in IAEA 3D benchmark calculation by CITATION

## 4. MOSRA-Lightコードの機能と詳細

### 4.1 コードの概要

本コードは、もともと、3次元拡散計算を繰り返し行うような計算コストが大きい問題、即ち核熱結合炉心燃焼計算や空間依存動特性計算などに利用する目的で開発を行った。このため、熱水力計算コード、燃焼計算コードなどの他コードとの結合を容易にすることを重視し、拡散方程式の解を与えるプログラムモジュールとして独立性の高いプログラミングを行ってある。従って、プログラムの内容を熟知しない者でも、比較的容易に他コードへの移植が可能である。MOSRA-Lightコードは、こうしたサブルーチン的な性格を持つ拡散計算モジュールに、本来の目的を損なわないように最低限度のプログラム変更を加え、単独コードとして利用できるように整備したものである。従って、MOSRA-Lightの機能は単純であり、それは、巨視的断面積と幾何形状などの基本的な入力データを与えれば、中性子束分布と出力分布を出力することである。しかし、ユーザーの希望は多岐であり、様々な定義に基づく反応率分布の計算、中性子漏洩の計算、動特性パラメータの計算、摂動計算などが要求される。こうした機能はモジュラーコードとしては歓迎されないため、全てMOSRA-Lightが出力するデータを利用して、ユーティリティコードで対応できるようにしている。以下に、MOSRA-Lightの機能と特長を示す。

- 1) 4次の多項式展開拡散ノード法に基づく高速ベクトル計算
- 2) 適用体系：X-Y-Z体系（1次元、2次元計算は反射境界により扱う）
- 3) 適用問題：固有値問題、固定源問題、随伴問題
- 4) エネルギー群数の制限無し（上方散乱考慮）
- 5) 可変長配列の採用により、メッシュ分割数等の制限無し
- 6) 詳細メッシュ計算機能（有限差分法と等価）
- 7) 3方向独立した方向依存拡散係数の使用可能
- 8) 6面独立した不連続因子の使用可能
- 10) 豊富な境界条件（真空、反射、アルベド、周期、90度回転、対角対称、内部黒体）
- 11) SRACコード<sup>25)</sup>が作成する巨視的断面積ファイルがそのまま使用可能
- 12) (n,2n)反応補正対応
- 13) ユーザーによる出力編集形式の変更
- 14) 収束パラメータの自動調整
- 15) 他コードへ移植が容易
- 16) 1点炉動特性パラメータの計算（ユーティリティ）
- 17) 1次摂動と厳密摂動計算（ユーティリティ）
- 18) 動作環境：UNIX系OSを使用するほとんどの計算機（大型機、EWS、PC/AT）
- 19) 会話型自動インストーラ、ヘルプ機能を装備

## 4.2 MOSRA-Lightコードの仕様

MOSRA-Lightが扱う所とする一般的な理論については、第2章に記述した。また、高速計算のアルゴリズムについては、第3章に記述した。ここでは、これまでに記述しなかったMOSRA-Light独自の扱いや計算手法の詳細について補足する。

### 4.2.1 固定源問題

第2章では、固有値問題に限定して話を進めたが、これを固定源問題に拡張する。ここでは、未臨界体系中に有限な大きさの外部源（例えばCf中性子源）がある場合を考える。核分裂源の有無は問わないが、外部源から発生する中性子に指向性は無く等方であると仮定する（拡散理論で記述できる外部源）。固定源問題の場合には、(2.1.6)式において便宜的に $k_{eff}=1$ とし、更に外部源の項を加えれば良い。即ち、第 $g$ 群の中性子源は、核分裂源、散乱源、外部源で構成され以下の式で表される。

$$Q_g^k(x,y,z) \equiv \chi_g^k \sum_{g'=1}^G v \sum_{g'}^k \phi_{g'}^k(x,y,z) + \sum_{\substack{g'=1 \\ g' \neq g}}^G \sum_{s,g' \rightarrow g}^k \phi_{g'}^k(x,y,z) + s_g^k(x,y,z) \quad (4.2.1)$$

ここで、 $s_g^k(x,y,z)$ は、 $k$ 番目ノード内における第 $g$ 群外部源の分布であり、 $k$ 番目ノードにおける外部源の強度を $S_{ex}^k(n/s)$ とすると、以下のように規格化される。

$$S_{ex}^k = \sum_{g=1}^G \int_{-\Delta k/2}^{\Delta k/2} \int_{-\Delta y/2}^{\Delta y/2} \int_{-\Delta z/2}^{\Delta z/2} s_g^k(x,y,z) dx dy dz \quad (4.2.2)$$

以降の式の操作については、固有値問題の場合と同様であり、(4.2.1)式を(2.1.3)式に代入し、トランスバース積分(第2.2節)、中性子束分布の多項式展開(第2.3節)、重み付き残差法(第2.4節)を適用すれば、各方向についてノードバランス式( $n=0$ )、1次のモーメントバランス式( $n=1$ )、及び2次のモーメントバランス式( $n=2$ )を得る。これらの式は、中性子源の項が(4.2.3)式に変わることを除けば、固有値問題の場合と同じである。

$$\begin{aligned} Q_{g,xn}^k &= \langle w_n(x), \bar{Q}_{g,x}^k(x) \rangle \\ &= \frac{1}{\Delta_x^k} \int_{-\Delta x/2}^{\Delta x/2} w_n(x) \bar{Q}_{g,x}^k(x) dx \\ &= \chi_g^k \sum_{g'=1}^G v \sum_{g'}^k \phi_{g',xn}^k + \sum_{\substack{g'=1 \\ g' \neq g}}^G \sum_{s,g' \rightarrow g}^k \phi_{g',xn}^k + s_{g,xn}^k \end{aligned} \quad (4.2.3)$$

ただし、

$$s_{g,xn}^k \equiv \frac{1}{\Delta_x^k} \int_{-\Delta_x^{k/2}}^{\Delta_x^{k/2}} w_n(x) \bar{s}_{g,x}^k(x) dx \quad (4.2.4)$$

$$\bar{s}_{g,x}^k(x) \equiv \frac{1}{\Delta_y^k \Delta_z^k} \int_{-\Delta_y^{k/2}}^{\Delta_y^{k/2}} \int_{-\Delta_z^{k/2}}^{\Delta_z^{k/2}} s_g^k(x,y,z) dy dz \quad (4.2.5)$$

上式は、x方向に対するものであるが、y方向、z方向に対しても同様である。

ノード内における外部源  $s_g^k(x,y,z)$  の分布が明確な関数で与えられる場合には、(4.2.4)式、及び(4.2.5)式の積分を行えるが、コードのユーザーが考える外部源の分布を一般的に表現することは困難である。このため、MOSRA-Lightでは、外部源の空間分布はノード内で一様であると仮定する。逆に、外部源の分布が無視できない場合には、その分布を考慮できるようにノード分割をする必要がある。

$$s_g^k(x,y,z) \approx s_{g,0}^k \text{ (constant)} \quad (4.2.6)$$

これを、(4.2.4)式、及び(4.2.5)式に代入し、 $w_n(x)$ に(2.3.4a)式～(2.3.4c)式で与えられる重み関数を使用すると、高次の外部源項はゼロとなる(2.3.6式参照)。

$$s_{g,x0}^k = s_{g,0}^k \int_{-1/2}^{1/2} d\xi = s_{g,0}^k \quad (4.2.7a)$$

$$s_{g,x1}^k = s_{g,0}^k \int_{-1/2}^{1/2} \xi d\xi = 0 \quad (4.2.7b)$$

$$s_{g,x2}^k = s_{g,0}^k \int_{-1/2}^{1/2} \left( 3\xi^2 - \frac{1}{4} \right) d\xi = 0 \quad (4.2.7c)$$

$s_{g,0}^k$  の規格化は、(4.2.2)式に(4.2.6)式を代入して、以下の式で与えられる。ただし、 $V^k$ はノード体積( $\Delta_x^k \Delta_y^k \Delta_z^k$ )である。

$$S_{ex}^k = V^k \sum_{g=1}^G s_{g,0}^k \quad (4.2.8)$$

結局、未臨界体系中にノード内で一様な外部源が存在する固定源問題を解くには、外部反復計算において、常時  $k_{eff}=1$  とするとともに、0次の源項に外部源のノード平均密度  $s_{g,0}^k$  ( $n/cm^3/s$ ) を加えるだけでよい。ちなみに、臨界または超臨界体系中に外部源が存在する場合には、体系中の中性子数は時間と共に増大し、定常解は存在しない。計算上も、外部反復

計算毎に体系中の中性子数が増大し、永遠に収束しない。 $k_{\text{eff}} \geq 1$ の可能性がある場合には、あらかじめ固有値計算を行って、中性子倍率を調べておく必要がある。

#### 4.2.2 随伴問題

1点炉動特性パラメータの計算や摂動計算などを行うには、随伴中性子束が必要になる。ここでは、固有値問題を例として、MOSRA-Lightが行う随伴中性子束分布の計算手法を説明する。先ず、話をより具体的なものとするため、以下のような簡単化された3群の拡散方程式を考える。これは、各群における中性子の漏洩、消滅(吸収+自群以外の散乱)、生成(核分裂+他群からの散乱)のバランス式である。

$$-D_1 \nabla^2 \phi_1 + \Sigma_{r,1} \phi_1 = \frac{\chi_1}{k_{\text{eff}}} (\nu \Sigma_{f,1} \phi_1 + \nu \Sigma_{f,2} \phi_2 + \nu \Sigma_{f,3} \phi_3) + (\Sigma_{2 \rightarrow 1} \phi_2 + \Sigma_{3 \rightarrow 1} \phi_3) \quad (4.2.9a)$$

$$-D_2 \nabla^2 \phi_2 + \Sigma_{r,2} \phi_2 = \frac{\chi_2}{k_{\text{eff}}} (\nu \Sigma_{f,1} \phi_1 + \nu \Sigma_{f,2} \phi_2 + \nu \Sigma_{f,3} \phi_3) + (\Sigma_{1 \rightarrow 2} \phi_1 + \Sigma_{3 \rightarrow 2} \phi_3) \quad (4.2.9b)$$

$$-D_3 \nabla^2 \phi_3 + \Sigma_{r,3} \phi_3 = \frac{\chi_3}{k_{\text{eff}}} (\nu \Sigma_{f,1} \phi_1 + \nu \Sigma_{f,2} \phi_2 + \nu \Sigma_{f,3} \phi_3) + (\Sigma_{1 \rightarrow 3} \phi_1 + \Sigma_{2 \rightarrow 3} \phi_2) \quad (4.2.9c)$$

これを行列式で表現すると、以下のようになる。

$$\begin{bmatrix} -D_1 \nabla^2 + \Sigma_{r,1} & -\Sigma_{2 \rightarrow 1} & -\Sigma_{3 \rightarrow 1} \\ -\Sigma_{1 \rightarrow 2} & -D_2 \nabla^2 + \Sigma_{r,2} & -\Sigma_{3 \rightarrow 2} \\ -\Sigma_{1 \rightarrow 3} & -\Sigma_{2 \rightarrow 3} & -D_3 \nabla^2 + \Sigma_{r,3} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} = \frac{1}{k_{\text{eff}}} \begin{bmatrix} \chi_1 \nu \Sigma_{f,1} & \chi_1 \nu \Sigma_{f,2} & \chi_1 \nu \Sigma_{f,3} \\ \chi_2 \nu \Sigma_{f,1} & \chi_2 \nu \Sigma_{f,2} & \chi_2 \nu \Sigma_{f,3} \\ \chi_3 \nu \Sigma_{f,1} & \chi_3 \nu \Sigma_{f,2} & \chi_3 \nu \Sigma_{f,3} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}$$

これを、しばしば、

$$[\mathbf{M}] \vec{\phi} = \frac{1}{k_{\text{eff}}} [\mathbf{F}] \vec{\phi} \quad (4.2.11)$$

と一般的に表わす。また、(4.2.11)式に対する随伴方程式は(4.2.12)式で与えられる。

$$[\mathbf{M}^t] \vec{\phi}^* = \frac{1}{k_{\text{eff}}} [\mathbf{F}^t] \vec{\phi}^* \quad (4.2.12)$$

$\vec{\phi}^*$ が目的とする随伴中性子束であり、 $[\mathbf{M}^t]$ と $[\mathbf{F}^t]$ は、それぞれ $[\mathbf{M}]$ と $[\mathbf{F}]$ の転置行列である。上記の3群拡散方程式を例にとれば、以下の式となる。

$$\begin{bmatrix} -D_1 \nabla^2 + \Sigma_{r,1} & -\Sigma_{1 \rightarrow 2} & -\Sigma_{1 \rightarrow 3} \\ -\Sigma_{2 \rightarrow 1} & -D_2 \nabla^2 + \Sigma_{r,2} & -\Sigma_{2 \rightarrow 3} \\ -\Sigma_{3 \rightarrow 1} & -\Sigma_{3 \rightarrow 2} & -D_3 \nabla^2 + \Sigma_{r,3} \end{bmatrix} \begin{bmatrix} \phi_1^* \\ \phi_2^* \\ \phi_3^* \end{bmatrix} = \frac{1}{k_{\text{eff}}} \begin{bmatrix} \chi_1 v \Sigma_{f,1} & \chi_2 v \Sigma_{f,1} & \chi_3 v \Sigma_{f,1} \\ \chi_1 v \Sigma_{f,2} & \chi_2 v \Sigma_{f,2} & \chi_3 v \Sigma_{f,2} \\ \chi_1 v \Sigma_{f,3} & \chi_2 v \Sigma_{f,3} & \chi_3 v \Sigma_{f,3} \end{bmatrix} \begin{bmatrix} \phi_1^* \\ \phi_2^* \\ \phi_3^* \end{bmatrix} \quad (4.2.13)$$

一般的な多次元多群の拡散方程式(forward)の場合には、3群の(4.2.10)式に対して、以下のように表される。

$$\nabla \cdot \mathbf{J}_g(\mathbf{r}) + \Sigma_{r,g}(\mathbf{r}) \phi_g(\mathbf{r}) = Q_g(\mathbf{r}) \quad (4.2.14)$$

$$\mathbf{J}_g(\mathbf{r}) = -D_g(\mathbf{r}) \nabla \phi_g(\mathbf{r}) \quad (4.2.15)$$

$$Q_g(\mathbf{r}) = \frac{\chi_g(\mathbf{r})}{k_{\text{eff}}} \sum_{g'=1}^G v \Sigma_{f,g'}(\mathbf{r}) \phi_{g'}(\mathbf{r}) + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{s,g' \rightarrow g}(\mathbf{r}) \phi_{g'}(\mathbf{r}) \quad (4.2.16)$$

一方、これに対する随伴方程式は、(4.2.13)式と対比して、以下の式で表される。

$$\nabla \cdot \mathbf{J}_g^*(\mathbf{r}) + \Sigma_{r,g}(\mathbf{r}) \phi_g^*(\mathbf{r}) = Q_g^*(\mathbf{r}) \quad (4.2.17)$$

$$\mathbf{J}_g^*(\mathbf{r}) = -D_g(\mathbf{r}) \nabla \phi_g^*(\mathbf{r}) \quad (4.2.18)$$

$$Q_g^*(\mathbf{r}) = \frac{v \Sigma_{f,g}(\mathbf{r})}{k_{\text{eff}}} \sum_{g'=1}^G \chi_{g'}(\mathbf{r}) \phi_{g'}^*(\mathbf{r}) + \sum_{\substack{g'=1 \\ g' \neq g}}^G \Sigma_{s,g \rightarrow g'}(\mathbf{r}) \phi_{g'}^*(\mathbf{r}) \quad (4.2.19)$$

(4.2.14)式と(4.2.17)式を比較すると、随伴方程式を解いて  $\phi_g^*(\mathbf{r})$  を得るには、forwardの式において、単に  $Q_g(\mathbf{r})$  を  $Q_g^*(\mathbf{r})$  に置き換えるだけで良いことが分かる。これは、群定数セットにおいて、 $\chi_g$  と  $v \Sigma_{f,g}$  を便宜的に入れ替え、更に散乱マトリックスを転置することに相当する。

$$\begin{cases} \chi_g(\mathbf{r}) \leftrightarrow v \Sigma_{f,g}(\mathbf{r}) \\ \Sigma_{s,g' \rightarrow g}(\mathbf{r}) \leftrightarrow \Sigma_{s,g \rightarrow g'}(\mathbf{r}) \end{cases} \quad (4.2.20)$$

一般に、高速群においては、上方散乱は含まれない。そこで、(4.2.10)式において、上方散乱を無視すると以下の式となる。

$$\begin{bmatrix} -D_1 \nabla^2 + \Sigma_{r,1} & 0 & 0 \\ -\Sigma_{1 \rightarrow 2} & -D_2 \nabla^2 + \Sigma_{r,2} & 0 \\ -\Sigma_{1 \rightarrow 3} & -\Sigma_{2 \rightarrow 3} & -D_3 \nabla^2 + \Sigma_{r,3} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix} = \frac{1}{k_{eff}} \begin{bmatrix} \chi_1 v \Sigma_{f,1} & \chi_1 v \Sigma_{f,2} & \chi_1 v \Sigma_{f,3} \\ \chi_2 v \Sigma_{f,1} & \chi_2 v \Sigma_{f,2} & \chi_2 v \Sigma_{f,3} \\ \chi_3 v \Sigma_{f,1} & \chi_3 v \Sigma_{f,2} & \chi_3 v \Sigma_{f,3} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}$$

(4.2.21)

ここで、第1群(1行目)の式には、散乱源が現れない。また、第2群(2行目)の式には、第1群からの散乱源が現れるが、第3群からの散乱源は現れない。こうした特性は、収束計算上有利となる。すなわち、外部反復計算においては、第1群から下群へと順次に拡散方程式が解かれるが、上の群で計算された中性子束の結果を即座に次の群の減速源として反映できるためである。一方、(4.2.13)式の随伴問題において、同様に上方散乱を無視すると、以下の式となる。

$$\begin{bmatrix} -D_1 \nabla^2 + \Sigma_{r,1} & -\Sigma_{1 \rightarrow 2} & -\Sigma_{1 \rightarrow 3} \\ 0 & -D_2 \nabla^2 + \Sigma_{r,2} & -\Sigma_{2 \rightarrow 3} \\ 0 & 0 & -D_3 \nabla^2 + \Sigma_{r,3} \end{bmatrix} \begin{bmatrix} \phi_1^* \\ \phi_2^* \\ \phi_3^* \end{bmatrix} = \frac{1}{k_{eff}} \begin{bmatrix} \chi_1 v \Sigma_{f,1} & \chi_2 v \Sigma_{f,1} & \chi_3 v \Sigma_{f,1} \\ \chi_1 v \Sigma_{f,2} & \chi_2 v \Sigma_{f,2} & \chi_3 v \Sigma_{f,2} \\ \chi_1 v \Sigma_{f,3} & \chi_2 v \Sigma_{f,3} & \chi_3 v \Sigma_{f,3} \end{bmatrix} \begin{bmatrix} \phi_1^* \\ \phi_2^* \\ \phi_3^* \end{bmatrix}$$

(4.2.22)

この場合に、第1群から下群へと計算を進めると、形式的に全群にわたって上方散乱を含む問題と同じになり、外部反復計算の収束が遅くなる。そこで、(4.2.22)式において、エネルギー群について解く順序を逆にする。

$$\begin{bmatrix} -D_3 \nabla^2 + \Sigma_{r,3} & 0 & 0 \\ -\Sigma_{2 \rightarrow 3} & -D_2 \nabla^2 + \Sigma_{r,2} & 0 \\ -\Sigma_{1 \rightarrow 3} & -\Sigma_{1 \rightarrow 2} & -D_1 \nabla^2 + \Sigma_{r,1} \end{bmatrix} \begin{bmatrix} \phi_3^* \\ \phi_2^* \\ \phi_1^* \end{bmatrix} = \frac{1}{k_{eff}} \begin{bmatrix} \chi_3 v \Sigma_{f,3} & \chi_2 v \Sigma_{f,3} & \chi_1 v \Sigma_{f,3} \\ \chi_3 v \Sigma_{f,2} & \chi_2 v \Sigma_{f,2} & \chi_1 v \Sigma_{f,2} \\ \chi_3 v \Sigma_{f,1} & \chi_2 v \Sigma_{f,1} & \chi_1 v \Sigma_{f,1} \end{bmatrix} \begin{bmatrix} \phi_3^* \\ \phi_2^* \\ \phi_1^* \end{bmatrix}$$

(4.2.23)

結局、散乱マトリックスを転置してからエネルギー群を逆順にするには、プログラム上で以下のような配列の入れ替え操作を行うことになる。

$$\Sigma_s(i,j) \rightarrow \Sigma_s(G-j+1, G-i+1) \quad (4.2.24)$$

(4.2.22)式と(4.2.23)式は同じ方程式であるが、第3群から逆順に解くことにより、外部反復計算の収束性が良くなる。ちなみに、MOSRA-Lightにより全107群(高速群61+熱群46)の1領域無限体系の問題を解いたところ、(4.2.1)式のように第1群から求めた場合と、(4.2.23)式の

ようにエネルギー群を逆にして解いた場合とでは、外部反復回数に約1.3倍の差が見られた。上方散乱を含む群が少ない場合には、更に大きな効果が予想される。

以上をまとめると、随伴問題においては、MOSRA-Lightは以下のようない操作を行う。

- 1) 散乱マトリックスを転置する前に $\Sigma_{r,g}$ を作成する。
- 2)  $\chi_g$ と $v\Sigma_{f,g}$ の入れ替え、及び散乱マトリックスの転置(4.2.20式)
- 3) 断面積配列のエネルギー群についての逆転
- 4) forwardの場合と同じプロセスで中性子束分布を計算する。

このような方法は、特別なものではなく、旧来から有限差分法コードで広く利用されている。しかし、ノード法の場合には少し事情が異なる。そもそも、随伴中性子束には2種類の定義がある<sup>9)</sup>。一つは、物理的随伴中性子束(Physical Adjoint Flux)と呼ばれるもので、空間的に連続な変数を扱うforward方程式に対して、先ず随伴方程式を導出し、これを離散化(空間メッシュ分割)して得られる方程式の解である。もう一つは、forward方程式を離散化して得られる行列式に対して、その係数行列を転置して得られる随伴行列式の解である。これを数学的随伴中性子束(Mathematical Adjoint Flux)と呼ぶ。先述の断面積の入れ替え操作によりforward方程式と同じ計算プロセスで得られる解は、物理的随伴中性子束である。

有限差分法の場合には、差分行列の対称性により、物理的随伴中性子束と数学的随伴中性子束が原理的に一致することが証明される。しかし、多項式展開ノード法では、重み付き残差法(第2.4節)やQTL近似(第2.6節)を導入し、これにより行列式が非対称となるため、物理的随伴中性子束と数学的随伴中性子束が一致しなくなる。

摂動計算などに随伴中性子束を利用するには、より高い精度が期待できる数学的随伴中性子束が推奨される。多項式展開ノード法で数学的随伴中性子束を直接求めるには、forward方程式を解く場合とは別の複雑なプロセスが必要になる。そこで、物理的随伴中性子束を算出してから、これを数学的随伴中性子束に変換する方法<sup>26)</sup>や、より効率的に数学的随伴中性子束を計算する方法<sup>27)</sup>が提案されている。

物理的随伴中性子束が数学的随伴中性子束と全く異なるわけではなく、両者の差異はむしろ精度的なものである。MOSRA-Light(4次のNEM)が計算出力する随伴中性子束は物理的随伴中性子束である。筆者がこれまでに行った計算の範囲では、MOSRA-Lightにより得られた物理的随伴中性子束は、詳細メッシュ有限差分法による数学的随伴中性子束と良く一致している。また、Kim等により4次の多項式展開ノード法により得られた物理的随伴中性子束と数学的随伴中性子束が比較されているが、両者の間に有為な差異は見られない<sup>27)</sup>。

#### 4.2.3 熱群反復計算

第2.9節と第3.4節での計算フローの説明では触れなかったが、MOSRA-Lightは上方散乱を含む多群問題一般に対応するため、外部反復計算の中で、いわゆる熱群反復計算を行う。

例えば、全4群の拡散方程式の例で、第1群と第2群が上方散乱源を含まない高速群であり、第3群と第4群は第4群から第3群への上方散乱を含む熱群であるものとする。この場合の拡散方程式は、以下のように表される。

$$-D_1 \nabla^2 \phi_1 + \Sigma_{r,1} \phi_1 = F_1 \quad (4.2.25a)$$

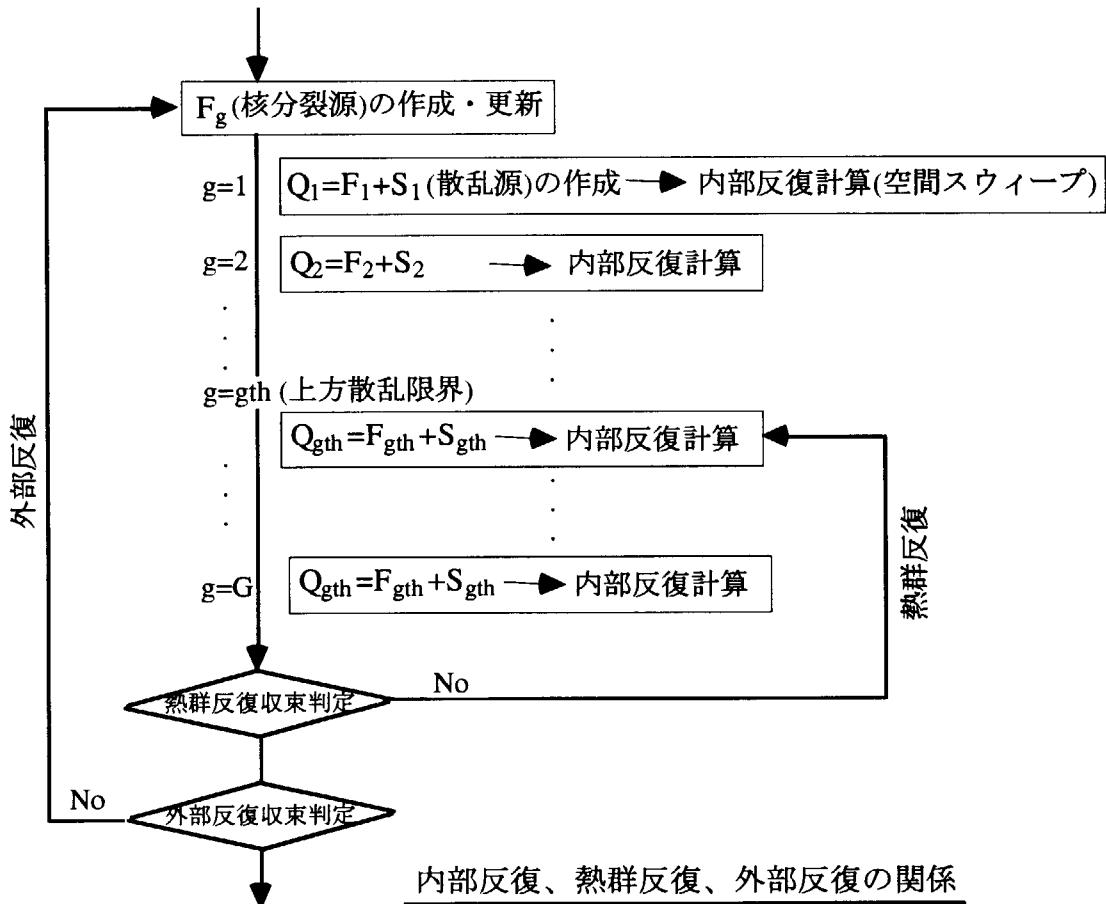
$$-D_2 \nabla^2 \phi_2 + \Sigma_{r,2} \phi_2 = F_2 + \Sigma_{1 \rightarrow 2} \phi_1 \quad (4.2.25b)$$

$$-D_3 \nabla^2 \phi_3 + \Sigma_{r,3} \phi_3 = F_3 + (\Sigma_{1 \rightarrow 3} \phi_1 + \Sigma_{2 \rightarrow 3} \phi_2) + \Sigma_{4 \rightarrow 3} \phi_4 \quad (4.2.25c)$$

$$-D_4 \nabla^2 \phi_4 + \Sigma_{r,4} \phi_4 = F_4 + (\Sigma_{1 \rightarrow 4} \phi_1 + \Sigma_{2 \rightarrow 4} \phi_2 + \Sigma_{3 \rightarrow 4} \phi_3) \quad (4.2.25d)$$

ただし、

$$F_g = \frac{\chi_g}{k_{\text{eff}}} (v \Sigma_{f,1} \phi_1 + v \Sigma_{f,2} \phi_2 + v \Sigma_{f,3} \phi_3 + v \Sigma_{f,4} \phi_4) \quad (4.2.26)$$



核分裂源  $F_g$  は、外部反復毎に更新されるもので、外部反復の中における計算では一定として扱われる。 $F_g$  が一定であるため、 $\phi_1$ のみに依存する(4.2.25a)式は即座に解くことができる。得られた  $\phi_1$  を(4.2.25b)式右辺第2項の減速源に代入し、 $\phi_2$  が得られる。上方散乱源を含む第3

群方程式(4.2.25c)では、 $\phi_4$ が未だ得られていないため、推定値または前回の外部反復で得られた値を使用することになる。こうして得られた $\phi_3$ は、当然精度的に劣ったものとなるが、それは、 $\phi_3$ を減速源として(4.2.25d)に代入して解かれる $\phi_4$ にも影響する。熱中性子炉の場合には、(4.2.26)式で与えられる核分裂源は、ほとんど熱群中性子束( $\phi_3$ と $\phi_4$ )により支配的に決められる。このため、次の外部反復計算に使用する $F_g$ の精度は、 $\phi_3$ と $\phi_4$ の精度に支配され、 $F_g$ を介して次の反復計算で全群に影響を及ぼすことになる。結果として、収束解を得るまでに多くの外部反復を必要とし効率的ではない。そこで、各外部反復計算の中で、上方散乱の影響を受ける群の範囲で反復計算を行う。上記の例では、(4.2.25c)式と(4.2.25d)式を繰り返し解くことになる。これを熱群反復計算(thermal iteration)と呼ぶ。

上方散乱を含む場合に熱群反復計算を行わなくても、外部反復計算により散乱源は更新されるため、収束解は得られることにはかわりない。下記の表は、NEA/NSCの2次元ベンチマーク問題(第6.2.3節参照)を、107群(高速群61+熱群46)でMOSRA-Lightにより計算した場合の、熱群反復計算の有無の効果を示したものである。

計算ケース	外部反復回数(無し/有り)	計算速度比(無し/有り)
H-MOX(no void)	35回 / 19回 =1.8倍	1.2倍
H-MOX(voided)	40回 / 23回 =1.7倍	1.4倍

使用計算機はVP-2600

熱群反復計算により内部反復計算の回数は増加するため、外部反復計算回数の減少がそのまま計算速度の向上に反映されるわけではない。熱群反復計算の導入による計算速度の向上は、エネルギー群構造や断面積などの問題の性質に依存する。また、計算コストの分布(第3.5.2節参照)は、計算機の性能や種類にも依存し、熱群反復計算の効果は使用する計算機にも依存することになる。

MOSRA-Lightでは、散乱マトリックスに有為な上方散乱( $\Sigma > 1.0E-6$ )があることを検知すると、上方散乱の上限エネルギー群を調べ、そのエネルギー群から下限エネルギー群の範囲で熱群反復計算をするようにしている。熱群反復計算は、以下の条件の何れかが満足された場合に終了する。

- (1) 热群反復計算の回数が、ユーザーまたはコードのデフォルト値が定める上限回数に達した場合。
- (2) 次の収束判定を全ての熱群反復計算群について満足した場合。

$$\max_{g, k} \frac{\left| \phi_{g,0}^k - \phi_{g,0}^{k, \text{old}} \right|}{\phi_{g,0}^k} \leq \varepsilon_{\text{inner}} \quad (4.2.27)$$

但し、 $\phi_{g,0}^{k, \text{old}}$ は一つ前の熱群反復計算におけるノード平均中性子束であり、 $\varepsilon_{\text{inner}}$ は内部反復計算に対する収束判定因子(第4.2.7節参照)である。

#### 4.2.4 詳細メッシュ計算モード

MOSRA-Lightは、4次の多項式展開ノード法(NEM)を標準的な解法とするが、それ以外に2次の多項式展開ノード法が利用できる。ここで、4次及び2次のNEMと有限差分法の各手法に導入されている近似を比較してみる。詳細については第2章を参照されたい。

##### [ 4次のNEMにおける近似 ]

- 1) トランスバース積分された1次元中性子束のノード内分布を4次の多項式で表現する。  
4次式の範囲でしか表現することができないノード内分布を、厳密な拡散方程式をなるべく満足するように近似する。即ち、4次式の5つの係数のうち、3つは物理的な根拠(ノード体積平均値の保存とカレントについての境界条件)をもって決定されるが、残り2つの係数を、重み付き残差法により決めている。
- 2) ノード内のトランスバース漏洩の分布を、隣接ノードを含めた3つの平均値を使用して2次式で表現する(QTLA:第2.6.3節参照)。

##### [ 2次のNEMにおける近似 ]

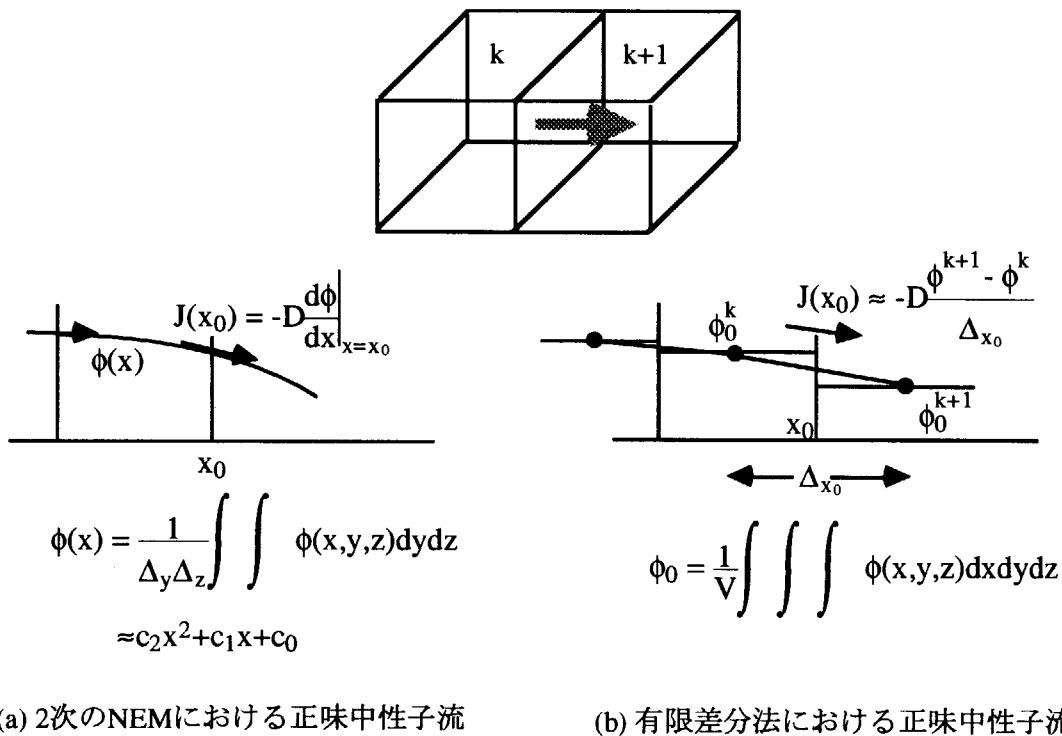
- 1) トランスバース積分された1次元中性子束のノード内分布を2次の多項式で表現する。  
分布の表現能力が4次の場合に比べて低いため、十分な精度を得るには詳細メッシュを使用する必要がある。

##### [ 有限差分法における近似 ]

- 1) 中性子束のノード(メッシュ)内分布を一様とする。ノード内分布を考慮しないため、十分な精度を得るには詳細メッシュを使用する必要がある。

ちなみに、トランスバース積分による方程式の1次元化は、3次元中性子束の変数分離などを仮定しているわけではなく、近似ではない。上記の比較からも明らかのように、計算時間やメモリー容量などの計算資源に全く制約がないものとすれば、原理的に2次のNEMや有限差分法は、4次のNEMよりも正確な解法と言える。

2次のNEMと有限差分法は、全体系の中性子バランス式を解くことに関しては精度的な優劣の差違はない。中性子束を未知数として方程式を解くか、部分中性子流を未知数として解くかの違いはあるが、共に、未知数と差分方程式の数は過不足無く一致し、重み付き残差法やQTLAのような新たな近似を導入する必要は無い。2次のNEMと有限差分法との差異は、ノード間の正味中性子流をFickの法則に基づき記述する際の表現の違いにより生じる。2次のNEMでは、表面積分されたノード内中性子束分布を2次式で近似し、その係数をノード平均中性子束と両側ノードとのカレントの連続により決める。一方、有限差分法では、ノード平均中性子束によりノード中心中性子束を代表させ、その隣接ノードとの勾配により中性子流を決める。下図は、x方向中性子流について、これを模式的に示したものである。



中性子束の分布が隣接ノード間で比較的滑らかであるなら、表面積分により面平均値として漏洩を扱う2次のNEMの方が、有限差分法に比べて精度的に有利と考えられる。一方、中性子束の分布が2次多項式の表現範囲を超えるような場合には、有限差分法の方が安定感がある。いずれの場合もメッシュ幅が十分小さければ問題とはならない。両手法の優劣は、問題の特性にも因るが、多くの場合、2次のNEMの方がメッシュ効果は小さいようである。

IAEAの3次元ベンチマーク問題(Fig.3.5.1参照)を軸方向1次元問題に変えて、4次のNEM(MOSRA-Light)、2次のNEM(MOSRA-Light)、及び有限差分法(CITATION)により、中性子増倍率のメッシュ効果を調べた。体系は、制御棒が挿入されていない内側燃料(Fuel Type-2)を炉心(340cm)とし、両側に20cmの反射体を付けた1次元平板体系(バックリング無し)である。なお、断面積には、ベンチマークで与えられている数値(Table 3.5.1参照)を使用している。結果をFig.4.2.1に示す。均質炉心であるためメッシュ効果があまり無い問題ではあるが、2次のNEMは有限差分法と同等もしくはそれ以上の精度があることが分かる。

MOSRA-Lightが2次のNEMの機能を持つ理由は、以下の2点である。

- 1) 有限差分法コード(VENTURE、CITATION)に代わる参照解の提供
- 2) 物質境界により粗メッシュが適用できない体系での高速計算

4次のNEMに代表される近代ノード法は、商用炉のシミュレータとして発展した経緯もあり、そうした炉心を高速高精度で解析することに関しては極めて優れている。一方、研究炉や臨界集合体では、炉心内部や周辺に照射孔や様々な測定機器などの構造があり、これらを

均質化することなく物質境界として扱う必要がある。また、中性子ビームを取り出す場合の解析では、反射体領域の中性子束の精度が重要になる。4次のNEMのQTLAは炉心部には良い近似であっても、真空境界周辺や熱中性子束が盛り上がる反射体領域では、その保証は無い。こうした場合には、計算コストが高くなつても、有限差分法や2次のNEMを利用する方が確実である。また、ベンチマーク問題などの参照解を得る場合に、全く異なる仕様の別コードを利用しなくとも良いという利点がある。こうした理由により、2次のNEMを使用して詳細メッシュ計算をする場合、MOSRA-Lightは4次NEM以上にベクトル化による高速性能をフルに発揮することができる。理由は3つ有り、第1に、外部境界に隣接する例外処理ノードの割合が小さくなり、境界分離チェックボードスウェープがより有効に働くためである(第3.4.2節参照)。第2に、ノード総数が多くなることによりベクトル長が長くなるためである。第3に、QTLAが入らないため、4次のNEMで律速となっているトランスペーパス漏洩項の更新に計算時間をとられないためである(Fig.3.5.4参照)。

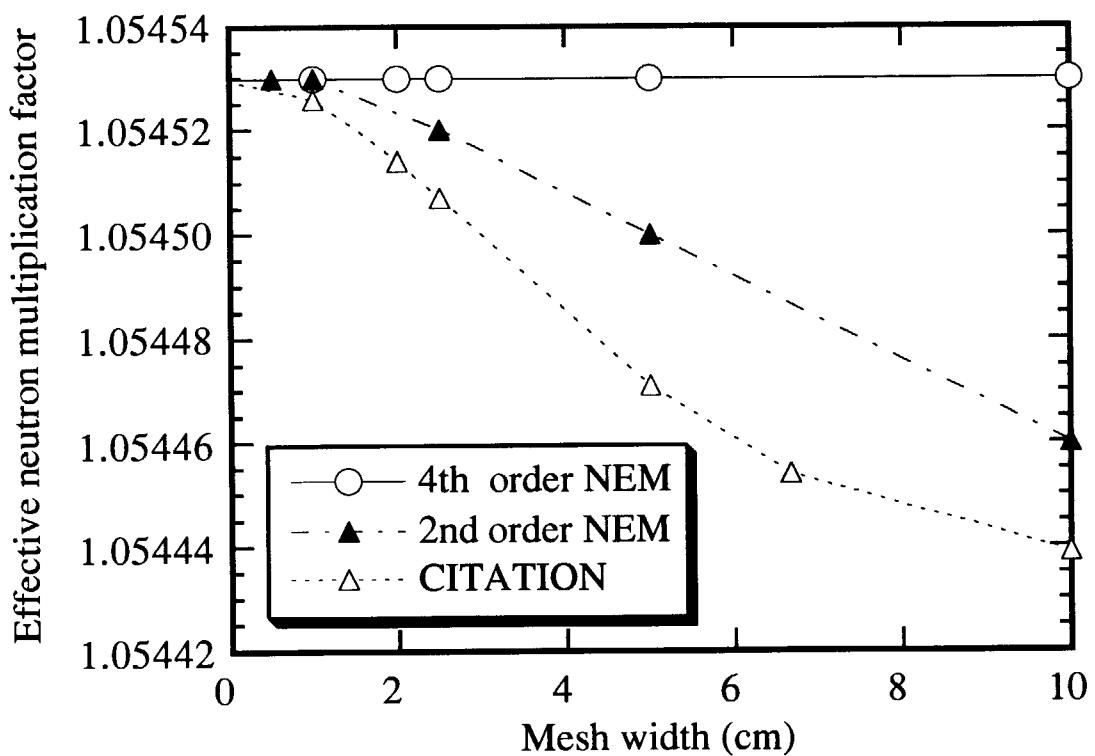


Fig.4.2.1 Comparison of mesh effect on k-eff among the 4th order NEM(MOSRA-Light), the 2nd order NEM (MOSRA-Light) and the finite difference method (CITATION) in the axial 1D calculation for the inner fuel in IAEA benchmark problem.

#### 4.2.5 加速法

MOSRA-Lightは、今のところ加速法にはあまり重点を置いていない。現在利用できる加速法は、1) 中性子源外挿法(source extrapolation method)と、2) 2次のNEMを利用する方法である。以下にその概要を述べる。

##### (1) 中性子源外挿法

大雑把に言えば、粗いメッシュを利用する場合ほど、内部反復回数を多くする必要性はなく、内部反復を1回や2回にしても外部反復の収束解が得られることが多い。また、MOSRA-Lightでは、内部反復部分はベクトル化により高速化されており、不用意に加速法を導入して独立した演算の妨げとは得策ではない。このようなコードの特性から考えて、外部反復に対する加速法を導入することが合理的である。外部反復の加速法には、中性子源外挿法、チエビシュフ加速法、ウィーラント法などがあり<sup>28)</sup>、MOSRA-Lightは中性子源外挿法<sup>29)</sup>を採用している。その原理や一般論については省略し、ここでは同加速法のMOSRA-Lightにおける具体的な扱いのみを述べる。

中性子源外挿法では、外部反復毎に規定の条件を満足した場合に、前回の外部反復(old)と現在の外部反復(now)で得られた核分裂源を使用して、以下に示す外挿により次の外部反復(next)に使用する核分裂源を作成する。

$$F_g^{k\text{next}} = F_g^{k\text{now}} + \tau \left\{ F_g^{k\text{now}} - F_g^{k\text{old}} \right\} \quad (4.2.28)$$

$$\text{ただし、 } F_g^k = \frac{\chi_g^k}{k_{\text{eff}}} \sum_{g'=1}^G v \Sigma_{f,g'}^k \phi_{g',0}^k \quad (4.2.29)$$

$\tau$ は外挿パラメータであり、以下のように決定される<sup>30)</sup>。

$$\tau = \frac{\mu_{\min}}{1-\mu_{\min}} \quad (4.2.30)$$

$$\begin{cases} \mu_{\min} \equiv \min(\sigma_{\max}, \sigma_{\text{mean}}) \\ \mu_{\max} \equiv \max(\sigma_{\max}, \sigma_{\text{mean}}) \end{cases} \quad (4.2.31)$$

$$\sigma_{\max} \equiv \frac{\varepsilon_{\max}^{\text{now}}}{\varepsilon_{\max}^{\text{old}}} (1 + \varepsilon_{\max}^{\text{old}}) \quad (4.2.32)$$

$$\sigma_{\text{mean}} \equiv \frac{\varepsilon_{\text{mean}}^{\text{now}}}{\varepsilon_{\text{mean}}^{\text{old}}} (1 + \varepsilon_{\text{mean}}^{\text{old}}) \quad (4.2.33)$$

$$\begin{cases} \varepsilon_{\max} \equiv \max |\varepsilon^k| \\ \varepsilon_{\text{mean}} \equiv \text{mean} |\varepsilon^k| \\ \varepsilon_{\min} \equiv \min |\varepsilon^k| \end{cases} \quad (4.2.34)$$

$$\varepsilon^k \equiv \frac{F^{k\text{now}} - F^{k\text{old}}}{F^{k\text{now}}} \quad (4.2.35)$$

$$F^k \equiv \sum_{g=1}^G F_g^k \quad (4.2.36)$$

なお、MOSRA-Lightでは、各外部反復において中性子源外挿を適用するのは、以下の条件が全て満たされた場合としている<sup>30)</sup>。

$$1) \text{外部反復回数} \geq 4 \quad (\text{解が漸近的になるまでの猶予}) \quad (4.2.37a)$$

$$2) \left| \frac{\varepsilon_{\max}^{\text{now}} - \varepsilon_{\min}^{\text{now}}}{k_{\text{eff}}^{\text{now}}} \right| \leq \left| \frac{\varepsilon_{\max}^{\text{old}} - \varepsilon_{\min}^{\text{old}}}{k_{\text{eff}}^{\text{old}}} \right| \quad (\text{収束方向である根拠}) \quad (4.2.37b)$$

$$3) \mu_{\max} < 1.0 \quad (2\text{次の固有値} < 1\text{次の固有値} \text{であるための本来的条件}) \quad (4.2.37c)$$

$$4) \mu_{\min} > 0.5 \quad (\text{適用効果が小さい場合の外挿を避ける}) \quad (4.2.37d)$$

$$5) \left| \frac{\tau^{\text{now}} - \tau^{\text{old}}}{\tau^{\text{old}}} \right| \leq 0.1 \text{を2回連続満足} \quad (\text{外挿パラメータの継続的安定性}) \quad (4.2.37e)$$

## (2) 2次のNEMを利用する方法

ここでの加速法は、4次のNEMを使用する場合に限られる。NEMに限らず、反復計算をより早く収束させるためには、良い初期値を与えることが重要である。4次のNEMによる計算に対し、2次のNEMで良い初期値を与えることができる。

4次のNEMと2次のNEMを比較すると、ノード総数が同じであれば、2次のNEMの方が精度は悪いが計算時間は短くなる。これは、2次のNEMは1つの中性子バランス式を解くだけであるのに対し、4次のNEMは、これに加えて、1次及び2次の高次モーメントバランス式を解く必要があるためである。このことは、2次のNEMの応答行列式(2.5.10式～2.5.16式)が、4次のNEMの応答行列式(2.5.27式～2.5.30式)に比べて、遙かに単純であることに見ることができる。また、MOSRA-Lightのベクトル計算では、QTLAの例外処理に計算時間を掛けている(Fig.3.5.4参照)、2次のNEMではこの部分の計算が不要である。この方法の計算手順は以下の通りである。

- 1) 初期値を与える。
- 2) 2次のNEMで外部反復を繰り返す。
- 3) 2次のNEMが収束するか外部反復が打ち切り限界に至ったら、4次のNEMに移行する。

4次のNEMに移行する際、高次モーメント中性子束の値が存在しないため、これを作成する必要がある。4次のNEMでは、トランスバース積分中性子束のx方向分布は、次の式で表される(第2.3節参照)。

$$\bar{\phi}_{g,x}^k(x) = \phi_{g,0}^k + a_{g,x,1}^k \xi + a_{g,x,2}^k \left(3\xi^2 - \frac{1}{4}\right) + a_{g,x,3}^k \left\{ \xi \left(\xi - \frac{1}{2}\right) \left(\xi + \frac{1}{2}\right) \right\} + a_{g,x,4}^k \left\{ \left(\xi^2 - \frac{1}{20}\right) \left(\xi - \frac{1}{2}\right) \left(\xi + \frac{1}{2}\right) \right\} \quad (4.2.38)$$

1次及び2次の係数は、2次のNEMと4次のNEM共に、ノード境界の表面平均中性子束により決定される(2.3.15式～2.3.16式)。一方、3次と4次の係数は、重み付き残差法により決定され、高次モーメント中性子束と以下のような関係がある(第2.4節参照)。

$$a_{g,x,3}^k = -120\phi_{g,x1}^k + 10a_{g,x,1}^k \quad (4.2.39)$$

$$a_{g,x,4}^k = -700\phi_{g,x2}^k + 35a_{g,x,2}^k \quad (4.2.40)$$

2次のNEMでは、トランスバース積分中性子束は2次式で表されるから、上式に  $a_{g,x,3}^k=0$ 、  $a_{g,x,4}^k=0$  を代入する。

$$\phi_{g,x1}^k = \frac{1}{12}a_{g,x,1}^k \quad (4.2.41)$$

$$\phi_{g,x2}^k = \frac{1}{20}a_{g,x,2}^k \quad (4.2.42)$$

$a_{g,x,1}^k$  と  $a_{g,x,2}^k$  は、部分中性子流とノード平均中性子束によって表され(2.4.2式)、それらを(4.2.41)式と(4.2.42)式に代入すれば以下の式を得る。

$$\phi_{g,x1}^k = \frac{1}{6}(J_{g,x+}^{out,k} + J_{g,x+}^{in,k} - J_{g,x-}^{out,k} - J_{g,x-}^{in,k}) \quad (4.2.43)$$

$$\phi_{g,x2}^k = \frac{1}{10}(J_{g,x+}^{out,k} + J_{g,x+}^{in,k} + J_{g,x-}^{out,k} + J_{g,x-}^{in,k} - \phi_{g,0}^k) \quad (4.2.44)$$

y方向及びz方向についても同様である。これらの式に2次のNEMで得られる部分中性子流とノード平均中性子束を代入し、高次モーメント中性子束の初期値を与えて4次のNEMに引き継ぐ。

IAEA3次元ベンチマーク問題(Fig.3.5.1参照)において、この加速法の効果を調べた。最初

の2次のNEMによる外部反復計算の回数を変え、最終的に4次のNEMで収束するまでの計算時間をベクトル計算機(FACOM/VPP-500)で測定した。メッシュ幅を20cm(内部反復回数最大2回)とする場合と、メッシュ幅を5cm(内部反復回数最大6回及び20回)とする場合について得られた結果をTable 4.2.1に示す。このテーブルでは、2次のNEMによる加速を行わない場合の計算時間を1に規格化して、スピードアップ率(規格化時間の逆数)で表してある。

Table 4.2.1において、2次のNEMによる計算回数に対してスピードアップ率が揺らいでいるのは、外部反復計算の途中で中性子源外挿が入るタイミングによるものである。概して言えば、メッシュ幅が粗い場合には、2次のNEMを使用するとかえって収束速度を遅くする傾向がある。これは、粗メッシュにおける2次のNEMは4次のNEMに対してあまり良い初期値を与えないためである。一般にメッシュ幅が小さくなるほど、収束に必要な内部反復回数が多くなる。Table 4.2.1に見られるように、2次のNEMによる加速はメッシュ幅が小さく、かつ内部反復回数を多く設定する場合に有効である。実際の計算では、予め最適な内部反復回数を予測することは困難であり、初めての問題に対しては収束性を重視して内部反復回数を多めに設定することが多い。筆者の経験では、内部反復回数を約20回以上とする場合や、中性子の平均自由行程を考慮して2次のNEMが良い近似となる問題に、この加速法を適用すると計算時間を1~3割程度節約することができる。最適な2次のNEMによる外部反復回数は、問題の性質や収束制御の方法(第4.2.7節参照)にも依存するが、平均的には20回程度が好ましい。MOSRA-Lightでは、この加速法の適用や2次のNEMの打ち切り反復回数の上限をユーザーが自由に設定できるようにしている。

Table 4.2.1 Speedup factor by acceleration of the 4th order NEM using the 2nd order NEM in IAEA 3D benchmark calculation on vector computer (FACOM/VPP-500,1-CPU)

Number of initial outer iterations by the 2nd order NEM	mesh=20cm (max.inner=2)	mesh=5cm (max.inner=6)	mesh=5cm (max.inner=20)
0	1.0 (1.44sec)	1.0 (92.45sec)	1.0 (154.20sec)
5	0.89	0.93	0.86
10	0.89	0.94	0.88
15	0.92	0.94	0.91
20	1.01	0.99	0.96
25	0.96	1.00	1.27
30	1.00	1.04	1.31

#### 4.2.6 初期値とリスタート計算

Fig.3.4.1に示したように、4次の多項式展開ノード法で反復計算を開始するためには、以下の諸量に対する初期値が必要である。

- 1)  $k_{\text{eff}}$
- 2)  $\Phi_g^k = \text{col} [\phi_{g,0}^k, \phi_{g,x1}^k, \phi_{g,y1}^k, \phi_{g,z1}^k, \phi_{g,x2}^k, \phi_{g,y2}^k, \phi_{g,z2}^k]$
- 3)  $J_g^{\text{out},k} = \text{col} [J_{g,x+}^{\text{out},k}, J_{g,x-}^{\text{out},k}, J_{g,y+}^{\text{out},k}, J_{g,y-}^{\text{out},k}, J_{g,z+}^{\text{out},k}, J_{g,z-}^{\text{out},k}]$
- 4)  $J_g^{\text{in},k} = \text{col} [J_{g,x+}^{\text{in},k}, J_{g,x-}^{\text{in},k}, J_{g,y+}^{\text{in},k}, J_{g,y-}^{\text{in},k}, J_{g,z+}^{\text{in},k}, J_{g,z-}^{\text{in},k}]$

MOSRA-Lightでは、次の3通りの場合について、上記のパラメータに対する適切な初期値を与えるようにしている。

- 1) 参照できる初期値が何も無い場合(コールドスタート)
- 2) 炉心の幾何形状が全く同じ場合で、断面積が異なるが体系全体の中性子束分布は大きく変わらない場合(初期値参照コールドスタート)
- 3) 計算時間の制限等により中断した計算を継続する場合(リスタート)

##### (1) コールドスタートの場合

参照とする初期値が全く無い場合には、MOSRA-Lightコードは、ノード平均中性子束を空間とエネルギーに関して一定として初期値を与える。また、実効中性子増倍率の初期値には、臨界状態を想定して1.0を与える。

$$\tilde{\phi}_{g,0}^k = C \quad (\text{constant for all } g \text{ and } k) \quad (4.2.45)$$

$$\tilde{k}_{\text{eff}} = 1.0 \quad (\text{注: 記号 } \tilde{\cdot} \text{ は初期値を意味するものとする}) \quad (4.2.46)$$

外部反復計算では、体系全体の中性子生成率が $k_{\text{eff}}$ に一致するように、中性子束が規格化される(2.9.1式参照)。したがって、

$$\sum_{k=1}^N \sum_{g=1}^G v \Sigma_{f,g}^k \tilde{\phi}_{g,0}^k V^k = \tilde{k}_{\text{eff}} \quad (4.2.47)$$

これに、(4.2.45)式と(4.2.46)式を代入すれば、Cが次のように定まる。

$$\tilde{\phi}_{g,0}^k = C = \frac{1}{\sum_{k=1}^N \sum_{g=1}^G v \Sigma_{f,g}^k V^k} \quad (4.2.48)$$

中性子束がノード内で一様であるとすれば、体積平均中性子束とノード境界の表面平均中性子束は一致する。したがって、例えばx方向について、

$$\tilde{\phi}_{g,x+}^k = \tilde{\phi}_{g,x-}^k = \tilde{\phi}_{g,0}^k \quad (4.2.49)$$

であり、これらの表面平均中性子束は表面平均部分中性子流との関係にある(2.1.17式、2.1.19式参照)。

$$\tilde{\phi}_{g,x\pm}^k = 2(\tilde{J}_{g,x\pm}^{out,k} + \tilde{J}_{g,x\pm}^{in,k}) \quad (4.2.50)$$

また、中性子束が一様であるため正味中性子流はゼロであり、ノード境界で表面平均部分中性子流がバランスする。

$$\tilde{J}_{g,x\pm}^{out,k} = \tilde{J}_{g,x\pm}^{in,k} \quad (4.2.51)$$

よって、x方向の表面平均部分中性子流の初期値は、体積平均中性子束の初期値を使用して、次のように与えられる。

$$\tilde{J}_{g,x\pm}^{out,k} = \tilde{J}_{g,x\pm}^{in,k} = \frac{\tilde{\phi}_{g,0}^k}{4} \quad (4.2.52)$$

ただし、真空境界や内部黒体に接するノード表面においては、それぞれの境界処理に応じた部分中性子流を与える(第2.8.1節、第2.8.5節参照)。

2次のNEMで計算する場合には、以上で反復計算に必要な初期値が全て揃うことになる。4次のNEMで計算するには、更に高次モーメント中性子束の初期値が必要になる。これらは、その定義(2.4.7式)と重み関数の特性(2.3.6式)から、ノード内の中性子束が一様の場合には、全てゼロとなる。

以上をまとめると、コールドスタートの場合の初期値は、中性子束がエネルギー的にも空間的にも一様であると仮定して、以下の式で与える。

$$1) \tilde{k}_{eff} = 1.0 \quad (4.2.53a)$$

$$2) \tilde{\Phi}_g^k = \text{col} \left[ \tilde{\phi}_{g,0}^k, 0, 0, 0, 0, 0, 0, 0 \right] \quad (4.2.53b)$$

$$3) \tilde{J}_g^{in,k} = \text{col} \left[ \tilde{\phi}_{g,0}^k/4, \tilde{\phi}_{g,0}^k/4, \tilde{\phi}_{g,0}^k/4, \tilde{\phi}_{g,0}^k/4, \tilde{\phi}_{g,0}^k/4, \tilde{\phi}_{g,0}^k/4 \right] \quad (4.2.53c)$$

$$4) \tilde{\mathbf{J}}_g^{\text{out},k} = \text{col} \left[ \tilde{\phi}_{g,0}^k / 4, \tilde{\phi}_{g,0}^k / 4 \right] \quad (4.2.53d)$$

ただし、 $\tilde{\phi}_{g,0}^k$ は、(4.2.48)式で与えられる。

随伴問題(第4.2.2節参照)においては、核分裂スペクトル  $\chi_g^k$  と生成断面積  $v\Sigma_{f,g}^k$  を入れ替えて計算を行うため(4.2.20式参照)、(4.2.48)式から随伴中性子束の初期値  $\tilde{\phi}_{g,0}^k$  は次の式で与えられる。

$$\tilde{\phi}_{g,0}^k = \frac{1}{\sum_{k=1}^N \sum_{g=1}^G \chi_g^k V^k} = \frac{1}{V^{\text{fissile}}} \quad (V^{\text{fissile}} : \text{体系中の全核分裂領域体積}) \quad (4.2.54)$$

## (2) 初期値参照コールドスタートの場合

断面積を部分的に変更して炉心の幾何形状が全く同じ複数の計算ケースを行う場合などでは、前の計算ケースで得られた解を後の計算ケースの初期値として利用することができる。MOSRA-Lightでは、1つの計算ケースが終了すると、そのケースの収束解である  $k_{\text{eff}}$ 、 $\Phi_g^k$ 、 $J_g^{\text{in},k}$ 、 $J_g^{\text{out},k}$  などがバイナリーファイルに収納される。これらは、MOSRA-Lightとは別に用意されるユーティリティコードによる反応率計算などに利用できる他、同じ幾何形状の問題での初期値としてそのまま利用可能である。

また、幾何形状が同じ場合で、2次のNEMと4次のNEMの計算結果を比較する場合にも、何れかの解を他方の初期値として利用することができる。2次のNEMの収束解を4次のNEMの初期値に利用する場合には、高次モーメント中性子束の初期値は(4.2.43)式と(4.2.44)式で与えられる。

初期値として参照する計算ケースはユーザーが与えるが、該当するファイルが存在しない場合には、自動的に(1)で記述した通常のコールドスタートの扱いで計算が実行される。

## (3) リスタート計算の場合

計算時間の制限等により中断した計算は、リスタート計算オプションにより続行することができる。MOSRA-Lightでは、外部反復計算毎に、それまでの経過時間と1回の外部反復計算に要する平均時間を測定しており、以下の条件式により次の外部反復計算に進むかどうかを判定する。

$$\begin{cases} t_{\text{next}} > t_{\text{limit}} & : \text{continue} \\ t_{\text{next}} \leq t_{\text{limit}} & : \text{stop} \end{cases} \quad (4.2.55)$$

ただし、

$$t_{\text{next}} \equiv \left\{ t_{\text{now}} + 2 \times \text{Integer} \left( \frac{t_{\text{now}}}{L} \right) \right\} / 60 \quad (4.2.56)$$

$t_{\text{now}}$  : 現在までのCPU経過時間(sec.)

$L$  : 現在の外部反復計算回数

$t_{\text{limit}}$  : ユーザーが与える計算時間の上限(min.)

である。(4.2.56)式で、1回の外部反復計算に要する平均時間( $t_{\text{now}}/L$ )の2倍をとっているのは、その後のプリント出力やファイル出力に要する時間を加味するためである。反復計算を中断した後、仮に経過時間が  $t_{\text{limit}}$  を越えても、リスタートファイルを出力して計算は正常に終了するが、計算機システムが別に計算時間を制限しているような環境では、 $t_{\text{limit}}$  の制限をシステムの制限時間より短くしておかなければならぬ。

$t_{\text{next}} \leq t_{\text{limit}}$  の場合には、収束判定条件(第4.2.7節参照)を満足していないくとも、外部反復計算を中断し、リスタートファイルが作成される。計算を中断するタイミングは、内部反復計算や群に関するループの途中ではなく、必ず外部反復計算の区切れとなる。従って、リスタートファイルは特殊なものではなく、計算が正常に収束した場合と同様の  $k_{\text{eff}}$ 、 $\Phi_g^k$ 、 $J_g^{\text{in},k}$ 、 $J_g^{\text{out},k}$  を含むバイナリーファイルである。リスタート計算では、これら未収束のパラメータをそのまま初期値として利用し計算が続行される。

リスタートファイルを収束時の出力ファイルと全く同じ形式にすることにより、ユーティリティコードの入力ファイル、(2)で述べた別計算ケースの初期値ファイル、及び同じ計算ケースのリスタートファイルとして共用できる。計算が要求精度内で収束した場合でも、後に収束判定条件をより厳しくする必要が生じた場合に、そのまま初期値として利用できる。また、プリント出力結果を保存しておかなくても、このファイルが残されていれば、数回の外部反復計算により短時間で計算結果を再現することができる。

#### 4.2.7 収束判定と反復計算の制御

MOSRA-Lightの計算は3種類の反復計算(内部反復計算、熱群反復計算、外部反復計算)で構成される。熱群反復計算については、既に第4.2.3節で詳しく述べた通りである。ここでは、内部反復計算と外部反復計算の収束判定法、反復計算の制御方法について記述する。

##### (1) 外部反復計算の収束判定

MOSRA-Lightが与えられた計算時間の制限内で計算を終了するのは、以下の何れかの場合である。

- 1) 外部反復計算がユーザーの要求精度内で収束した場合(正常終了)
- 2) 外部反復計算回数Lが上限値 $L_{max}$ を越えた場合(準異常終了)
- 3) 固有値が上限値または下限値を越えた場合(異常終了)

上限値や下限値は、ユーザーが特に指定しない限り、コードのデフォルト設定値が採用される。2)の場合には、メッセージが出力されて反復計算を中断するが、要求精度は満足されていないものの、計算結果のプリントやバイナリーファイルの出力は行われ、リスタート計算が可能である。一方、3)の場合には、反復計算の情報の後にエラーメッセージがプリントされ、計算結果のプリント、バイナリーファイルの出力は行われない。1)の外部反復計算の収束判定は、以下に示す方法で行われる。

$$|k_{eff}(L) - k_{eff}(L-1)| \leq \epsilon_{outer}^k \text{ (デフォルト値: 1.0E-5)} \quad (4.2.57)$$

$$\max_{g, k} \left| \frac{\phi_{g,0}^k(L) - \phi_{g,0}^k(L-1)}{\phi_{g,0}^k(L-1)} \right| \leq \epsilon_{outer}^p \text{ (デフォルト値: 1.0E-4)} \quad (4.2.58a)$$

または、

$$\max_{g, k} \left| \frac{F_g^k(L) - F_g^k(L-1)}{F_g^k(L-1)} \right| \leq \epsilon_{outer}^p \quad (4.2.58b)$$

または、

$$\max_{g, k} \left| \frac{J_{g,x\pm}^{out,k}(L) - J_{g,x\pm}^{out,k}(L-1)}{J_{g,x\pm}^{out,k}(L-1)}, \frac{J_{g,y\pm}^{out,k}(L) - J_{g,y\pm}^{out,k}(L-1)}{J_{g,y\pm}^{out,k}(L-1)}, \frac{J_{g,z\pm}^{out,k}(L) - J_{g,z\pm}^{out,k}(L-1)}{J_{g,z\pm}^{out,k}(L-1)} \right| \leq \epsilon_{outer}^p \quad (4.2.58c)$$

ただし、 $F_g^k(L)$ はL回目の反復計算で得られる核分裂源(4.2.29式)である。収束判定条件として(4.2.58a)式～(4.2.58c)式の何れを採用するかはユーザーの選択による。 $\epsilon_{outer}^k$ と $\epsilon_{outer}^p$ は収束判定因子であり、(4.2.57)式と(4.2.58)式の双方が2回連続して満足された場合に、計算が収束したものと判定される。2回連続としているのは偶発的にこれらの式を満足する場合を避けるためである。反射体領域が大きい問題では、一般に炉心部の中性子束に比べて反射体領域の中性子束の収束が遅い。このため、中性子増倍率や出力分布などの炉心特性にほとんど影響を及ぼさない反射体領域の中性子束が全体の収束速度を律速することがある。(4.2.58b)式による収束判定はこれを避けるためのものであり、非核分裂領域の中性子束分布を特に必要としない場合には、こちらを使用する方が効率的である。

MOSRA-Lightに限らず、反復計算の新値と旧値で比較する収束判定方法では、 $\epsilon_k$ や $\epsilon_p$ は真の解からの誤差を保証するものではない。また、現在のMOSRA-Lightでは、主要な計算部分で倍精度演算をしているが、断面積などは単精度の扱いであり、1.0E-6以下の収束判定因子を使用する場合には、計算の収束は保証されない。

## (2) 内部反復計算の収束判定

内部反復計算は、一般的に言えば、着目する群の中性子源分布  $Q_g(\mathbf{r})$  が正しく与えられたものと仮定して、以下の拡散方程式を1群の固定源問題として解き(空間スワイープ)、着目群の中性子束や中性子流の空間分布を暫定的に求める役割をする。

$$\nabla \cdot J_g(\mathbf{r}) + \Sigma_{r,g}(\mathbf{r})\phi_g(\mathbf{r}) = Q_g(\mathbf{r}) \quad (4.2.59)$$

しかし、 $Q_g(\mathbf{r})$  が正しい値を持つのは、外部反復計算が収束した時であり、その進行過程では、内部反復計算を必ずしも完全に収束させる必要はない。MOSRA-Lightでは、以下の条件が満足された場合に、着目群に対する内部反復計算を終了させる。

- 1) 内部反復計算がコードが与える要求精度内で収束した場合
- 2) 内部反復計算回数  $I$  がユーザーまたはコードが与える上限値  $I_{max}$  に達した場合

内部反復計算の収束判定は、各エネルギー群毎に、以下の式により行われる。

$$\max_k \left| \frac{\phi_{g,0}(I) - \phi_{g,0}(I-1)}{\phi_{g,0}(I-1)} \right| \leq \varepsilon_{inner}(L) \quad (4.2.60)$$

内部反復回数の上限  $I_{max}$  と収束判定因子  $\varepsilon_{inner}$  は、オプションにより外部反復を通して一定値とすることもできるが、外部反復計算の収束状況と連動して、コードが自動的かつ動的( $L$  に依存)に調整するようにしている。

## (3) 反復計算の自動制御

外部反復計算を安定に収束させるためには、理想的には内部反復計算が十分な精度で収束するまで待つことが望ましい。しかし、外部反復の初期においては、中性子源  $Q_g(\mathbf{r})$  は未だ収束解に十分漸近しておらず、内部反復を精度良く行うのは効率的ではない。また、外部反復が進む過程で内部反復も徐々に収束解に漸近することが期待できる。このため、内部反復回数の上限を1~2回程度にすることも珍しくはない。

外部反復計算の収束性は、様々な要因に支配されるが、ノード法の場合にはメッシュサイズに対する依存性に注意を要する。一般に、メッシュサイズが大きい場合には、多くの内部反復を必要としないが、メッシュサイズが小さくなるほど、内部反復計算の回数を多くしないと外部反復計算が収束しなくなる傾向がある。大雑把な方針を言うなら、粗メッシュの場合には、内部反復計算を少なくして計算時間を短縮し、メッシュサイズが小さい場合で安定な収束を期待する場合には、内部反復計算回数を多くすることが望ましい。

MOSRA-Lightでは、反復計算の収束性に関するパラメータ(反復上限値や収束判定因子など)を自由に設定することが可能である。しかし、ユーザーの立場からすれば、あまりこのようなパラメータの設定に煩わされたくないであろう。このため、ユーザーが特に指定しない限り、以下に述べる経験的な方法により反復計算の制御をコードが自動的に行うようしている。

先ず、外部反復計算の収束状況を以下の3つのレベルに分類する。

レベル-0：通常時(レベル-1とレベル-2以外の場合)

レベル-1：レベル-0に対するパラメータの自動調整の結果、外部反復の収束が停滞していると判断された場合

レベル-2：外部反復計算の上限回数の半分を経過しても収束解が得られない場合

外部反復計算が順調に収束方向にある通常時(レベル-0)においては、L回目の外部反復計算が終了した時点で、次の外部反復計算における内部反復計算の収束判定因子 $\epsilon_{inner}(L+1)$ を次の式により与える。

$$\epsilon_{inner}(L+1) = \max \left\{ \epsilon_{inner}^{tmp}, \frac{\epsilon_{outer}^p}{10} \right\} \quad (4.2.61)$$

$$\epsilon_{inner}^{tmp} = \min \left\{ \frac{\delta(L)}{10}, \epsilon_{inner}(L=1), \epsilon_{inner}(L) \right\} \quad (4.2.62)$$

$$\delta(L) = \max_{g, k} \left| \frac{\phi_{g,0}^k(L) - \phi_{g,0}^k(L-1)}{\phi_{g,0}^k(L-1)} \right| \quad (4.2.63)$$

ただし、 $\epsilon_{inner}(L=1)$ は入力で与える初期値(デフォルト値：1.0E-3)である。これらの式は次のような意味を持つ。正確な中性子源分布が得られていない外部反復の初期においては、内部反復の収束を厳しくすることは効率的ではない。そこで、外部反復計算で得られたポイント中性子束の最大変化割合の1/10まで、 $\epsilon_{inner}(L+1)$ をあまく設定することを許容するが、以前よりもあまくすることは避ける。ただし、この判定により定まる $\epsilon_{inner}^{tmp}$ が最終的なポイント中性子束(またはポイント核分裂源)の収束判定因子 $\epsilon_{outer}^p$ (4.2.58式参照)の1/10より小さくなった場合には、 $\epsilon_{outer}^p/10$ を内部反復計算の収束判定因子として採用する。

レベル-0の反復計算制御を採用すると、場合によっては外部反復計算の収束が滞ることがある。これは、問題の性質と入力パラメータの値にも依存するが、外部反復における $\delta(L)$ と内部反復の収束判定因子 $\epsilon_{inner}$ が回帰的であるため、たまたまこれらがバランスすると、 $\epsilon_{inner}$ が不変となり、外部反復計算がそれ以上収束しなくなるためである。このような状態に陥った状態をレベル-1とし、具体的には、初期の15回を除き10回以上外部反復計算を繰り返しても $\epsilon_{inner}$ が変わらない場合、内部反復計算の収束判定を以下のように厳しくする。

$$\epsilon_{\text{inner}}(L+1) = \max \left\{ \epsilon_{\text{inner}}^{\min}, \frac{\epsilon_{\text{inner}}(L)}{10} \right\} \quad (4.2.64)$$

$$I_{\max} \rightarrow 2I_{\max} \quad (4.2.65)$$

ただし、 $\epsilon_{\text{inner}}^{\min}$ は内部反復収束判定因子の最小限界値であり、以下のように与える。

$$\epsilon_{\text{inner}}^{\min} = \frac{\epsilon_{\text{outer}}^p}{100} \quad (4.2.66)$$

また、 $I_{\max}$ は内部反復計算回数の上限値で、(4.2.65)式は以降の上限値を入力値の2倍にすることを意味している。これらの処理を行った後は、

$$\epsilon_{\text{inner}}(L+1) = \min \left\{ \frac{\delta(L)}{10}, \epsilon_{\text{inner}}(L=1), \epsilon_{\text{inner}}(L) \right\} \quad (4.2.67)$$

により、内部反復計算の収束判定因子を決定する。

レベル-3は、外部反復計算の上限値 $L_{\max}$ の半分を経過( $L > L_{\max}/2$ )しても、計算が収束しない場合であり、この場合には内部反復計算の収束判定因子を動的に変更することを止め、強制的に以下のようないかだいな設定とする。

$$\epsilon_{\text{inner}}(L+1) = \epsilon_{\text{inner}}^{\min} \quad (4.2.68)$$

$$I_{\max} \rightarrow 2I_{\max} \quad (4.2.69)$$

以上の手法は全く経験的なものであり、理論的な根拠はない。概して言えば、計算時間の短縮よりは、むしろ外部反復計算の収束安定性を重視したものである。これは、MOSRA-Lightを核熱結合動特性計算や炉心燃焼計算などのコードに組み込む場合に、核熱フィードバックや燃焼により断面積がどのように変化しても、安定に収束解を出力させるための配慮でもある。

#### (4) 反復計算モードの選択

反復計算の収束は、その高速性と安定性がトレードオフの関係にあることが多い。例えば、安定した収束を望むのであれば、内部反復計算の回数を多くする方が望ましいが、これにより計算時間は長くなる。逆に計算時間の短縮のため内部反復計算をおろそかにすると、収束解が得られなくなることがある。内部反復回数の他、熱群反復回数の上限、各種反復計算の収束判定因子、加速法に関わる調整因子など、収束制御に関わるパラメータは

数多くある。これらの最適値は問題に依存するため、ユーザーはその設定に悩まされることが少なくない。

収束制御パラメータをどのように設定するかは、コードの利用目的や使用する計算機にも依存する。例えば、初めての問題で少数回の計算の実行で目的を終了してしまうようなユーザーは、少々計算時間が長くなってしまっても、手っ取り早く安定な収束解を得たいと考えるであろう。逆に設計計算のパラメトリックサーベイなどのように、類似の問題を数多く実行する場合には、経験的に最適な収束制御パラメータが予測でき、その調整により計算の高速化を計ろうとするであろう。また、ベクトル計算機を使用する場合には、スカラーカルカの場合に比べて内部反復計算の所要時間が極めて短いため、その上限回数を多く設定しても、全体の計算時間に大きな影響を及ぼさない。このような場合には、収束の安定性を重視した設定をする方が得策である。いずれにせよ、ユーザーにとっては、問題ごとに収束制御パラメータを調整して入力することは煩わしい作業にすぎない。MOSRA-Lightでは、こうしたユーザーの諸事情に応じて収束制御パラメータの入力負担を軽減させるため、反復計算制御に関して以下に述べる4つの計算モードを用意している。

- 1) 収束制御パラメータの自動最適化モード
- 2) 安定収束優先モード
- 3) 高速計算優先モード
- 4) 収束制御パラメータのユーザー調整モード

自動最適化モードでは、ユーザーが入力するノード分割やエネルギー群数などの情報を基にコードが適切な収束制御パラメータを自動設定する(筆者の経験に基づく)。また、外部反復計算の過程で、その設定により収束の傾向が見られないとコードが判断した場合には、収束制御パラメータを次に述べる安定収束優先モードの設定に変更して計算を最初からやり直し収束に導く。安定収束優先モードでは、内部反復計算回数などの上限値を大きくした設定としており、高速計算性を犠牲にして確実に収束解を得られるようにしている。これとは逆に、高速計算優先モードでは、内部反復回数の上限を2回に設定しており、これにより計算時間は短くなるが、問題によっては収束しなくなることもある。1)~3)の計算モードを選択する場合には、ユーザーは数ある収束制御パラメータを入力する必要は無い。ユーザー調整モードでは、ユーザーが入力する収束制御パラメータを使用して計算を実行する。

これらの計算モードの選択指針を述べる。初めての問題に対しては1)の利用を推奨する。これにより収束解が得られない場合には2)を利用する。あるいは、ジョブ投入型の計算機などで、計算結果を即座に要求しない場合には、初めから2)を利用する方が賢明であろう。安定収束優先モードでも収束解が得られない場合の対処については、付録-B.2を参照して頂きたい。類似の問題を繰り返し行う場合には、代表的な問題に対して先ず高速計算優先モードにより収束解が得られるかを確認して、以降の問題に対してもこれを利用すると効率的である。あるいは、経験的により効率的な計算をするための収束制御パラメータが予測できる場合には、4)のユーザー調整モードを使用すると良いであろう。

#### 4.2.8 SRACコードの巨視的断面積ファイルの利用

MOSRA-Lightでは、計算体系の巨視的断面積を標準入力で与えられる他、SRACコード<sup>25)</sup>が出力する巨視的断面積のバイナリーファイルをそのまま利用することができる。SRACコードは、16種類の格子形状に対する衝突確率計算モジュールの他、ANISN(1次元Sn輸送)、TWOTRAN(2次元Sn輸送)、CITATION(多次元拡散)などの既存コードを統合したコードで、炉型に対する汎用性が高く、また超詳細群衝突確率計算による実効共鳴断面積の計算、格子燃焼計算などの機能を有している。利用できる群定数ライブラリーも豊富であり、現在までにENDF/B-IV, ENDF/B-V, ENDF/B-VI, JENDL-3.1, JENDL-3.2, JEF-2.2に基づくライブラリーが用意されている。各計算モジュールにより得られる以下のような巨視的断面積をMOSRA-Lightに直接利用することができる。

- 1) 各モジュールによる体系計算で指定された領域に対する均質化詳細群断面積
- 2) 1)の断面積を中性子スペクトルを重みに群縮約した少数群断面積
- 3) 格子燃焼計算により得られる1)及び2)の断面積

ただし、現在のSRACは不連続因子を計算してファイルに収納する機能は持っていない。不連続因子を使用する場合には、これをMOSRA-Lightの標準入力で与えるか、定められた形式でユーザーが不連続因子ファイル(バイナリー)を用意する必要がある。

SRACの断面積ファイルは、1階の階層構造を持ち、1つのディレクトリファイル内にメンバーと呼ばれる複数のサブファイルが存在する。メンバーのファイル名は、SRACの規約に従う8文字の英数字で構成される。例えば、均質化断面積メンバーには以下のような名前が付けられる。

メンバー名 : caseebxp (= case/e/b/x/p)

case : ユーザーが自由に定める計算ケースの名前

e : 断面積のエネルギー範囲を表す。e=F(高速群のみ)、e=A(高速群+熱群)

b : 格子燃焼計算における燃焼度に対応する英数字(0,1,2,3,...,9,A,B,...,Z)

x : 均質化領域に対応する英数字(1,2,3,...,9,A,B,...,Z)

p : 断面積タイプを表す。p=0(少数群P0断面積)、p=2(詳細群P0断面積)、p=N  
(少数群[n,2n]反応巨視的断面積、p=M(詳細群[n,2n]反応断面積)など

断面積メンバーの具体的な構造や内容については、第7.3節で詳しく述べる。MOSRA-LightでSRACが出力する断面積を利用する場合には、ただメンバー名を指定すれば良い。

#### 4.2.9 (n,2n)反応の補正

(n,2n)反応による効果は、熱中性子炉では無視できることが多いが、スペクトルの硬い炉心やベリリウムなどを含む体系では無視できないことがある。(n,2n)反応を含む拡散方程式は(4.2.70)式で与えられる。通常の決定論的な輸送コードや拡散コードは、(n,2n)反応を直接には扱わないため、(4.2.71)式のように散乱マトリックスに(n,2n)反応の寄与分を加味しておく。これにより(4.2.70)式は(4.2.72)式のように一般的な拡散コードで扱える方程式に変形できる。この際、自群散乱を扱わない拡散コードでは、除去断面積の項を(4.2.73)式のように補正しておく必要がある。

$$\begin{aligned} -\nabla \cdot D_g \nabla \phi_g + \left\{ \sum_{g' \neq g} \Sigma_{s, g \rightarrow g'} + \sum_{g'} \Sigma_{n2n, g \rightarrow g'} \right\} \phi_g \\ = \frac{\chi_g}{k_{\text{eff}}} \left\{ \sum_{g'} v \Sigma_{f, g} \phi_{g'} \right\} + \sum_{g' \neq g} \Sigma_{s, g' \rightarrow g} \phi_{g'} + \sum_{g'} 2 \Sigma_{n2n, g' \rightarrow g} \phi_{g'} \end{aligned} \quad (4.2.70)$$

$$\Sigma_{s, g \rightarrow g'} \equiv \Sigma_{s, g \rightarrow g'} + 2 \Sigma_{n2n, g \rightarrow g'} \quad (4.2.71)$$

$$-\nabla \cdot D_g \nabla \phi_g + \Sigma_{r, g} \phi_g = \frac{\chi_g}{k_{\text{eff}}} \left\{ \sum_{g'} v \Sigma_{f, g} \phi_{g'} \right\} + \sum_{g' \neq g} \Sigma_{s, g' \rightarrow g} \phi_{g'} \quad (4.2.72)$$

$$\text{ただし、} \Sigma_{r, g} \equiv \left\{ \Sigma_{a, g} + \sum_{g' \neq g} \Sigma_{s, g \rightarrow g'} \right\} - \sum_{g'} \Sigma_{n2n, g \rightarrow g'} \quad (4.2.73)$$

SRACコードにより得られる断面積ファイルをMOSRA-Lightに利用する場合(第4.2.8節参照)には、このような(n,2n)反応の補正が自動的に行われる。SRACが出力する断面積ファイル内の散乱マトリックスは、(4.2.71)式の補正が既に行われたものであり、MOSRA-Lightでは、(4.2.73)式の補正を行って除去断面積を作成する。したがって、MOSRA-Lightがプリントまたはファイルに出力する除去断面積は、(4.2.73)式に従うものである。ただし、断面積ファイルに(n,2n)反応断面積を収納するメンバー(8文字目がNまたはMのメンバー)が存在しない場合には、この補正是行われない。また、標準入力により断面積を与える場合にもこの補正是行われず、特に必要がある場合には、(4.2.71)式と(4.2.73)式の補正を加えた断面積を入力する(除去断面積は入力項目に無いため吸収断面積を補正する)。

#### 4.2.10 計算結果の規格化

本節では、MOSRA-Lightが输出する中性子束分布や出力密度などの数値の規格化方法を述べ、プリンターやファイルに書き出される計算結果の定義を明確にしておく。

通常の固有値問題の場合、外部反復計算の直後には、中性子束分布は必然的に体系内の

中性子生成率が中性子増倍率に一致するように規格化される(第2.9節参照)。

$$\sum_{k=1}^N \sum_{g=1}^G v \Sigma_{f,g}^k \phi_{g,0}^k V^k = k_{\text{eff}} \quad (4.2.74)$$

しかし、実際の中性子束の値は、原子炉の熱出力  $P_{\text{th}}$  に依存し、以下の式に従う。

$$P_{\text{th}} = \sum_{k=1}^N \sum_{g=1}^G \kappa^k \Sigma_{f,g}^k \Phi_{g,0}^k V^k \quad (4.2.75)$$

ただし、

- $P_{\text{th}}$  : 体系全体の熱出力 (W)
- $\kappa^k$  : 1核分裂当たりに放出されるエネルギー (J/fission)
- $\Phi_{g,0}^k$  : 中性子束の絶対値 ( $n/cm^2 s$ )

である。 $\kappa^k$ は、約200MeV( $=3.204E-17[J]$ )程度の値として良く知られているが、正確にはウランやプルトニウムなどの核分裂核種により異なり燃料の組成に依存する。MOSRA-Lightでは、 $\kappa^k$ をノード毎に割り当てられる物質(断面積セット)に対する入力としている。正確な値は、SRACコード<sup>25)</sup>の格子燃焼計算により得ることができる。 $\kappa^k=1.0$  (for all k)とする場合には、出力分布は核分裂反応率の分布として出力される。

相対値として得られるノード平均中性子束  $\phi_{g,0}^k$  を絶対値  $\Phi_{g,0}^k$  に変換する規格化係数を Fact とすると、

$$\Phi_{g,0}^k = \text{Fact} \cdot \phi_{g,0}^k \quad (4.2.76)$$

であり、(4.2.75)式と(4.2.76)式から、ユーザーが入力する熱出力  $P_{\text{th}}$  を使用して、規格化係数 Fact が次式により決定される。

$$\text{Fact} = \frac{P_{\text{th}}}{\sum_{k=1}^N \sum_{g=1}^G \kappa^k \Sigma_{f,g}^k \phi_{g,0}^k V^k} \quad (4.3.77)$$

ノード平均中性子束の他、高次モーメント中性子束、表面平均部分中性子流、表面平均正味中性子流も、外部反復計算の後に得られた値を Fact 倍した値として出力される。この中性子束の値を使用し、ノード平均出力密度  $q'''^k [W/cm^3]$  は次式で計算される。

$$q'''^k = \sum_{g=1}^G \kappa^k \Sigma_{f,g}^k \Phi_{g,0}^k \quad (4.2.78)$$

(4.2.75)式と(4.2.78)式より、 $q''^k$ を体系内で積分すれば総熱出力に一致する。

$$P_{th} = \sum_{k=1}^N q''^k V^k \quad (4.2.79)$$

以上は一般的な扱いであるが、MOSRA-Lightでは、ユーザーの用途に応じて以下の規格化法を選択できるようにしている。

- (a) ユーザー入力の熱出力値に規格化
- (b) 規格化無し (Fact=1)
- (c) 热出力が核分裂領域の体積に一致するように規格化
- (d) 全中性子吸收反応率を1に規格化
- (e) 全核分裂反応率を1に規格化
- (f) 全核分裂生成反応率を1に規格化

(a)の場合は、既に述べたように(4.2.75)式が満足される。固有値計算(forward)の場合に(b)のオプションを選択すると、中性子束のレベルは(4.2.80)式を満足するように規格化され、ノード出力密度 $q''^k$ (4.2.78式参照)はこの中性子束レベルにより決定される。

$$\sum_{k=1}^N \sum_{g=1}^G v \Sigma_{f,g}^k \Phi_{g,0}^k V^k = k_{eff} \quad (4.2.80)$$

固定源問題(第4.2.1節参照)の場合には、中性子束や出力密度の値は、ユーザーが与える固定源の強度(4.2.8式)により決定されるため、無条件に(b)の扱いとなる。また、随伴問題(第4.2.2節参照)の場合には、随伴中性子束は相対値として得られるが、そのレベルに本質的な意味は無いため、(b)の扱いとしている。(d), (e), (f)の場合には、それぞれ、(4.2.81a)式、(4.2.81b)式、(4.2.81c)式が満足されるように規格化係数Factが決定される。

$$\sum_{k=1}^N \sum_{g=1}^G \Sigma_{a,g}^k \Phi_{g,0}^k V^k = Fact \left\{ \sum_{k=1}^N \sum_{g=1}^G \Sigma_{a,g}^k \phi_{g,0}^k V^k \right\} = 1.0 \quad (4.2.81a)$$

$$\sum_{k=1}^N \sum_{g=1}^G \Sigma_{f,g}^k \Phi_{g,0}^k V^k = Fact \left\{ \sum_{k=1}^N \sum_{g=1}^G \Sigma_{f,g}^k \phi_{g,0}^k V^k \right\} = 1.0 \quad (4.2.81b)$$

$$\sum_{k=1}^N \sum_{g=1}^G v \Sigma_{f,g}^k \Phi_{g,0}^k V^k = Fact \left\{ \sum_{k=1}^N \sum_{g=1}^G v \Sigma_{f,g}^k \phi_{g,0}^k V^k \right\} = 1.0 \quad (4.2.81c)$$

(c)の規格化は、炉心平均出力密度に対する位置依存の出力密度の比 $F_{r,z}^k$  (power form factor)を出力させたい場合に使用する。この場合、中性子束レベルの規格化は次の式に従う。

$$\sum_{k=1}^N \sum_{g=1}^G \kappa^k \Sigma_{f,g}^k \Phi_{g,0}^k V^k = \text{Fact} \sum_{k=1}^N \sum_{g=1}^G \kappa^k \Sigma_{f,g}^k \phi_{g,0}^k V^k = \sum_{\substack{k=1 \\ k \in \text{fissile}}}^N V^k \equiv V_{\text{fissile}} \quad (4.2.82)$$

その結果、炉心平均出力密度は、

$$\overline{q}''' = \frac{P_{\text{th}}}{V_{\text{fissile}}} = \frac{\sum_{k=1}^N \sum_{g=1}^G \kappa^k \Sigma_{f,g}^k \phi_{g,0}^k V^k}{V_{\text{fissile}}} = 1.0 \quad (4.2.83)$$

となる。よって、

$$F_{r,z}^k \equiv \frac{q'''^k}{q'''} = q'''^k = \sum_{g=1}^G \kappa^k \Sigma_{f,g}^k \Phi_{g,0}^k \quad (4.2.84)$$

すなわち、総熱出力が核分裂領域の体積に一致するように規格化することにより、出力密度の計算値として、無次元の出力分布係数( $F_{r,z}^k$ )が出力される。

#### [ 対称性や反射境界を利用する場合、および1次元、2次元計算をする場合の注意点 ]

中性子束の規格化(a)に使用される熱出力 $P_{\text{th}}$ (4.2.75式参照)は、MOSRA-Lightに与えられる計算体系における熱出力であり、計算モデルの与え方によっては、実炉心の熱出力と異なることがある。以下に具体的な例を挙げる。なお、実炉心の総熱出力を $P_{\text{th}}^{\text{Real}}$ として、MOSRA-Lightの入力データとすべき $P_{\text{th}}$ と区別する。

##### (1) 反射境界を利用した対称炉心モデルを使用する場合

炉心の水平方向の対称性を考慮して、反射境界条件を利用した1/2炉心モデル、あるいは1/4炉心モデルで計算を行う場合には、それぞれ、 $P_{\text{th}} = P_{\text{th}}^{\text{Real}}/2$ 、 $P_{\text{th}} = P_{\text{th}}^{\text{Real}}/4$ とする。更に、垂直方向についても対称な1/8炉心モデルを使用する場合には、 $P_{\text{th}} = P_{\text{th}}^{\text{Real}}/8$ とする。一般的には、次の式で与えられる。

$$P_{\text{th}} = P_{\text{th}}^{\text{Real}} \left( \frac{V_{\text{fissile}}^{\text{Cal.}}}{V_{\text{fissile}}^{\text{Real}}} \right) \quad (4.2.85)$$

ただし、 $V_{\text{fissile}}^{\text{Real}}$ は実炉心の核分裂領域体積、 $V_{\text{fissile}}^{\text{Cal.}}$ は計算モデルにおける核分裂領域体積である。

## (2) 1次元または2次元計算の場合

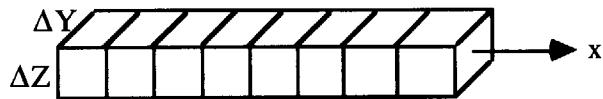
MOSRA-Lightでは、1次元計算や2次元計算に特化したプログラムは用意していない。このため、1次元または2次元計算を行う際には、考慮しない方向にも仮想的な有限炉心長を設定し、反射境界条件を適用することになる。

例えば、x-y面についての2次元計算を行う場合には、右図に示すように、任意な長さの炉心長 $\Delta Z$ を与え、z軸方向の外部境界条件を両側共に反射境界に設定する。 $\Delta Z$ のノード分割数も任意であるが、1分割でも問題はない。このような場合には、MOSRA-Lightに与えるべき熱出力は、以下の式で考えると分かり易い。

$$P_{th} = \bar{q}'' V_{fissile}^{Cal.} = \left( \frac{P_{th}^{Real}}{V_{fissile}^{Real}} \right) V_{fissile}^{Cal.} = P_{th}^{Real} \left( \frac{\Delta Z}{L_z^{Real}} \right) \quad (4.2.86)$$

ちなみに、2次元計算に特化された他コードでは、考慮しない方向について、単位長の扱い( $\Delta Z=1$ )をしており、実炉心の熱出力を炉心有効長( $L_z^{Real}$ )で割った熱出力を与えることが多い。これは、MOSRA-Lightの3次元計算で $\Delta Z=1$ とする計算と同じ結果を与える。

1次元計算の場合も同様で、例えば下図のようなx軸方向1次元計算を考える。また、実炉心のy軸方向及びz軸方向の炉心有効長を、それぞれ $L_y^{Real}$ 、 $L_z^{Real}$ とすれば、 $P_{th}$ は(4.2.87)式で与えられる。



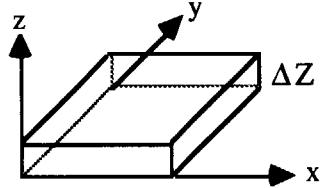
$$P_{th} = \bar{q}'' V_{fissile}^{Cal.} = \left( \frac{P_{th}^{Real}}{V_{fissile}^{Real}} \right) V_{fissile}^{Cal.} = P_{th}^{Real} \left( \frac{\Delta Y}{L_y^{Real}} \right) \left( \frac{\Delta Z}{L_z^{Real}} \right) \quad (4.2.87)$$

1次元または2次元計算では、考慮しない方向の炉心長を任意に与えることができるが、絶対値として正しい出力密度値を出力させるためには、ユーザーが設定した炉心長( $\Delta Y$ や $\Delta Z$ )に依存した熱出力を与える必要がある。

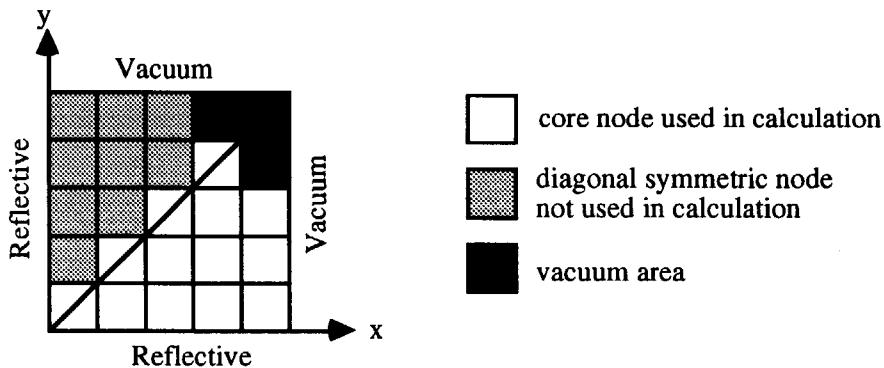
また、1次元計算や2次元計算を、便宜的に反射境界条件を使用した3次元計算で行うため、全てのノードが外部境界に接することになり、3次元計算の場合のように境界分離チェックカードスウェープ(第3.4.2節参照)による高速ベクトル計算の効果は得られない。これを避けるためには、1次元計算や2次元計算に特化したプログラムを新たに加える必要がある。しかし、そもそも1次元や2次元計算ではノード総数が少なく、計算コストはあまり問題にならない。

## (3) 対角境界条件を使用する場合

x-y面上で対角対称境界条件(第2.8.6節参照)を適用する場合には、特に注意が必要である。例えば、下図のような1/4炉心モデルを考える。ただし、話を簡単にするために、反射体が無



い裸の炉心で、z軸方向には全炉心長をモデル化しているものとする。



上図において、黒色の領域は真空領域または完全黒体ノードである。また、白色のノードは、MOSRA-Lightが計算の対象とするノードである。灰色のノードは、炉心領域であるが、対角対称境界条件を使用する場合には、MOSRA-Lightの計算からは除外される。この場合、1/8炉心モデルとも言えるが、 $P_{th} = P_{th}^{Real}/8$ とはならない。これは、MOSRA-Lightでは、対角線上のノードを三角ではなく、あくまで四角形のノードとして扱うためである。よって、正しくは次の式で $P_{th}$ を与えることになる。

$$P_{th} = \bar{q}^m V_{fissile}^{Cal.} = \left( \frac{P_{th}^{Real}}{V_{fissile}^{Real}} \right) V_{fissile}^{Cal.} = P_{th}^{Real} \left( \frac{S_{fissile}^{Cal.}}{S_{fissile}^{Real}} \right) \quad (4.2.88)$$

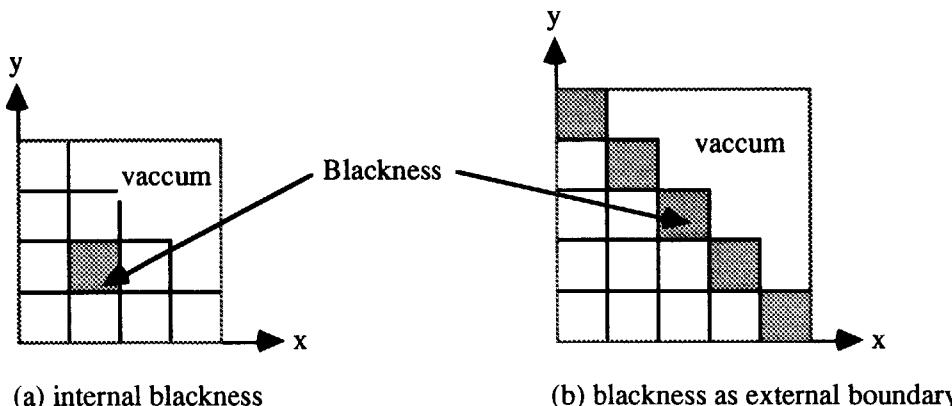
ただし、 $S_{fissile}^{Real}$  はx-y面から見た実炉心の核分裂領域面積であり、 $S_{fissile}^{Cal.}$  はx-y面から見た計算対象とする核分裂領域ノード面積である。すなわち、上の例の場合、白色ノードの全面積が $S_{fissile}^{Cal.}$  に相当する。

(1)～(3)のケースが複合する場合にも同様な考え方をすれば良い。結局、(4.2.85)式～(4.2.88)式が意味することは同じであり、MOSRA-Lightが計算対象とする全核分裂領域ノードの体積と実炉心の核分裂領域体積の比を、 $P_{th}^{Real}$ に乘じて $P_{th}$ とすれば良い。

#### 4.2.11 内部黒体(Internal Blackness)の拡張利用

内部黒体は、第2.8.5節で詳しく述べたように、本来は炉心体系の外表面ではなく炉心内部に存在する一種の境界条件である(下図(a)参照)。完全な黒体と異なるのは、エネルギー一群に依存して、通常の断面積を使用した扱いも可能である点である。CITATIONコード<sup>1)</sup>が、同様な機能を有しているが、黒体の種類は一種類に限定されている。MOSRA-Lightでは、断面積や外挿距離定数が異なる複数の黒体を扱うことができる。また、下図(b)に示すように、黒体を外表面に配置することにより、真空境界条件やアルベド境界条件として代用す

ることができる。



### (1) 内部黒体として利用する場合

強い熱中性子吸收材が炉心内部に塊として存在し、断面積が他の物質と均質化されていないような場合には、一般に拡散近似の精度が低下する。このような場合には、該当する領域とエネルギー一群を内部黒体として扱うことにより、外挿距離境界条件を適用することができる。この際、4次のNEMの場合には、内部黒体に隣接するノードにおいて2次式によるトランスペース漏洩項のフィッティング(QTLA)ができないため、MOSRA-Lightでは低次のフィッティングで代用している(第2.8.5節参照)。このことが、メッシュ幅の効果として現れることがあるため注意を要する。一方、QTLAが不要な2次のNEMの場合(第4.2.4節参照)には、原理的にCITATIONコードの内部黒体と全く同じ扱いとなる。

### (2) 外部真空境界条件として利用する場合

体系外表面を真空境界として扱う場合、MOSRA-Lightでは、以下の3つの方法から選択することができる。

- 1) 境界条件として真空境界条件を与える方法(通常のコードと同じ)
- 2) 計算対象から除外される真空ノード(上図のvacuum部分)で囲む方法
- 3) 黒体を体系表面に配置する方法(上図(b)参照)

1)の方法は、ノード分割で決まる最外ノードに面した境界面に真空境界条件を与える一般的な方法である。この場合、炉心形状が円筒型であっても、問題で実質的に対象とする体系表面から境界面までの領域(上図のvacuum部分に相当)を反射体や疑似真空物質などで埋め尽くす必要がある。結果として、計算の収束が遅くなったり、疑似真空物質として与える断面積の値によっては収束しなくなることもある。

真空境界条件の与え方には、入射部分中性子流をゼロとする考え方(2.8.9式参照)と、

Milne問題の外挿定数を与える考え方(2.8.4式参照)とがある。裸の炉心でない限りこれらの差異による影響は無視できるものであるが、MOSRA-Lightでは前者を採用している。

2)の方法は、炉心計算の本質に不必要的真空ノードをあらかじめ計算対象から除外するため、効率的な計算となる(第3.4.2節参照)。真空ノードに接するノード表面では、入射部分中性子流をゼロとする扱いとなる。なお、真空ノードに漏れ出した中性子の体系への再入射は考慮されない(完全黒体の扱い)。また、真空ノードにより絶縁された外部境界条件は意味を持たない。

1)と2)の場合は、真空境界面において入射部分中性子流をゼロとするものであり、これは、外挿定数を0.5にすることを意味する(第2.8.1節参照)。MOSRA-Lightには、真空境界の外挿定数を変更したり、アルベド境界条件を与えるオプションは設定していない。これらの要求がある場合には、3)の方法を利用することになる。すなわち、体系の外表面に一層の黒体を配置し、その外挿定数を入力で調整する。例えば、真空境界面の外挿距離を変更する場合には(2.8.3)式を使用し、アルベド境界を設定するには(2.8.58)式を使用して、黒体の外挿定数を与える。

上記の何れの目的においても黒体を利用すると、プログラムにおいて黒体ノードをエネルギー群毎に識別するための判定が増え、同時に境界分離チェックボードスウェープに対して不利となるため、計算速度は黒体を使用しない場合に比べて遅くなる。特に必要がない限りは、1)または2)の方法を推奨する。

#### 4.2.12 反応率バランス

##### (1) 反応率バランスの一般式

先ず、話を簡単にするために固有値問題に限定し、(n,2n)反応の補正(第4.2.9節参照)やバックリング及び内部黒体は考慮しないものとして、反応率バランスの一般的な表現を示す。

計算が精度良く収束した場合には、体系中の各ノード、各エネルギー群(g)に対して(2.1.7)式のノードバランス式が成立する。この両辺にノード体積 $V^k = \Delta_x^k \Delta_y^k \Delta_z^k$ を乗じると、以下の群依存の反応率バランス式が得られる。

$$\langle L_g^k \rangle + \langle R_g^k \rangle = \frac{\langle P_g^k \rangle}{k_{\text{eff}}} + \langle S_{\text{in}, g}^k \rangle \quad (4.2.89)$$

ただし、

$$\langle L_g^k \rangle \equiv \langle L_{g,x+}^k \rangle + \langle L_{g,x-}^k \rangle + \langle L_{g,y+}^k \rangle + \langle L_{g,y-}^k \rangle + \langle L_{g,z+}^k \rangle + \langle L_{g,z-}^k \rangle \quad (4.2.90)$$

$$\equiv \left\{ J_{g,x+}^k \Delta_y^k \Delta_z^k \right\} + \left\{ -J_{g,x-}^k \Delta_y^k \Delta_z^k \right\} + \left\{ J_{g,y+}^k \Delta_z^k \Delta_x^k \right\} + \left\{ -J_{g,y-}^k \Delta_z^k \Delta_x^k \right\} + \left\{ J_{g,z+}^k \Delta_x^k \Delta_y^k \right\} + \left\{ -J_{g,z-}^k \Delta_x^k \Delta_y^k \right\}$$

$$\langle R_g^k \rangle \equiv \sum_{r,g}^k \phi_{g,0}^k V^k \quad (4.2.91)$$

$$\langle P_g^k \rangle \equiv \chi_g^k \sum_{g'=1}^G v \sum_{f,g'}^k \phi_{g',0}^k V^k \quad (4.2.92)$$

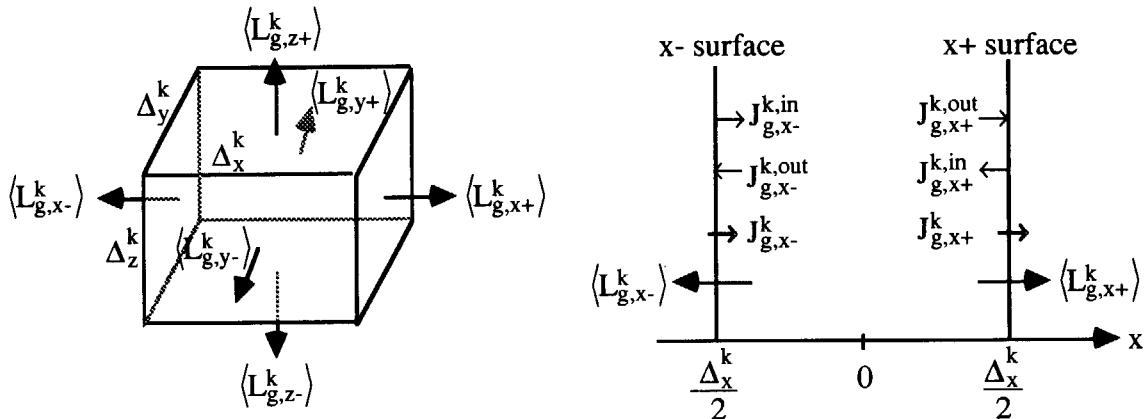
$$\langle S_{in,g}^k \rangle \equiv \sum_{\substack{g'=1 \\ g' \neq g}}^G \sum_{s,g' \rightarrow g}^k \phi_{g',0}^k V^k \quad (4.2.93)$$

なお、ブラケット記号( )は、着目領域(今の場合はk番目ノード)における体積または表面で積分された反応率(単位は個/sec)であることを明示するためのものである。(4.2.90)式は、下図に示すように6つのノード表面からの中性子の漏れ(負の場合は流入)を表面平均正味中性子流で表したものである。各項は更に、(2.1.16)式と(2.1.18)式により、表面平均部分中性子流を使って次のように表される。

$$\langle L_{g,x \pm}^k \rangle = (J_{g,x \pm}^{out,k} - J_{g,x \pm}^{in,k}) \Delta_y^k \Delta_z^k \quad (4.2.94a)$$

$$\langle L_{g,y \pm}^k \rangle = (J_{g,y \pm}^{out,k} - J_{g,y \pm}^{in,k}) \Delta_z^k \Delta_x^k \quad (4.2.94b)$$

$$\langle L_{g,z \pm}^k \rangle = (J_{g,z \pm}^{out,k} - J_{g,z \pm}^{in,k}) \Delta_x^k \Delta_y^k \quad (4.2.94c)$$



(4.2.91)式は、いわば第g群からの除去反応率を意味し、除去断面積の定義(2.1.5)式から、以下のように分解される。

$$\langle R_g^k \rangle = \langle A_g^k \rangle + \langle S_{out,g}^k \rangle \quad (4.2.95)$$

ただし、

$$\langle A_g^k \rangle \equiv \sum_{a,g}^k \phi_{g,0}^k V^k \quad (4.2.96)$$

$$\langle S_{out,g}^k \rangle \equiv \sum_{\substack{g'=1 \\ g' \neq g}}^G \sum_{s,g \rightarrow g'}^k \phi_{g,0}^k V^k \quad (4.2.97)$$

以上の式をまとめると、群依存のノード反応率は次の式を満足する。

$$\langle L_g^k \rangle + \langle A_g^k \rangle + \langle S_{out,g}^k \rangle = \frac{\langle P_g^k \rangle}{k_{eff}} + \langle S_{in,g}^k \rangle \quad (4.2.98)$$

(4.2.98)式を  $g=1 \sim G$ について総和をとれば、

$$\sum_{g=1}^G \langle S_{out,g}^k \rangle = \sum_{g=1}^G \langle S_{in,g}^k \rangle \quad (4.2.99)$$

すなわち、out-scatteringとin-scatteringの反応率は相殺し、以下のエネルギー積分された反応率のバランス式が得られる。

$$\langle L^k \rangle + \langle A^k \rangle = \frac{\langle P^k \rangle}{k_{eff}} \quad (4.2.100)$$

ただし、

$$\langle L^k \rangle \equiv \sum_{g=1}^G \langle L_g^k \rangle \equiv \langle L_{x+}^k \rangle + \langle L_{x-}^k \rangle + \langle L_{y+}^k \rangle + \langle L_{y-}^k \rangle + \langle L_{z+}^k \rangle + \langle L_z^k \rangle \quad (4.2.101)$$

$$\langle A^k \rangle \equiv \sum_{g=1}^G \langle A_g^k \rangle \quad (4.2.102)$$

$$\langle P^k \rangle \equiv \sum_{g=1}^G \langle P_g^k \rangle \quad (4.2.103)$$

次に、複数のノードから成る任意の領域をDで表し、領域D内で(4.2.100)式の総和をとる。この際、正味中性子流の保存則(これは不連続因子の有無に依存しない)により、隣接ノード間で中性子の漏洩と流入は相殺する(例えば  $\langle L_{x+}^k \rangle + \langle L_{x-}^{k+1} \rangle = 0$ )。よって、領域D内において積分された反応率バランス式は以下のようになる。

$$\langle L^{D^S} \rangle + \langle A^D \rangle = \frac{\langle P^D \rangle}{k_{eff}} \quad (4.2.104)$$

ただし、 $D^S$ は領域Dの外表面を表し、

$$\begin{aligned}
 \langle L^D \rangle &\equiv \sum_{k \in D} \langle L^k \rangle \\
 &= \sum_{k' \in D^{Sx-}} \langle L_{x+}^{k'} \rangle + \sum_{k' \in D^{Sx+}} \langle L_x^{k'} \rangle + \sum_{k' \in D^{Sy+}} \langle L_{y+}^{k'} \rangle + \sum_{k' \in D^{Sy-}} \langle L_y^{k'} \rangle + \sum_{k' \in D^{Sz+}} \langle L_{z+}^{k'} \rangle + \sum_{k' \in D^{Sz-}} \langle L_z^{k'} \rangle
 \end{aligned} \tag{4.2.105}$$

$$\langle A^D \rangle \equiv \sum_{k \in D} \langle A^k \rangle \tag{4.2.106}$$

$$\langle P^D \rangle \equiv \sum_{k \in D} \langle P^k \rangle \tag{4.2.107}$$

(4.2.105)式で、 $D^{Sx\pm}$ 、 $D^{Sy\pm}$ 、 $D^{Sz\pm}$ は、領域Dの各方向の外表面を意味し、領域内部における漏洩と流入の相殺の結果、これらの外表面( $D^S = D^{Sx+} + D^{Sx-} + D^{Sy+} + D^{Sy-} + D^{Sz+} + D^{Sz-}$ )からの全漏洩により $\langle L^D \rangle$ が決定される。

ここで、領域Dを問題の対象とする全体系(D=all)、あるいは全核分裂領域(D=fissile)とすれば、実効中性子増倍率を反応率バランスで表す見慣れた式となる。

$$k_{\text{eff}} = \frac{\langle P^{\text{all}} \rangle}{\langle L^{\text{all}} \rangle + \langle A^{\text{all}} \rangle} = \frac{\langle P^{\text{fissile}} \rangle}{\langle L^{\text{fissile}} \rangle + \langle A^{\text{fissile}} \rangle} \tag{4.2.108}$$

なお、MOSRA-Lightが output する体積平均中性子束と表面平均部分中性子流の値を使用して各種反応率や中性子漏洩を計算すると、第4.2.10節で述べた規格化法に従った値として得られる。

以上は固有値問題の場合について述べたものであるが、固定源問題(第4.2.1節参照)の場合には、これまでの式において便宜的に  $k_{\text{eff}} = 1$  とし、更に(4.2.89)式の右辺に外部源のノード平均密度  $s_{g,0}^k$  (4.2.6式参照)を加えて考えれば良い。この場合には、(4.2.104)式に対応する式として、次のバランス式が得られる。

$$\langle L^D \rangle + \langle A^D \rangle = \langle P^D \rangle + \langle S_{\text{ex}}^D \rangle \tag{4.2.109}$$

$$\text{ただし、} \langle S_{\text{ex}}^D \rangle \equiv \sum_{k \in D} \sum_{g=1}^G s_g^k V^k \tag{4.2.110}$$

以上は、反応率バランスの一般的表現であり、MOSRA-Lightコードに限定した話ではない。MOSRA-Lightの計算では、これまでに述べてきた反応率バランスの一般式が満足されないことがある。以降では、その例外となるケースについて述べる。

## (2) (n,2n)反応の補正を行う場合

SRACコードが出力する巨視的断面積ファイルには、(n,2n)反応の断面積情報が含まれている。これをMOSRA-Lightコードの入力断面積として与える場合には、第4.2.9節で述べた(n,2n)反応の補正が行われる。以下では、(n,2n)反応の補正が行われる場合の反応率バランス式を導出する。

(n,2n)反応の補正がなされた拡散方程式は(4.2.72)式で与えられる。除去断面積と散乱マトリックスに(n,2n)反応の補正が加えられるが、散乱マトリックスについては、SRACコードにより(4.2.71)式の補正済みの値が出力されるため、ユーザーがこれを意識することは無い。そこで、以下では補正された除去断面積の反応率に対してのみ注目する。

補正された拡散方程式に対して、(4.2.89)式の導出と同じ操作を行うと、以下の反応率バランス式が得られる。

$$\langle L_g^k \rangle + \langle R_g^k \rangle = \frac{\langle P_g^k \rangle}{k_{\text{eff}}} + \langle S_{\text{in},g}^k \rangle \quad (4.2.111)$$

ただし、 $\langle R_g^k \rangle$ は、(4.2.73)式に対応して、以下のように表される。

$$\begin{aligned} \langle R_g^k \rangle &\equiv \sum_{r,g}^k \phi_{g,0}^k V^k \\ &= \left\{ \sum_{a,g}^k + \sum_{\substack{g'=1 \\ g' \neq g}}^G \sum_{s,g \rightarrow g'}^k - \sum_{g'=1}^G \sum_{n2n,g \rightarrow g'}^k \right\} \phi_{g,0}^k V^k \\ &= \langle A_g^k \rangle + \langle S_{\text{out},g}^k \rangle - \langle W_g^k \rangle \end{aligned} \quad (4.2.112)$$

ただし、 $\langle W_g^k \rangle$ は以下の式で定義される第g群の(n,2n)反応率である。

$$\langle W_g^k \rangle \equiv \left\{ \sum_{g'=1}^G \sum_{n2n,g \rightarrow g'}^k \right\} \phi_{g,0}^k V^k = \sum_{n2n,g}^k \phi_{g,0}^k V^k \quad (4.2.113)$$

また、 $\langle S_{\text{in},g}^k \rangle$ と $\langle S_{\text{out},g}^k \rangle$ は、(4.2.71)式で補正された散乱マトリックスに基づく、in-scatteringとout-scatteringの反応率である。ここで、(4.2.112)式を(4.2.111)式に代入し、全エネルギー群について総和をとると、エネルギー積分された反応率バランス式を得る。

$$\langle L^k \rangle + \langle A^k \rangle - \langle W^k \rangle = \frac{\langle P^k \rangle}{k_{\text{eff}}} \quad (4.2.114)$$

$$\text{ただし、} \langle W^k \rangle \equiv \sum_{g=1}^G \langle W_g^k \rangle \quad (4.2.115)$$

(4.2.114)式は、(n,2n)反応補正をしない場合の(4.2.100)式に対応する。(4.2.114)式を任意領域Dについて総和をとれば、(4.2.104)式に対応して、

$$\langle L^{D^s} \rangle + \langle A^D \rangle - \langle W^D \rangle = \frac{\langle P^D \rangle}{k_{\text{eff}}} \quad (4.2.116)$$

を得る。更にD=all(全体系)とすれば、(n,2n)反応を考慮した場合の反応率バランスと実効中性子増倍率の関係が得られる。

$$k_{\text{eff}} = \frac{\langle P^{\text{all}} \rangle}{\langle L^{\text{all}^s} \rangle + \langle A^{\text{all}} \rangle - \langle W^{\text{all}} \rangle} \quad (4.2.117)$$

(n,2n)反応は、1つの中性子を余分に生成するため、いわば負の吸収反応の寄与をし、(n,2n)反応を考慮しない場合に比べ、実効中性子増倍率は僅かに大きくなる。

### (3) バックリングを使用する場合

1次元または2次元計算を行う場合には、計算対象としない方向への中性子の漏洩をバックリングで考慮することが多い。MOSRA-Lightでは、群依存のバックリング( $B_g^2$ )を入力で与え、 $D_{g,B}^k B_g^2$ を疑似的な吸収断面積として扱う。 $D_{g,B}^k$ はバックリング補正用の拡散係数であり、体系に与える方向依存の拡散係数からユーザーが選択する。

実際のMOSRA-Lightの計算では、反応率バランスなどの計算に頻繁に使用される吸収断面積を変更することは避け、除去断面積に $D_{g,B}^k B_g^2$ を加えている。すなわち、以下の式で定義されるバックリングを考慮した除去断面積を使用する。

$$\Sigma''_{r,g}^k \equiv \Sigma_{r,g}^k + D_{g,B}^k B_g^2 \quad (4.2.118a)$$

更に(n,2n)反応の補正を行う場合には、

$$\Sigma''_{r,g}^k \equiv \Sigma_{r,g}^k + D_{g,B}^k B_g^2 - \sum_{g'=1}^G \Sigma_{n2n,g \rightarrow g'}^k \quad (4.2.118b)$$

を使用する。結局、領域Dにおける反応率バランスの式は、これまでと同様にして以下のように与えられる。

$$\langle L^D \rangle + \langle A^D \rangle + \langle B^D \rangle - \langle W^D \rangle = \frac{\langle P^D \rangle}{k_{\text{eff}}} \quad (4.2.119)$$

$$\text{ただし、} \langle B^D \rangle \equiv \sum_{k \in D} \sum_{g=1}^G D_{g,B}^k B_g^2 \phi_{g,0}^k V^k \quad (4.2.120)$$

なお、MOSRA-Lightのプログラムでは、負のバックリング値(中性子の流入)も許容している。また、3次元計算の場合でも、入力されたバックリングは(4.2.118)式により処理される。

#### (4) 内部黒体を使用する場合

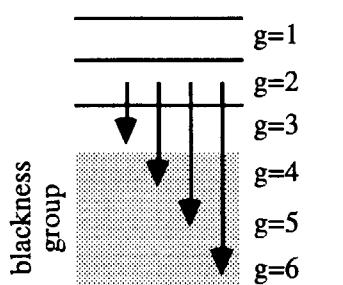
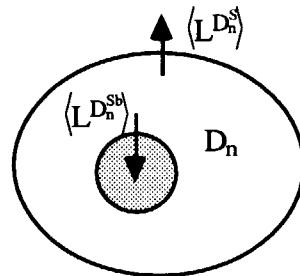
内部黒体(第2.8.5節、第4.2.11節参照)を使用する場合の反応率バランスには特別な注意が必要である。領域D内に内部黒体が含まれない場合には、これまでの反応率バランス式はそのまま利用することができる。しかし、領域Dが内部黒体を含む場合には、これまでのバランス式は破綻する。

先ず、単純な場合として、全エネルギー群について黒体の扱いをするノード( $k_b$ )が炉心内部に存在するものとする。黒体内部での反応率は定義できないが、黒体に流入する中性子は全て黒体に吸収されて消滅する。そこで、黒体ノードに隣接するノードの正味中性子流から中性子バランスを得る。右図のように、黒体を除く領域を $D_n(k_b \notin D_n)$ とし、 $D_n$ から漏れる中性子を外表面への漏れ $\langle L^{D_n} \rangle$ と、黒体表面への漏れ $\langle L^{D_n^{\text{sb}}} \rangle$ に分解すれば、次のようなバランス式となる。ただし、 $(n,2n)$ 反応の補正やバックリングは式を単純にするため考えない。

$$\langle L^{D_n} \rangle + \langle L^{D_n^{\text{sb}}} \rangle + \langle A^{D_n} \rangle = \frac{\langle P^{D_n} \rangle}{k_{\text{eff}}} \quad (4.2.121)$$

本式において、 $\langle L^{D_n^{\text{sb}}} \rangle$ が黒体に単位時間当たりに吸収される中性子数である。

問題となるのは、高エネルギー群(F)に断面積を使用し、低エネルギー群(T)だけを黒体として扱う場合である。この場合には、高エネルギー群側の断面積による反応があるため、黒体ノードを除いて反応率バランスを考えるわけにはいかない。右図は、全6群の内、第4群から第6群までを黒体扱いとする例である。この場合、例えば第2群から減速する中性子を考えると、黒体扱いするエネルギー群( $g_b \in T$ )に減速するものは、黒体に吸収されて消滅する。第g群( $g \in F$ )からの散乱により黒体エネルギー群(T)に減速する中



性子の総和は次の式で与えられ、

$$\langle S_{b,g}^{k_b} \rangle \equiv \sum_{g_b \in T} \sum_{s,g \rightarrow g_b}^{k_b} \phi_{g,0}^{k_b} V^{k_b} \quad (g \in F) \quad (4.2.122)$$

黒体ノードの高エネルギー群における反応率バランスは以下のようになる。

$$\langle L_g^{k_b} \rangle + \langle A_g^{k_b} \rangle + \langle S_{out,g}^{k_b} \rangle = \frac{\langle P_g^{k_b} \rangle}{k_{eff}} + \langle S_{in,g}^{k_b} \rangle \quad (g \in F) \quad (4.2.123)$$

ただし、

$$\langle S_{out,g}^{k_b} \rangle \equiv \sum_{\substack{g' \in F \\ g' \neq g}} \sum_{s,g \rightarrow g'}^{k_b} \phi_{g,0}^{k_b} V^{k_b} + \langle S_{b,g}^{k_b} \rangle \quad (4.2.124)$$

$$\langle S_{in,g}^{k_b} \rangle \equiv \sum_{\substack{g' \in F \\ g' \neq g}} \sum_{s,g' \rightarrow g}^{k_b} \phi_{g',0}^{k_b} V^{k_b} \quad (4.2.125)$$

更に、(4.2.123)式を高エネルギー群の範囲( $g \in F$ )で総和をとれば、(4.2.124)式の右辺第1項と(4.2.125)式の右辺は相殺し、以下の式を得る。

$$\langle L_F^{k_b} \rangle + \langle A_F^{k_b} \rangle + \langle S_{b,F}^{k_b} \rangle = \frac{\langle P_F^{k_b} \rangle}{k_{eff}} \quad (4.2.126)$$

$\langle S_{b,F}^{k_b} \rangle$ は、低エネルギー群の減速源になるわけではなく、そのまま黒体に吸収される。したがって、黒体ノードの低エネルギー群では、内部の反応率を考える必要はない。また、低エネルギー群で黒体に流入する中性子は、外側領域( $D_n$ )での漏れとして扱う。(4.2.126)式を、領域D内に含まれる全ての黒体ノードについて総和をとる( $D_b=D-D_n$ )。ただし、D内に含まれる黒体ノードの黒体境界エネルギー(Fの下限)はすべて等しいものと仮定する。

$$\langle L_F^{D_b^S} \rangle + \langle L_F^{D_b^{Sn}} \rangle + \langle A_F^{D_b} \rangle + \langle S_{b,F}^{D_b} \rangle = \frac{\langle P_F^{D_b} \rangle}{k_{eff}} \quad (4.2.127)$$

$$\text{ただし、} \langle S_{b,F}^{D_b} \rangle \equiv \sum_{k_b \in D} \langle S_{b,F}^{k_b} \rangle \text{とした。} \quad (4.2.128)$$

(4.2.127)式において、 $D_b^S$ は黒体ノードが接する領域Dの外表面を意味し、 $D_b^{Sn}$ は黒体ノードがD内の非黒体ノードに接する表面を意味する。すなわち、(4.2.127)式の第1項は、黒体ノードから外表面へ漏れ出る高エネルギー群の中性子数であり、第2項は黒体ノードから非黒体ノードへの流出である。

次に、非黒体ノード( $k_n$ )で構成される領域 $D_n$ の反応率バランスを考える。非黒体ノードに

については、全エネルギー群について、(4.2.121)式が成立するが、その第2項を高エネルギー群(F)と低エネルギー群(T)の寄与に分割する。

$$\langle L_{D_n}^{D_n^S} \rangle + \langle L_F^{D_n^S} \rangle + \langle L_T^{D_n^S} \rangle + \langle A^{D_n} \rangle = \frac{\langle P^{D_n} \rangle}{k_{\text{eff}}} \quad (4.2.129)$$

(4.1.127)式と(4.2.129)式を足し合わせたものが、領域Dに対する反応率バランスである。 $\langle L_F^{D_n^S} \rangle + \langle L_T^{D_n^S} \rangle = 0$ の関係を使用すれば、以下の式を得る。

$$\langle \langle L_F^{D_n^S} \rangle + \langle L_T^{D_n^S} \rangle \rangle + \langle \langle A^{D_n} \rangle + \langle A_F^{D_n} \rangle \rangle + \langle \langle L_T^{D_n^S} \rangle + \langle S_{b,F}^{D_n} \rangle \rangle = \frac{\langle P_F^{D_n} \rangle + \langle P_T^{D_n} \rangle}{k_{\text{eff}}} \quad (4.2.130)$$

本式の左辺第1項は領域Dの外表面への中性子の漏れ、第2項はD内での吸収、第3項は黒体への流入と減速による消滅、右辺は核分裂による生成である。なお、D内に複数の黒体ノードが含まれ、これらの黒体境界エネルギーが不揃いな場合には、更に複雑な式となるが、ここでは省略する。

(n,2n)反応の補正やバッククリングを考慮している場合には、(4.2.130)式は以下のようになる。

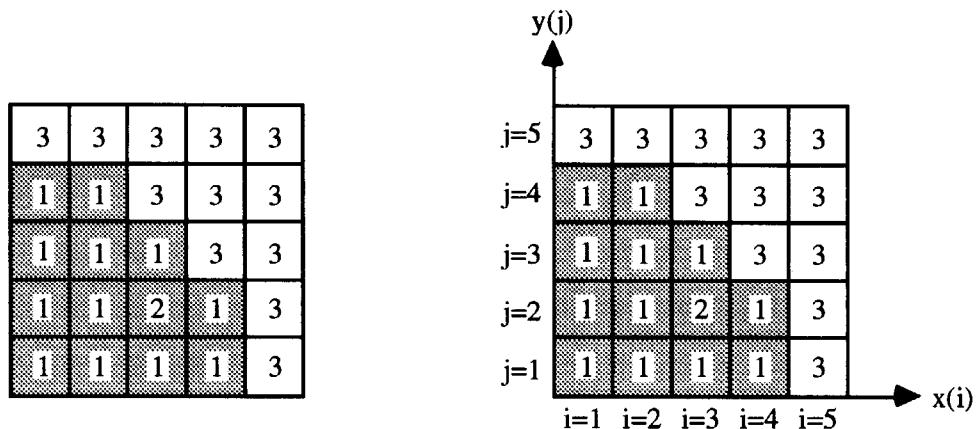
$$\begin{aligned} & \langle \langle L_F^{D_n^S} \rangle + \langle L_T^{D_n^S} \rangle \rangle + \langle \langle A^{D_n} \rangle + \langle A_F^{D_n} \rangle \rangle + \langle \langle B^{D_n} \rangle + \langle B_F^{D_n} \rangle \rangle - \langle \langle W^{D_n} \rangle + \langle W_F^{D_n} \rangle \rangle + \langle \langle L_T^{D_n^S} \rangle + \langle S_{b,F}^{D_n} \rangle \rangle \\ &= \frac{\langle P_F^{D_n} \rangle + \langle P_T^{D_n} \rangle}{k_{\text{eff}}} \end{aligned} \quad (4.2.131)$$

(4.2.130)式に比べて、左辺に $\langle \langle B^{D_n} \rangle + \langle B_F^{D_n} \rangle \rangle - \langle \langle W^{D_n} \rangle + \langle W_F^{D_n} \rangle \rangle$ が加わった他、高エネルギー群の(n,2n)反応により黒体に遷移して消滅する中性子は、(4.2.71)式により補正される散乱マトリックスを使用する $\langle S_{b,F}^{D_n} \rangle$ の中に加味されている。すなわち、 $\langle S_{b,F}^{D_n} \rangle$ は、以下の式で定義される。

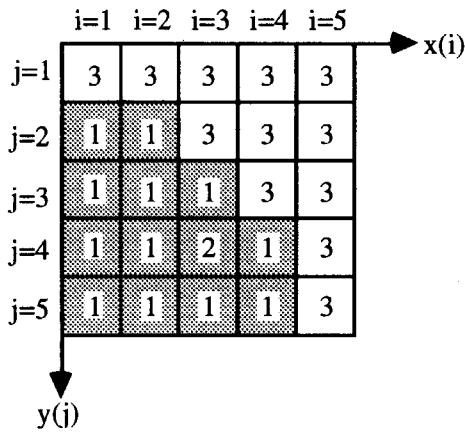
$$\langle S_{b,F}^{D_n} \rangle \equiv \sum_{k_b \in D} \langle S_{b,F}^{k_b} \rangle \equiv \sum_{k_b \in D} \sum_{g \in F} \langle S_{b,g}^{k_b} \rangle \equiv \sum_{k_b \in D} \sum_{g \in F} \sum_{g_b \in T} \left\{ \left( \sum_{s,g \rightarrow g_b}^{k_b} + 2 \sum_{n2n,g \rightarrow g_b}^{k_b} \right) \phi_{g,0}^{k_b} V^{k_b} \right\} \quad (4.2.132)$$

#### 4.2.13 入出力座標系の選択

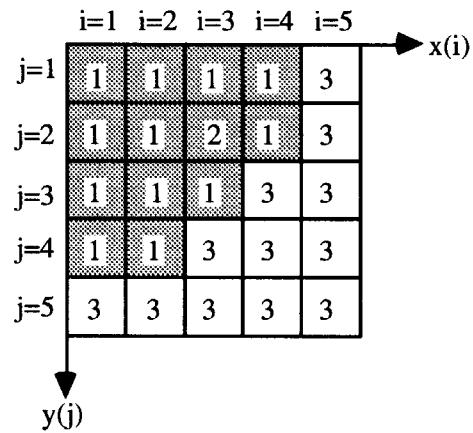
MOSRA-Lightの標準入出力において、水平方向(x-y)の座標系を選択することができる。例えば、下図(a)のような5×5ノードに分割された反射体付き1/4炉心の2次元体系の計算を考える。この時、MOSRA-Lightでは図(b)のような標準座標系と図(c)のような上下を反転した座標系が選択可能である。

(a) 3領域1/4炉心モデル  
(1-2: 燃料、3: 反射体)

(b) 標準座標系



(c) 反転座標系



(d) 反転炉心+反転座標系

反転座標系は、y軸方向をコンピュータ端末の画面スクロールやプリンター出力の方向に合わせたもので、CITATIONなどこの座標系を使用するコードが多い。反転座標系が使用されるのはコンピュータの入出力装置やプログラムの都合であり、図(a)のような問題では標準座標系(b)を使用する方が実用的である。なぜなら、反射体の厚さは必ずしも実寸である必要はなく、図(c)のように反射体を原点とするy座標を使用すると、炉心ノードの位置(x, y)または座標番号(i, j)が反射体部分の厚さやノード分割に依存してしまい、特に大きな体系では、出力結果の編集が分かりにくいものとなる。反転座標系しか利用できないコードを使用する場合には、原点を炉心中に合わせるために、わざわざ図(d)のように炉心も上下反転させたモデルで計算することも多い。しかし、この場合には物質番号(断面積セット)を分割ノードに割り当てる際に混乱を生じやすく、物質構成が複雑な場合に入力ミスをおかす原因となる。また、出力分布などの文献値は、ほとんど標準座標系でまとめられており、標準座標系で計算をすると文献値との比較が容易である。一方、反転座標系を採用するコードと計算結果を比較する場合には、これに座標系を合わす方が便利であろう。座標系の選択は、MOSRA-Lightにとって、入力読み込みとプリント出力の処理の違いだけであり、計算の本質部分の処理は全く同じである。従って、問題や目的に応じてどちらの座標系を選択してもよい。

択するか決めると良い。

入力データの作成において、物質番号を分割ノードに割り当てる際には、どちらの座標系を使用しても、体系図を見たとおりに物質番号を入力するようにしている。例えば、上の図(b)と図(c)の例では、物質番号のマップを以下のように入力する。

3	3	3	3	3	(j=5)
1	1	3	3	3	(j=4)
1	1	1	3	3	(j=3)
1	1	2	1	3	(j=2)
1	1	2	1	3	(j=1)

[標準座標系(b)の物質番号入力]

3	3	3	3	3	(j=1)
1	1	3	3	3	(j=2)
1	1	1	3	3	(j=3)
1	1	2	1	3	(j=4)
1	1	2	1	3	(j=5)

[反転座標系(c)の物質番号入力]

物質番号の水平方向入力マップは同じであるため、入力データを大幅に変更することなく、使用する座標系の変更が可能である。ただし、y軸方向境界条件の入力データを逆転する必要がある(y+側とy-側が入れ替わるため)。

計算結果のプリント形式は、入力マップに倣った並びとして表示される。例えば、出力分布や中性子束分布などは、標準座標系(b)の場合には以下のようないmageでプリントされる(炉心部の数値は便宜的に物質番号とした)。

	1	2	3	4	5
5	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
4	<b>1.00000E+00</b>	<b>1.00000E+00</b>	0.00000E+00	0.00000E+00	0.00000E+00
3	<b>1.00000E+00</b>	<b>1.00000E+00</b>	<b>1.00000E+00</b>	0.00000E+00	0.00000E+00
2	<b>1.00000E+00</b>	<b>1.00000E+00</b>	<b>2.00000E+00</b>	<b>1.00000E+00</b>	0.00000E+00
1	<b>1.00000E+00</b>	<b>1.00000E+00</b>	<b>1.00000E+00</b>	<b>1.00000E+00</b>	0.00000E+00

一方、反転座標系(c)の場合には、同じ計算結果が以下のように表示される。

	1	2	3	4	5
1	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00
2	<b>1.00000E+00</b>	<b>1.00000E+00</b>	0.00000E+00	0.00000E+00	0.00000E+00
3	<b>1.00000E+00</b>	<b>1.00000E+00</b>	<b>1.00000E+00</b>	0.00000E+00	0.00000E+00
4	<b>1.00000E+00</b>	<b>1.00000E+00</b>	<b>2.00000E+00</b>	<b>1.00000E+00</b>	0.00000E+00
5	<b>1.00000E+00</b>	<b>1.00000E+00</b>	<b>1.00000E+00</b>	<b>1.00000E+00</b>	0.00000E+00

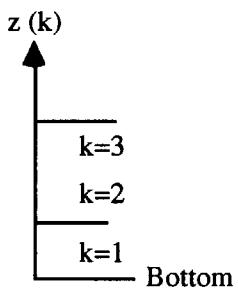
また、炉心も反転した反転座標系(d)の場合には、以下のように表示される。

	1	2	3	4	5
1	<b>1.00000E+00</b>	<b>1.00000E+00</b>	<b>1.00000E+00</b>	<b>1.00000E+00</b>	0.00000E+00
2	<b>1.00000E+00</b>	<b>1.00000E+00</b>	<b>2.00000E+00</b>	<b>1.00000E+00</b>	0.00000E+00
3	<b>1.00000E+00</b>	<b>1.00000E+00</b>	<b>1.00000E+00</b>	0.00000E+00	0.00000E+00
4	<b>1.00000E+00</b>	<b>1.00000E+00</b>	0.00000E+00	0.00000E+00	0.00000E+00
5	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00

z軸方向については、水平方向の場合のような不便は無いため、入出力座標系の選択オプ

ションは設定していない。プリント出力上では、重力方向の体系境界面をBottomと表示するようにしているが、むろん中性子の輸送は重力には関係ないため、Bottomをどの方向と考えるかはユーザー次第である。なお、入出力マップはz軸方向のノード番号K=1, 2, …の順に従う。

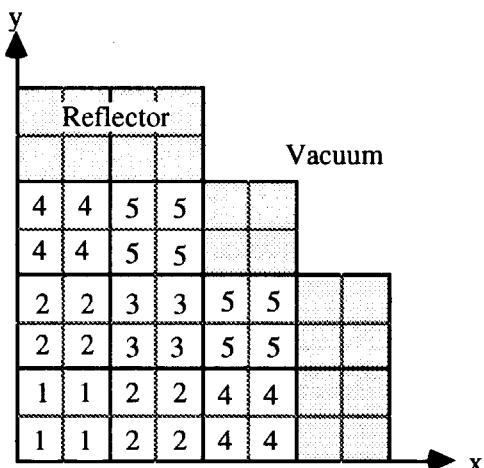
実際の炉心体系をコードのx-y-z座標系にどのように配置させるかは、ユーザーの自由である。しかし、90度回転境界条件(第2.8.4節参照)と対角対称境界条件(第2.8.6節参照)を使用する場合には、プログラムの制限上、x軸とy軸がなす面上に回転面や対角線がくるように、炉心を配置する必要がある。



#### 4.2.14 編集領域(Edit-Region)の利用

##### (1) Edit-Regionの機能

MOSRA-Lightは、中性子束と出力密度を、ノード単位の他、ユーザーが自由に指定する領域の平均値としてプリント出力することができる。MOSRA-Lightでは、このような領域を編集領域(Edit-Region)と名付けている。例えば下図(a)のように、8つの燃料集合体とこれを囲む反射体から成る1/4炉心モデルを考える。また、各燃料集合体は水平方向に4ノード分割(2×2)されているものとする。



(a) Calculation model and Edit-Region

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
4	4	5	5	0	0	0	0	0
4	4	5	5	0	0	0	0	0
2	2	3	3	5	5	0	0	0
2	2	3	3	5	5	0	0	0
1	1	2	2	4	4	0	0	0
1	1	2	2	4	4	0	0	0

(b) Sample Input Data for Edit-Region

燃料集合体単位で中性子束や出力分布などをプリントさせたい場合には、Edit-Region番号を図(a)のように割り当てれば良い。Edit-Regionの番号付けは、飛び番号が無い限りユーザーの自由である。この場合に対応するEdit-Regionの入力データは、図(b)のようになる。計算から除外される真空ノード(第4.2.11節参照)や編集を不要とする反射体領域などには、Edit-

Region番号にゼロを指定する。3次元計算の場合には、z軸方向のノード分割順(K=1,2,3,...)に図(b)のような入力データを繰り返せばよい。

Edit-Regionの形は任意であり、必ずしも1つの閉じた領域である必要は無く、体系中に離れて存在していても良い。例えば上部ブランケットと下部ブランケットの平均中性子束や平均出力密度をプリントさせる場合などである。ただし、Edit-Regionの指定は、1つの計算ケースに対して1回しか許されず、重なる複数の領域を一度に指定することはできない。このような場合には、同じ体系で複数の計算ケースを実行させる必要がある。MOSRA-Lightでは、複数の計算ケースを1回のジョブとして計算できる(第5.2節参照)。最初の計算ケース以降では、初期値参照コールドスタート計算またはリスタート計算のオプション(第4.2.6節参照)を使用すれば計算時間はほとんど掛からない。

中性子束と出力密度のEdit-Region平均値の計算は、以下の式に従って行われる。

$$\Phi_{g,0}^I = \frac{\sum_{k \in I} \Phi_{g,0}^k V^k}{\sum_{k \in I} V^k} \quad (4.2.133)$$

$$q'''^I = \frac{\sum_{k \in I} q'''^k V^k}{\sum_{k \in I} V^k} \quad (4.2.134)$$

$\Phi_{g,0}^I$  と  $q'''^I$  は、I番目のEdit-Regionにおける平均中性子束と出力密度を意味する。 $\Phi_{g,0}^k$  と  $q'''^k$  は、それぞれ、(4.2.76)式と(4.2.78)式で定義され、 $\Phi_{g,0}^I$  と  $q'''^I$  の数値は中性子束の規格化方法(第4.2.10節参照)により異なる。

内部黒体(第4.2.11節参照)を使用する場合には注意を要する。内部黒体ノードにおいて、断面積が使用されるエネルギー群については中性子束が計算されるが、境界条件の扱いとなるエネルギー群については中性子束は定義されない(コードでは便宜的にゼロ値とする)。したがって、内部黒体ノードをEdit-Regionに含めた場合には、そのEdit-Regionに対する  $\Phi_{g,0}^I$  ( $g$  : 境界条件の扱いとなるエネルギー群)と出力密度  $q'''^I$  は正しく得られる保証はない。

## (2) プリント出力形式

Edit-Regionに対する中性子束と出力密度の情報は、プリント出力された後には失われ、MOSRA-Lightが outputするバイナリーファイル(第7章参照)にはノード毎の値だけが収納される。バイナリーファイルからデータを読みとて自由に再編集をすることは可能であるが、MOSRA-Lightではプリント出力されるEdit-Region平均値をユーザーが容易に加工できるようにしている。

以下は、MOSRA-Lightがプリント出力するEdit-Regionに対する出力密度の表示例である。

```
=====
EDIT-REGION POWER DENSITY (UNIT: RELATIVE )
=====

* AVERAGED POWER DENSITY = 1.00000E+00
* FISSILE VOLUME (CM3)    = 3.36600E+06

1      7.35709E-01
2      1.28913E+00
3      1.42976E+00
4      1.20073E+00
:
:
```

Edit-Regionの番号と出力密度の値は、スペースの無駄を承知で縦に並べられる。最近の計算機端末(PCやEWS)では、ディスプレイのテキスト情報を記憶バッファに複写して表計算ソフトなどのワークシートに貼り付ける機能(copy & paste)が備わっている。単純な縦並びとしているのは、市販の表計算ソフトや描画ソフトにEdit-Regionのデータを取り込み、データの再編集やプロットを直接できるようにするための配慮である。Edit-Regionの番号付けは任意ではあるが、目的に応じた順番に設定しておくと、出力分布をプロットしたり、他コードの計算結果と比較する場合などに便利である。

一方、中性子束の場合には、オプションにより以下に示すような2通りの表示方法が選択できる。

```
=====
FORWARD FLUX IN EDIT-REGION (UNIT: RELATIVE )
=====

* FLUX NORMALIZATION : (TOTAL POWER = FISSILE VOLUME)

E-RGN      1-G          2-G
1      6.40458E+01  1.08859E+01
2      8.21421E+01  1.90752E+01
3      8.98553E+01  2.11575E+01
4      7.64257E+01  1.77704E+01
:
:
```

```
=====
FORWARD FLUX IN EDIT-REGION (UNIT: RELATIVE )
=====

* FLUX NORMALIZATION : (TOTAL POWER = FISSILE VOLUME)

GROUP      1-R          2-R          3-R      ...
1      6.41248E+01  8.22418E+01  8.99577E+01  ...
2      1.08994E+01  1.90983E+01  2.11816E+01  ...
:
:
```

前者は空間分布の編集を念頭に置いた表示法であり、後者は特に詳細群計算における中性子スペクトルの編集を念頭においていた表示法である。例えば、後者の数値を、表計算ソフト

などを使用し、レザージ幅で割ってプロットすれば、Edit-Region毎の中性子スペクトルを容易に比較することができる。

#### 4.2.15 ユーザー編集ルーチン

MOSRA-Lightは、モジュラーコードとしての性格上、多くの編集機能は持ちあわせていない。複雑な処理を要する編集については、MOSRA-Lightが outputするバイナリーファイルを利用するユーティリティ(第4.2.16節参照)に任せている。しかし、多大なルーチンワークをするような場合には、MOSRA-Lightの計算の中でユーザーが要求するデータを自由に編集処理できたほうが便利である。そこで、多少ともFORTRAN言語によるプログラミングができるユーザーが、MOSRA-Lightの計算結果を加工して出力できるようにするために、ユーザー作成用のサブルーチン(USERED)を提供している。

ユーザー編集ルーチンは、1つの計算ケースに対しMOSRA-Lightが行う一連の計算処理が全て終了した時点において、オプション指定により呼び出される。このユーザー編集ルーチンの内容は以下のようなもので、変数の意味は内部にコメントで記してある。

```

SUBROUTINE USERED( NOUT2,
1      NG, NGT, NMAT, NX, NY, NZ, NOD, IOPT, IBCD, IFG,
2      IX, IY, IZ, NXYZ, XXX, YYY, ZZZ, DELX, DELY, DELZ,
3      VOLN, IDMAT, IERGN, IHLK, XJFIS, BSQ, EXTZO,
4      AKEFF, FLXA0, POWDT,
5      CINXP, CINXN, CINYP, CINYN, CINZP, CINZN,
6      COTXP, COTXN, COTYP, COTYN, COTZP, COTZN,
7      SIGA, SIGR, SIGF, SIGP, SIGX, SIGDX,
8      SIGDY, SIGDZ, SIGS )
C
C*****SUBROUTINE USERED : EDITOR WHICH WILL BE PROGRAMED BY USER*****
C
C NOUT2      : PRINT OUT DEVICE NUMBER
C NG         : NUMBER OF TOTAL GROUP
C NGT        : NUMBER OF THERMAL GROUP
C NMAT       : NUMBER OF MATERIAL
C NX         : MAX NUMBER OF NODE IN X-DIRECTION
C NY         : MAX NUMBER OF NODE IN Y-DIRECTION
C NZ         : MAX NUMBER OF NODE IN Z-DIRECTION
C NOD        : TOTAL NUMBER OF EFFECTIVE NODE
C IOPT(1..20) : FLAG FOR GENERAL CONTROL OPTION (SEE INPUT DATA)
C IBCD(1..7)  : FLAG FOR BOUNDARY CONDITION (SEE INPUT DATA)
C IFG         : CURRENT CALCULATION TYPE (=0:FORWARD, =1:ADJOINT)
C IX(L)       : X-POSITION OF THE L-TH NODE (LEFT->RIGHT)
C IY(L)       : Y-POSITION OF THE L-TH NODE (BACK->FRONT)
C IZ(L)       : Z-POSITION OF THE L-TH NODE (BOTTOM->TOP)
C NXYZ(I,J,K) : SEQUENTIAL NODE NUMBER OF THE NODE POSITION (I,J,K)
C                 NXYZ(IX(L),IY(L),IZ(L)) = L
C XXX(1..NX+1) : X-DISTANCE(CM), XXX(1) = 0.0

```

```

C YYY(1..NY+1) : Y-DISTANCE(CM), YYY(1) = 0.0
C ZZZ(1..NZ+1) : Z-DISTANCE(CM), ZZZ(1) = 0.0
C DELX(L)      : X-WIDTH OF THE L-TH NODE
C DELY(L)      : Y-WIDTH OF THE L-TH NODE
C DELZ(L)      : Z-WIDTH OF THE L-TH NODE
C VOLN(L)      : VOLUME(CM3) OF THE L-TH NODE
C IDMAT(I,J,K) : MATERIAL ID NUMBER
C                 IDMAT=0:VACCUME NODE, IDMAT<0:BLACKNESS
C IERGN(I,J,K) : EDIT-REGION NUMBER
C IHLK(L)      : LIST VECTOR ON GEOMETRICAL CONDITION OF THE L-TH NODE
C                 6-DIJITS (IJKLMN, EX. 102020)
C                 I : BOUNDARY CONDITION ON Z(-) SIDE
C                     =0 NODE EXISTING
C                     =1 VACCUME
C                     =2 REFLECTIVE
C                     =3 PERIODIC
C                     =4 90-DEG ROTATIONAL
C                 J : BOUNDARY CONDITION ON Z(+) SIDE
C                 K : BOUNDARY CONDITION ON Y(-) SIDE
C                 L : BOUNDARY CONDITION ON Y(+) SIDE
C                 M : BOUNDARY CONDITION ON X(-) SIDE
C                 N : BOUNDARY CONDITION ON X(+) SIDE
C
C XJFIS(M)     : J/FISSION VALUE OF THE M-TH MATERIAL
C BSQ(IG)       : BUCKLING VALUE (1/CM2) BY GROUP
C EXTSO(L,IG)   : EXTRRA SOURCE ONLY WHEN FIXED SOURCE PROBLEM
C AKEFF         : EIGENVALUE(K-EFF) (1.0 WHEN FIXED SOURCE PROBLEM)
C FLXA0(L,IG)   : NODAL AVERAGED FLUX
C POWDT(L)      : NODAL AVERAGED POWER DENSITY (W/CM3)
C CINXP(L,IG)   : PARTIAL IN-CURRENT AT X(+) SURFACE
C CINXN(L,IG)   : PARTIAL IN-CURRENT AT X(-) SURFACE
C CINYP(L,IG)   : PARTIAL IN-CURRENT AT Y(+) SURFACE
C CINYN(L,IG)   : PARTIAL IN-CURRENT AT Y(-) SURFACE
C CINZP(L,IG)   : PARTIAL IN-CURRENT AT Z(+) SURFACE
C CINZN(L,IG)   : PARTIAL IN-CURRENT AT Z(-) SURFACE
C
C COTXP(L,IG)   : PARTIAL OUT-CURRENT AT X(+) SURFACE
C COTXN(L,IG)   : PARTIAL OUT-CURRENT AT X(-) SURFACE
C COTYP(L,IG)   : PARTIAL OUT-CURRENT AT Y(+) SURFACE
C COTYN(L,IG)   : PARTIAL OUT-CURRENT AT Y(-) SURFACE
C COTZP(L,IG)   : PARTIAL OUT-CURRENT AT Z(+) SURFACE
C COTZN(L,IG)   : PARTIAL OUT-CURRENT AT Z(-) SURFACE
C
C ****
C
      REAL*8  DELX, DELY, DELZ, VOLN, AKEFF, FLXA0,
      &        FLXX1, FLXY1, FLXZ1, FLXX2, FLXY2, FLXZ2,
      &        CINXP, CINYP, CINZP, CINXN, CINYN, CINZN,
      &        COTXP, COTYP, COTZP, COTXN, COTYN, COTZN
C
      DIMENSION IOPT(20), IBCD(7), IX(NOD), IY(NOD), IZ(NOD),
      1          NXYZ(NX,NY,NZ), XXX(NX+1), YYY(NY+1), ZZZ(NZ+1),
      2          DELX(NOD), DELY(NOD), DELZ(NOD), VOLN(NOD),
      3          IDMAT(NX,NY,NZ), IERGN(NX,NY,NZ), IHLK(NOD),
      4          XJFIS(NMAT), BSQ(NG), EXTSO(NOD,NG), FLXA0(NOD,NG),
      5          POWDT(NOD), CINXP(NOD,NG), CINYP(NOD,NG), CINZP(NOD,NG),
      6          CINXN(NOD,NG), CINYN(NOD,NG), CINZN(NOD,NG),

```

ユーザーは、 "%"で囲まれた範囲に必要なプログラムを書き込み、ロードモジュールを作成する。例えば、ノード位置(I=1, J=2, K=3)における第1群中性子束と出力密度をプリントさせるには、

```
L = NXXYZ(1,2,3)
IG = 1
WRITE (IOUT2,*) FLXA0(L,IG), POWDT(L)
```

とすれば良い。プログラム作成上のルールとしては、プログラムに用意された変数を代入文で再定義してはならない。ユーザーが必要な配列は、DIMENSION文でローカルに定義すること(COMMON文で定義しない)である。

#### 4.2.16 ユーティリティプログラム (More-MOSRA)

MOSRA-Lightは、計算が終了すると、幾何形状情報、中性子束、部分中性子流、出力分布、断面積などの基本的な情報をファイルに出力する。これらを利用してMOSRA-Light本体が持たない多くの機能をユーティリティプログラム(More-MOSRA)がサポートしている。ユーティリティプログラムは、以下のような機能を提供する。

- 1) 一点炉動特性パラメータ(実効遅発中性子割合, 即発中性子寿命, 先行核崩壊定数)の計算  
(第7.4節の(4)参照)
  - 2) 一次摂動と厳密摂動計算
  - 3) 反応率計算(指定領域の各種反応率、漏洩、検出器反応率)
  - 4) 群縮約と均質化(指定領域における中性子束と断面積)
  - 5) ノード内中性子束分布の計算<sup>31),32)</sup>

なお、これらの計算手法の詳細や利用方法については、別途報告する予定である。

## 5. MOSRA-Lightの入力

### 5.1 自由形式入力

MOSRA-Lightのデータ入力は、文字データの入力を除き、カラム指定の無い自由形式(フリーフォーマット)で行う。以下にその特徴と使用上のルールを述べる。なお、この入力形式はSRACコード<sup>25)</sup>のものと共通である。

- (1) 文字データの入力を除き、カラム指定は無い。入力指定(第5.3節参照)で要求される必要なデータ数を任意のカラムに入力する。ただし、カラム1～カラム72をデータ入力領域とする。データ数が多い場合には、次の行に入力する。

例：10個の整数を入力する場合

```
(column) -----1-----2-----3-----4--- ... ---7--
      1 2     3   4           5
      6   7     8   9       10
```

- (2) 1ワード(1つの整数または実数データ)は、原則として1つ以上のブランクにより区別する。ただし、ブランクをタブで代用してはならない。

例1：5個の実数を入力する場合

```
-----1-----2-----3-----4--- ...
      1.1   2.22    3.333   4.4444        5.55555
```

また、他コードなどの出力結果を利用できるように、1つのカンマ','、符号'+'、'-'により区別することも可能である。ただし符号'±'は、データの区切りと同時にデータの正負も意味する。

例2：5つの整数值(1, 2, 3, 4, -5)を入力する場合

```
-----1-----2-----3-----4--- ...
      1,2   3+4-5
```

例3：E12.5のフォーマットで書かれた5つの実数値(0.01, 0.2, 3.0, -40., 500.)を入力する。

```
-----1-----2-----3-----4-----5-----6 ...
      0.10000E-01 0.20000E+00 0.30000E+01-0.40000E+02-0.50000E+03
```

正負の符号と数字の間にブランクを入れてはならない。

誤例：3つの整数(1, -2, 3)を入力する場合

```
-----1-----2-----3-----4--- ...
      1 -   2   +   3
```

- (3) 実数データの入力は、基本実定数形式(Fタイプ：x.xxxx)、または単精度指数形式(Eタイプ：x.xxxExx)で行う。指数部の' E 'の代わりに小文字の' e 'を使用することは許される。ただし、' E 'の後ろにブランクを挿入してはならない。また、倍精度指数形式(Dタイプ：x.xxxDxx)は許されない。

例：FタイプとEタイプの実数を入力する場合

```
-----1-----2-----3-----4--- ...
-12.543 0.00125 1.0E-4 -.4E12 2.0e-2
```

誤例：2つの実数(1.0E-5と1.0D-4)を入力する場合

```
-----1-----2-----3-----4--- ...
1.000E -5 1.0D-4
```

- (4) 1つのデータ(ワード)を、行をまたがって入力してはならない。

誤例：実数1.0E-4を入力する場合

```
-----1-----2--- ... --6-----7--
1.0E
-4
```

- (5) 1つの入力ブロックに、3つのタイプ(文字列、整数、実数)のデータを並べて入力することができます。ただし、入力の順序は、文字列、整数、実数の順である。入力ブロックとは、1セットの複数データの並びであり、各タイプの変数をブロックに幾つ入力するかは入力ブロック毎に指示される利用マニュアルの記述(第5.3節参照)に従う。

例1：4文字(A4)、5つの整数、4つの実数が要求される入力ブロックの場合

```
-----1-----2-----3-----4-----5--- ...
ABCD 1 2 3 4 5 1.111 2.222 3.333 4.444E1
```

例2：8文字(A8)と4つの実数が要求される入力ブロックの場合

```
-----1-----2-----3-----4-----5--- ...
MOSRA-lt 1.111 2.222 3.333 4.444E1
```

文字型データは、原則として1カラム目から指定された文字数だけ入力する。特殊記号(ブランク、カンマ、\$, %, #など)の使用も許されるが、MOSRA-Lightでは入力された文字列は入出力ファイル名として参照されるため、利用する計算機システムでファイル名として許容されないような文字は避ける。なお、文字型データに対しては、特にことわりがない限り、上記及び下記の自由形式入力は適用できない。

- (6) 繰り返し関数' ( )'による整数および実数データの入力が可能である。繰り返し関数は、 $i( )$ の形式で使用する。 $i$ は整数值で、( )内に記述されたデータを $i$ 回繰り返し入力することを意味する。 $i$ と' ( )'の間には複数のブランクがあつても良いし無くても良い。

例1：データ' 1 2 2 2 1.0E-4 1.0E-4 'を繰り返し関数を使用して入力する場合

```
-----1-----2-----3-----4--- ...
1 3(2) 2 (1.0E-4)
```

繰り返し開始記号' ( 'と終了記号' )'は、必ずしも同じ入力行にある必要はない。ただし、入力行の始めに' ( 'または' )'を配置してはならない。また、多重の' ( )'は許されない。

例2：水平方向炉心メッシュに物質番号(1:炉心, 2:反射体)を割り当て、これを軸方向のメッシュ分割数(例えば10)だけ繰り返すような場合。

```
-----1-----2-----3-----4--- ...
10 ( 2 2 2 2 2 2 2 2 2
      2 2 2 1 1 1 2 2 2
      2 2 1 1 1 1 1 2 2
      2 2 1 1 1 1 1 2 2
      2 2 2 1 1 1 2 2 2
      2 2 2 2 2 2 2 2 2 )
```

誤例1：繰り返し終了記号' )' を入力行の始めに配置

```
-----1-----2-----3-----4--- ...
2 (1 2 3 4
      ) 5 6 7 8
```

誤例2：繰り返し関数を多重で使用

```
-----1-----2-----3-----4--- ...
10 ( 2(2 1) 2(3 1) 2(4 1) )
```

- (7) 等差数列関数'\*'による整数および実数データの入力が可能である。等差数列関数は、 $i*b$ の形式で使用する。 $i*b$ は、1つ前のデータを $a$ とするとき、 $a+b$ を初項とし増分 $b$ の等差数列による $i$ 個のデータ( $a+b, a+2b, a+3b, \dots, a+i*b$ )を入力することを意味する。 $i$ は常に整数でなければならないが、 $a$ と $b$ は同じデータタイプであれば、整数でも実数でも良い。

例1：1から10までの整数值を等差数列関数を使用して入力する場合

```
-----1-----2-----3-----4--- ...
1 1*9
```

例2：データ'0.0 1.0 2.0 3.0 4.0 2.0 0.0'を等差数列関数を使用して入力する場合

```
-----1-----2-----3-----4--- ...
0.0 4*1.0 2*-2.0
```

例3：0.0から20.0までを15等分割する16個のデータを入力する場合

```
-----1-----2-----3-----4--- ...
0.0 14*1.333333 20.0
```

上記のような例で、境界値(20.0)が計算結果に少なからず影響を及ぼすと考えられる場合には、等差数列関数の数値誤差を避けるために、境界値を別に明記する。

なお、等差数列関数を使用する場合には、初項を省略したり、繰り返し関数' ()'と共に用することは許されない。

誤例1：等差数列関数が参照する初項が無い

```
-----1-----2-----3-----4--- ...
1*10
```

誤例2：繰り返し関数' ()'との共用

```
-----1-----2-----3-----4--- ...
10 (1 1*9)
```

- (8) 記号' / 'は、各入力ブロックにおける全データの入力終了時に、以降のカラムをコメン

ト挿入に使用するためのものである。コメントを挿入しない場合には、' /'を入力する必要はない。

例：10個の整数型データが要求される入力ブロックにコメントを挿入する場合

```
-----1-----2-----3-----4--- ...
5(0) 5(1) / Input Data for Block-1
```

(9) 入力ブロックにおけるデータ入力の途中で、以降のカラムをコメント挿入に使用する場合には、記号' &'を使用する。また、入力ブロックとは関係なく、' &'から始まる行をコメント行とすることができます。

例：10個の整数型データが要求される入力ブロックと、これに続く5個の整数型データが要求されるブロックにコメントを挿入する場合

```
-----1-----2-----3-----4-----5--- ...
1 2 3 4 5 & Input Data for Block-1 (1-5)
& Comment-1
& Comment-2
6 7 8 9 10 / End of Block-1
& Start of Block-2
1 2 3 4 5
```

MOSRA-Lightでは、入力ブロックで要求されるデータ数やタイプが実際に入力されたデータと比べて不整合であることを検出すると、その原因と検出された入力データの位置をエラーメッセージとして出力する。しかし、次の入力ブロックのデータによっては、見かけ上不整合と見なされないことがある。このような場合にも、最終的に入力データの不整合は発生するが、エラーメッセージの内容が実際の原因や不適切なデータの位置と異なることがある。メッセージの内容に疑問がある場合には、不整合が検出された入力ブロックの前のブロックまでさかのぼってデータを確認することを勧める。

## 5.2 入力データの構成

MOSRA-Lightでは、幾何形状や断面積などが異なる複数の計算ケースを連続して実行することができる。入力データは、下図に示すように、全体の計算を制御するための入力セクション(Section-1)と、各計算ケース(CASE-1, CASE-2, ...)毎に与える入力セクション(Section-2～Section-11)により構成される。

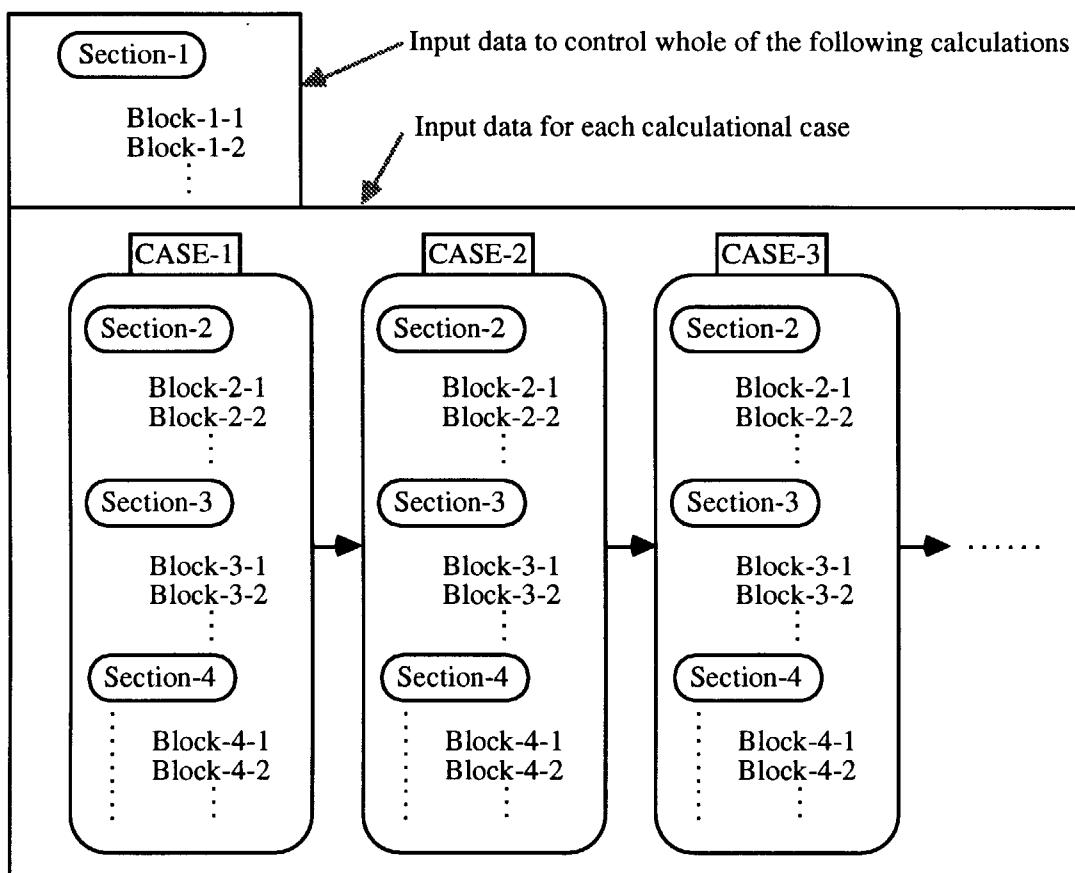


Fig. 5.2.1 Structure of input data for the MOSRA-Light code

MOSRA-Lightの入力は、以下に示す11個の入力セクションにより構成される。

- 1) 全体計算制御 (常時必要)
- 2) ケース毎の計算制御 (常時必要)
- 3) 幾何形状入力 (常時必要)
- 4) 物質指定 (常時必要)
- 5) 初期値参照ケースの指定 (オプションにより必要)

- 6) 断面積値入力(オプションにより必要)
- 7) 編集領域(Edit-Region)の指定(オプションにより必要)
- 8) バックリング入力(オプションにより必要)
- 9) 黒体の入力(オプションにより必要)
- 10) 外部源の入力(オプションにより必要)
- 11) 不連続因子の入力(オプションにより必要)

入力データの作成では、1)の入力セクションに続き、計算ケース毎に2)～11)の入力セクションのうち必要な部分を繰り返し入力する。各入力セクションは、更に複数の入力ブロックで構成される。入力ブロックで要求されるデータの並びを、所定の自由形式入力(第5.1節参照)に従って記述する。

各入力ブロックで要求されるデータのタイプと数は異なる。具体的な入力データの内容については次節で述べるが、以下では一般的なデータ作成方法について述べておく。入力データ作成マニュアル(第5.3節)は、概ね次のような形式をとっている。

```
*****
Block名      入力ブロックのタイトル          /データのタイプと数/
変数名      入力変数の意味
:
[注意] データ作成に関する諸注意
[詳細] 本報告書の中で詳細な説明がしてある節の番号など
*****
```

一般的には、1つの入力ブロック内には、文字型データ、整数型データ、実数型データをそれぞれ複数個入力することができる。各ブロックに応じた内容をスラッシュ記号' / 'で囲まれた部分に記述する。そこでは、次のような7つのバリエーションがある。

- |  |            |
|--|------------|
| 1) 1文字の文字型データを入力する場合                             | /A1/       |
| 2) $m$ 個の整数型データを入力する場合                           | /m/        |
| 3) $n$ 個の実数型データを入力する場合                           | /n/        |
| 4) 1文字の文字型データと $m$ 個の整数型データを入力する場合               | /A1, m/    |
| 5) 1文字の文字型データと $n$ 個の整数型データを入力する場合               | /A1, n/    |
| 6) $m$ 個の整数型データと $n$ 個の実数型データを入力する場合             | /m, n/     |
| 7) 1文字の文字型データと $m$ 個の整数型データと $n$ 個の実数型データを入力する場合 | /A1, m, n/ |

2)と3)、及び4)と5)の区別は、変数名が整数型(I～Nで始まる)であるか、実数型(A～H, O～Zで始まる)であるかにより判断する。なお、変数名は便宜的なもので、必ずしもプログラムソース内で使用されている変数名と一致しているとは限らない。

## 5.3 入力データの作成

### 5.3.1 全体計算制御(セクション-1)

【本セクションに対する入力は最初の計算ケースに対してのみ必要】

Block-1-1	セクション名(常時必要)	/A8/
KEYWOD	8文字'%TIMESET'を入力する。	

[注意]：各入力セクションの先頭に付けるセクション名である。入力データ全体の整合性(データの過不足やタイプ)をプログラムがセクション単位でチェックするために使用される。必ず大文字で入力する。

Block-1-2	計算時間上限(常時必要)	/1/
-----------	--------------	-----

CPULIM	分単位の計算時間(CPU)の上限
--------	------------------

[注意]：複数の計算ケースを行う場合には、全ケースにわたる計算時間を入力する。使用する計算機システムが設定する上限がある場合には、これよりも小さく設定する。

[詳細]：第4.2.6節の(3)

### 5.3.2 ケース毎の計算制御(セクション-2)

【本セクションに対する入力は各計算ケース毎に常時必要】

Block-2-1	セクション名(常時必要)	/A8/
KEYWOD	8文字'%CONTROL'を入力する。	

Block-2-2	ケース名(常時必要)	/A4/
-----------	------------	------

CASEID	任意の4文字から成る英数字
--------	---------------

[注意]：計算ケースを区別するための名前である。また、出力バイナリーファイルのファイル名(8文字)の先頭4文字として利用される(第7.4節参照)。小文字や特殊文字も許されるが、ファイル名として適切な文字列を入力する(以下に述べる場合を除いてブランクを含んではならない)。4文字を全てブランクとする場合には、一連の計算ケースを終了する。この場合には、以降の入力データは全て不要である。

Block-2-3	各計算ケースに対する1行目コメント(常時必要)	/A72/
-----------	-------------------------	-------

TITLE1	各計算ケースに対する覚え書き
--------	----------------

[注意]：計算条件やモデルなど、ユーザーが自由に書くことができる。大文字、

小文字、特殊文字(フリーフォーマットで使用されるものも含む)などの使用は自由である。また、全てプランクとしても構わない。

この情報は、プリント出力されるが、出力バイナリーファイルの内容には残されない。

Block-2-4	各計算ケースに対する2行目コメント(常時必要)	/A72/
TITLE2	各計算ケースに対する覚え書き(Block-2-3と同じ)	

Block-2-5	断面積ファイルのアロケーション(常時必要)	/A72/
PREFIX(1)	実行用シェルスクリプトにおいて、本計算ケースで使用する巨視的断面積ファイル(ディレクトリ)に対応する環境変数名を入力する。 例えば、シェルスクリプトにおいて以下のように環境変数(MACRO1～MACRO3)を定義したものとする。	

```
setenv MACRO1 /home/okumura/MacroXS1
setenv MACRO2 /home/okumura/MacroXS2
setenv MACRO3 /home/okumura/MacroXS3
```

ただし、/home/okumura/MacroXS1～MacroXS3は、巨視的断面積メンバー(第7.3節参照)を含むディレクトリのパス名である。これらのうち、ディレクトリMacroXS3の配下に存在する巨視的断面積メンバーを使用したい場合には、PREFIX(1)に文字列'MACRO3'を入力する。

[注意]：標準入力から巨視的断面積を与える場合(Block-2-8のIOPT(4)=1)には、入力された断面積は、ここで指定するディレクトリ配下に巨視的断面積メンバーとして書き込まれる。

[詳細]：第4.2.8節、第6.1節、第7.3節

Block-2-6	Eggファイルのアロケーション(常時必要)	/A72/
PREFIX(2)	MOSRA-Lightの主要な計算結果は全て1つのディレクトリ配下に複数のメンバーとして収納される。このディレクトリファイルのことをEggファイルと呼ぶ。PREFIX(2)には、本計算ケースで使用するEggファイルに対応する環境変数名を入力する。 例えば、シェルスクリプトにおいて以下のように環境変数(EGG1～EGG3)を定義したものとする。	

```
setenv EGG1 /home/okumura/Egg1
setenv EGG2 /home/okumura/Egg2
setenv EGG3 /home/okumura/Egg3
```

ただし、/home/okumura/Egg1～Egg3は、ユーザーが作成するディレクトリファイルのパス名である。これらのうち、Egg2に計算結果(メンバー)を出力させたい場合には、PREFIX(2)に文字列'EGG2'を入力する。

[詳細]：第6.1節、第7.4節

Block-2-7	エネルギー群情報(常時必要)	/2/
NG	全エネルギー群数( $NG \geq 1$ )	
	[注意] : SRACコードが出力する断面積を使用する場合には、エネルギー群数に関する情報を断面積ファイルが持っており、そのエネルギー群数とここで指定するエネルギー群数が不一致する場合には、エラーメッセージが出されて計算はストップする。	
	[詳細] : 第7.3節の(1)	
NGT	熱群のエネルギー群数( $0 \leq NGT \leq NG$ )	
	[注意] : 例えば、高速群が4群、熱群が3群で構成される場合、 $NG=7$ 、 $NGT=3$ とする。ここで言う熱群とは、上方散乱の影響を受ける群のことである。 $NGT$ は、熱群反復計算を行うエネルギー範囲を決めるために使用され、実際の値( $NGT$ -Real)と一致していなくても計算結果には影響しない。 $NGT < NGT$ -Realの場合には、入力された熱群の範囲内で熱群反復計算が行われ、収束が遅くなることがある。 $NGT > NGT$ -Realの場合には、プログラムが断面積から $NGT$ -Realを調べ、実際に上方散乱の影響を受ける群内で熱群反復計算が行われ効率的な計算となる。ただし、熱群反復計算に必要なワークメモリーが $NGT$ に比例して多くなる。 なお、熱群反復計算を強制的に行わないようにするには、 $NGT$ に1または0を入力する。	
	[詳細] : 第4.2.3節	
Block-2-8	一般制御(常時必要)	/20/
IOPT(1)	問題のタイプ	
	=0 固有値問題	
	=1 固定源問題(セクション-10が必要)	
IOPT(2)	スタートオプション	
	=0 コールドスタート	
	=1 リスタート	
	=2 初期値参照コールドスタート(セクション-5が必要)	
	[注意] : リスタート計算の場合には、前に行った計算(未収束)で得られるEggファイルを使用し、入力データは本項以外を変更する必要は無い。計算の本質に関わる幾何形状、断面積など以外の入力データは変更可能である。ただし、Block-2.2で指定する計算ケース名は変更してはならない。初期値参照コールドスタートでは、初期値を参照する計算ケースで得られたEggファイルを使用する。幾何形状モデル、境界条件、エネルギー	

群数が同じであれば参照可能である。またNGT(Block-2-7)やNEMの次数(IOPT(5))が異なっていても構わない。Block-2-2で指定する計算ケース名が参照するケース名(Block-5-2)と同じ場合には、Eggファイルの中にある古い計算結果は失われ、本計算ケースの結果で置き換えられる。

[詳細]：第4.2.6節、第7.4節

**IOPT(3) 中性子束のタイプ**

- =0 通常の中性子束 (forward flux)
- =±1 随伴中性子束 (adjoint flux)
- =±2 通常の中性子束と随伴中性子束の両方

[注意]：IOPT(3)<0にすると、随伴中性子束の計算をエネルギー群を逆転せずに計算を行う。エネルギー群数が多い場合には、随伴中性子束計算の収束速度は遅くなる。符号によりEggファイルに書き出される随伴中性子束と随伴部分中性子流のデータがエネルギー群について反転する。

IOPT(3)=±2の場合には、先ず通常の中性子束計算が行われ、続けて随伴中性子束の計算が行われる。

[詳細]：第4.2.2節、第7.4節の(3), (5)

**IOPT(4) 巨視的断面積の与え方**

- =0 バイナリーファイル(MACRO)から読み込む
- =1 標準入力で与える(セクション-6が必要)
- =2 Eggファイルから読み込む(将来オプション)

[注意]：IOPT(4)=0の場合には、SRACコードで出力される断面積ファイルそのまま利用することができる。IOPT(4)=1の場合には、(n,2n)反応の補正(第4.2.9節参照)は行われない。

[詳細]：第4.2.8節、第4.2.9節、第7.3節

**IOPT(5) 多項式展開ノード法の展開次数**

- =0 4次のNEM(粗メッシュ計算)
- =1 2次のNEM(詳細メッシュ計算)

[詳細]：第4.2.4節

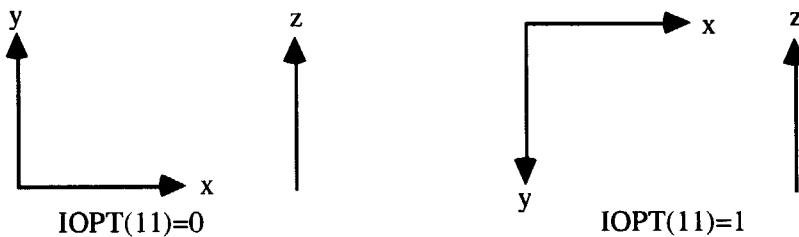
**IOPT(6) 編集領域(Edit-Region)の利用**

- =0 Edit-Regionオプションを使用しない。
- =1 Edit-Regionオプションを使用する。(セクション-7が必要)

[詳細]：第4.2.14節

- IOPT(7) ユーザー編集ルーチンの使用オプション  
 =0 ユーザー編集ルーチンを使用しない。  
 =1 ユーザー編集ルーチンを使用する。  
 [注意]：ユーザー編集ルーチンを使用するには、ユーザーがプログラムを作成し、ロードモジュールを再作成する。  
 [詳細]：第4.2.15節、付録A-1
- IOPT(8) バックリングオプション  
 =0 バックリングを使用しない。  
 =1 バックリングを使用する。(セクション-8が必要)  
 [注意]：セクション-8で入力する群依存のバックリング値を使用して、除去断面積に $D_{u,g}B_g^2$ が加算される。 $D_{u,g}$ は方向依存の拡散係数であり、どの方向の $D_{u,g}$ を使用するかは、セクション-4で指定する。なお、3次元計算の場合にもバックリングオプションは有効である。  
 [詳細]：第4.2.12節の(3)
- IOPT(9) 内部黒体オプション  
 =0 内部黒体を使用しない。  
 =1 内部黒体を使用する。(セクション-9が必要)  
 [注意]：内部黒体の種類は複数許される。また、内部黒体は必ずしも体系の内部にある必要は無い。特にベクトル計算では、内部黒体を使用しない場合に比べて計算時間が長くなる。  
 [詳細]：第4.2.11節、第4.2.12節の(4)、第2.8.5節
- IOPT(10) 外部反復計算における収束判定方法の選択  
 =0 ノード毎の核分裂源(出力分布)のエラーで判定  
 =1 ノード毎の中性子束のエラーで判定  
 =2 上記(0または1)に加え、ノード毎の部分中性子流のエラーで判定  
 [注意]：指定値が大きいほど計算時間は長くなるため、用途に応じて使い分ける。反射体など炉心に影響しない領域の中性子束分布を重要視しない場合にはIOPT(10)=0で良い。More-MOSRAによりノード内中性子束分布の計算(中性子束再構成)を行う場合など、ノード表面平均中性子束の精度を期待する場合には、IOPT(10)=2を指定する。  
 [詳細]：第4.2.7節
- IOPT(11) Y軸方向の選択(次ページの図参照)  
 =0 標準座標系(下から上へ)  
 =1 反転座標系(上から下へ)

[詳細]：第4.2.13節



IOPT(12) 不連続因子オプション

- =0 不連続因子を使用しない。
- =1 不連続因子を入力する。(セクション-11が必要)

[詳細]：第2.7節、第7.3節の(5)

IOPT(13) 反復計算の収束制御パラメータの設定

- =0 デフォルト(IOPT(13)=+1と同じ)
- =±1 コードによる自動最適化モード
- =±2 高速計算優先モード
- =±3 安定収束優先モード
- =±4 ユーザー調整モード (Block-2-10が必要)

[注意]：IOPT(13)≠±4の場合には、外部反復計算、内部反復計算、熱群反復計算に関わる反復制御パラメータにコードが設定する推奨値が使用される（設定値は6番のファイルにプリントされる）。IOPT(13)=±4の場合には、これらをユーザーがBlock 2-10で入力する。

IOPT(13)≥0の場合には、反復計算の収束判定因子などのパラメータにコードが設定するデフォルト値(Block-2-11参照)が使用される。IOPT(13)<0の場合には、これらをユーザーがBlock 2-11で入力する。

[詳細]：第4.2.7節の(4)

IOPT(14) Eggファイルへのメンバー出力オプション

- <0 Eggファイルには何も出力しない。
- =0 主要メンバーのみを出力する。
- =+2 主要メンバーに加えノード毎の断面積をファイルに出力する。
- =+4 主要メンバーに加えノード毎の不連続因子をファイルに出力する。

[注意]：主要メンバーとは、幾何形状情報、中性子束、中性子流、出力分布に関するメンバーである。IOPT(14)<0の場合には、反復計算が途中で終了してもリスタート計算はできない。

IOPT(14)>0の場合には、主要メンバーに加えノード毎の巨視的断面積または不連続因子データをEggファイルのメンバーとして出力する

(IOPT(3)=±1では無効)。双方を出力する場合には、IOPT(14)=2+4=6を入力する(加算入力)。

[詳細]：第7.4節

IOPT(15)	=0 未使用
IOPT(16)	=0 未使用
IOPT(17)	=0 未使用
IOPT(18)	=0 未使用
IOPT(19)	=0 未使用
IOPT(20)	=0 未使用

Block-2-9 プリント制御(常時必要) /10/

IPRN(1)	ノードマップ =0 プリントしない。 =+2 ノード毎の物質番号をプリントする。 =+4 ノード毎のEdit-Region番号をプリントする(IOPT(6)=1で有効)。 =+8 コード内部で使用されるノード番号をプリントする。
---------	--

[注意]：加算入力で指定する。例えば、物質番号とEdit-Region番号の双方をプリントするには、IPRN(1)=2+4=6を指定する。また、全てをプリントするには、IPRN(1)=2+4+8=14を指定する。

物質番号とEdit-Region番号は、それぞれセクション-3とセクション-7で入力する。また、個々のノードには、境界分離チェックボードスワイプが行われる順序に従い、連続的な1次元の番号が付けられる(Fig.3.4.2参照)。これをノード番号と呼ぶが、一般のユーザーがこれを認識する必要はほとんど無い。ノード番号がプリントされる場合には、チェックボード状に赤色ノードに対して正の番号、黒色ノードに対しては負の番号が表示される。

なお、ノードマップはIOPT(11)で指定した座標系で表示される。

[詳細]：第4.2.14節、第3.4.2節

IPRN(2)	外部反復計算の収束モニター =0 外部反復計算の最後の収束情報のみをプリントする。 =1 外部反復計算毎に収束情報をプリントする。 =2 デバッグ用の詳細情報をプリントする。
---------	--

[注意]：収束情報には、固有値、ポイント中性子源やポイント中性子束の最大エラーなどが含まれる。

[詳細]：第4.2.7節

IPRN(3) 計算ステップモニター

- =0 標準
- =1 詳細(外部反復毎の情報)
- =2 デバッグ用(外部反復/群毎の情報)
- =3 デバッグ用(外部反復/群/内部反復毎の情報)

[注意]：計算ステップモニターは、反復計算で使用したサブルーチン名、経過時間、収束状況などを時系列的に6番のファイルにプリントする。IPRN(3)>1の場合にはプリント量が膨大となる。

[詳細]：第7.1節

IPRN(4) 出力分布のプリント

- =0 プリントしない。
- =+2 ノード単位でプリントする。
- =+4 Edit-Region単位でプリントする(IPRT(6)=1で有効)。

[注意]：加算入力で指定する。

IPRN(5) 中性子束分布のプリント

- =0 プリントしない。
- =+2 ノード単位でプリントする。
- =+4 Edit-Region単位でプリントする(IPRT(6)=1で有効)。

[注意]：加算入力で指定する。ただし、入力値の符号によりプリント形式を以下のように選択できる(Edit-Regionに対する出力のみに有効)。

----- IPRN(5)>0 -----				----- IPRN(5)<0 -----			
Region	1-Group	2-Group	...	Group	1-Region	2-Region	...
1				1			
2				2			
3				3			

[詳細]：第4.2.14節の(2)

IPRN(6) 外部中性子源分布のプリント(IPRT(1)=1の場合に有効)

- =0 プリントしない。
- =1 ノード単位でプリントする。

[詳細]：第4.2.1節

IPRN(7) 物質単位の巨視的断面積のプリント

- =0 プリントしない。
- =1 物質単位でプリントする。

IPRN(8)	表面平均中性子流のプリント	
	=0 プリントしない。	
	=+2 ノード単位で6面の正味中性子流をプリントする。	
	=+4 ノード単位で6面の部分中性子流をプリントする。	
[注意]	: 加算入力で指定する。プリント量は、ノード総数×エネルギー群×6面 ×(1: 正味中性子流、2: 部分中性子流)に比例する。	
IPRN(9)	=0 未使用	
IPRN(10)	=0 未使用	
Block-2-10	反復計算制御 (IOPT(13)=±4の場合に必要)	/10/
ITCTL(1)	外部反復計算の制御	
	=0 収束したら外部反復計算を終了する。	
	=1 外部反復計算が収束しても反復計算を続行する。	
[注意]	: ITCTL(1)=1の場合、収束判定に関係なくITCTL(3)で指定する反復計算回数の上限または計算時間の上限(Block-1-2)に至るまで外部反復計算が続行される。計算時間が許す限り精度を上げる場合やデバッグに使用する。	
ITCTL(2)	外部反復計算当たりの内部反復計算回数の上限	
	=0 デフォルト値(20)を使用する。	
	=n 入力値(n)を使用する。	
[注意]	: 一般にnが小さいと計算時間は短くなるが、ノード幅が小さい場合にnを過度に小さく設定すると収束が不安定となる。 リストア計算以外で、外部反復回数がコード設定条件を越えても収束しない場合には、内部反復回数の上限値は本入力値の2倍の値に変更される。	
[詳細]	: 第4.2.7節	
ITCTL(3)	外部反復計算回数の上限	
	=0 デフォルト値(300)を使用する。	
	=n 入力値(n)を使用する。	
[注意]	: 外部反復計算回数が本設定値を越えると、警告を出して計算を終了する。プリント制御(Block-2-9)で指定した項目は出力される他、Eggファイルへのメンバー出力も行われる。	
[詳細]	: 第4.2.7節	

## ITCTL(4) 热群反復計算回数の上限

- =0 デフォルト値(10)を使用する。
- =n 入力値(n)を使用する。

[注意]：热群反復計算が行われない場合(Block-2-7のNGT参照)には、本入力は無効である。また、ITCTL(4)=1とする場合には热群反復計算は行われない。なお、热群反復計算は計算結果には影響しないが、特にエネルギー群数が多い場合に収束速度に影響する。

[詳細]：第4.2.3節

## ITCTL(5) 外部反復計算過程における负の中性子束の対処

- <0 無視して計算を続行する。
- =0 负の中性子束を強制的にゼロ値に置き換え計算を続行する。
- >0 デバッグ用

[注意]：负の中性子束が現れると収束が不安定となることがある。多くは、外部反復の過程で消滅するが、場合によってはそのまま収束しなくなることもある。ITCTL(5)=0によりこれを強制的に排除するが、ゼロ値として収束することがある(警告がプリントされる)。

## ITCTL(6) 2次のNEMによる4次のNEMの加速

- <0 加速法を使用しない(始終4次のNEMで計算)。
- =0 最初の外部反復20回(推奨値)を限度として2次のNEMで計算する。
- =n 最初の外部反復n回を限度として2次のNEMで計算する。

[注意]：本オプションは、2次のNEMの計算(IOPT(5)=1)、リスタート計算(IOPT(2)=1)、初期値参照コールドスタート計算(IOPT(2)=2)では無効である。指定回数以下であっても2次のNEMが収束した場合には、即座に4次のNEMへ移行して計算を続行する。本加速法はメッシュサイズが小さい場合に有効である。

[詳細]：第4.2.5節の(2)

## ITCTL(7) 中性子源外挿法による加速

- <0 加速法を使用しない。
- =0 加速法を使用する。

[詳細]：第4.2.5節の(1)

ITCTL(8) =0 未使用

ITCTL(9) =0 未使用

ITCTL(10) =0 未使用

Block-2-11	収束判定因子 (IOPT(13)<0の場合に必要) IOPT(13)≥0の場合には、EPSCV(1~10)=0.0 (デフォルト)の設定となる。	/10/
EPSCV(1)	外部反復計算における核分裂源 / 中性子束 / 部分中性子流 (IOPT(10)参照)に対する収束判定因子 =0.0 デフォルト値(1.0E-4)を使用する。 ≠0.0 入力値を使用する。 [詳細] : 第4.2.7節 (4.2.58)式	
EPSCV(2)	外部反復計算における固有値に対する収束判定因子 =0.0 デフォルト値(1.0E-5)を使用する。 ≠0.0 入力値を使用する。 [注意] : 固定源問題(IOPT(1)=1)の場合には無効である。 [詳細] : 第4.2.7節 (4.2.57)式	
EPSCV(3)	内部反復計算における収束判定因子の初期値 =0.0 デフォルト値(1.0E-3)を使用する。 ≠0.0 入力値を使用する。 [注意] : 内部反復計算における収束判定因子は、外部反復計算に連動して動的に変更される。一定値を使用したい場合には負の値を入力する。 [詳細] : 第4.2.7節	
EPSCV(4)	固有値(実効中性子増倍率)の下限値 =0.0 デフォルト値(1.0E-2)を使用する。 ≠0.0 入力値を使用する。 [注意] : 固定源問題(IOPT(1)=1)の場合には無効である。固有値問題の外部反復計算の過程で、本項で入力する下限値を超える固有値が検出された場合には、計算はストップする。	
EPSCV(5)	固有値(実効中性子増倍率)の上限値 =0.0 デフォルト値(1.0E+2)を使用する。 ≠0.0 入力値を使用する。	
EPSCV(6)	=0.0 未使用	
EPSCV(7)	=0.0 未使用	
EPSCV(8)	=0.0 未使用	
EPSCV(9)	=0.0 未使用	
EPSCV(10)	=0.0 未使用	

Block-2-12	原子炉の熱出力 (常時必要)	/1/
POWER	中性子束と出力分布の規格化方法の指定	
	> 0.0 本項で入力する計算体系の熱出力(ワット)で規格化(4.2.75式)	
	= 0.0 規格化を行わない。	
	= -1.0 出力分布が平均値に対する相対値になるように規格化 (4.2.82式)	
	= -2.0 体系全体の吸収反応率が1.0になるように規格化 (4.2.81a式)	
	= -3.0 体系全体の核分裂反応率が1.0になるように規格化 (4.2.81b式)	
	= -4.0 体系全体の生成反応率が1.0になるように規格化 (4.2.81c式)	
[注意]	： 固有値問題(IOPT(1)=0)において本項に正の値を入力すると、中性子束レベルは入力した熱出力に規格化され、出力分布は[W/cm <sup>3</sup> ]の単位で出力される。ただし、1/4炉心などの対称体系を扱う場合には、熱出力も1/4として与える(4.2.85式参照)。また、2次元または1次元計算の場合には、(4.2.86)式、(4.2.87)式を使用する。対角対称境界条件(Block-3-3でIBCD(7)<0)を使用する場合には(4.2.88)式を使用する。	
	固有値問題でPOWER=0.0とすると、体系全体の生成反応率が固有値と一致するように規格化される(4.2.80式)。	
	POWER=-1とすると、全体系の熱出力が核分裂領域体積と一致するよう規格化され、結果的に出力分布は炉心平均出力密度に対する相対値(ピーピング係数)として出力される。	
	随伴問題のみの計算(IOPT(3)=1)でも本項の入力は必要であるが、入力値に関係なく随伴中性子束の規格化は行われない。	
	固定源問題(IOPT(1)=1)の場合には中性子束レベルはセクション-10で入力する中性子源強度で決まるため、自動的にPOWER=0.0の扱いとなる。	
[詳細]	： 第4.2.10節	

### 5.3.3 幾何形状入力 (セクション-3)

【本セクションに対する入力は各計算ケース毎に常時必要】

Block-3-1	セクション名 (常時必要)	/A8/
KEYWORD	8文字'%GEOMSET'を入力する。	
Block-3-2	ノード分割数 (常時必要)	/3/
NX	x方向ノード分割数	
NY	y方向ノード分割数	
NZ	z方向ノード分割数	

[注意]：NX、NY、NZの何れかがゼロの場合( $NX*NY*NZ=0$ )には、前の計算ケースの幾何形状データをそのまま利用することができる(断面積やEdit-Regionを変更するような場合)。この場合、本セクションにおける以降のBlockに対する入力は不要である。例えば、連続計算ケースで制御棒を動かすような場合には、制御棒の移動範囲に予め異なる物質番号を割り当てておけば、以降の入力を省略することができる。

NXの値が負の場合( $NX<0$ )には、メッシュオーバーレイオプション(Block-3-8)を使用をして、Block-3-7で入力するノードと物質番号の対応を上書き修正することができる。制御棒などが部分的に存在する体系の記述に便利である。

Block-3-3	境界条件 (NX*NY*NZ≠0の場合に必要)		/7/
IBCD(1)	x(-)方向の境界条件		
	=1 真空境界条件		
	=2 反射境界条件		
	=3 周期境界条件 (IBCD(2)=3とする)		
	=4 90度回転境界条件 (IBCD(3)=4とする)		
[注意]	真空境界条件に使用される外挿定数を変更する場合やアルベド境界条件を使用する場合には、黒体を拡張利用する(第4.2.11節参照)。		
	90度回転境界条件を設定するには、以下の条件が必要である。ただし、XXX(i)とYYY(i)はそれぞれBlock-3-4とBlock-3-5で入力する値である。		
	$NX=NY$ 、 $(XXX(i)=YYY(i), i=1 \sim NX)$		
[詳細]	第2.8.1節(真空)、第2.8.2節(反射)、第2.8.3節(周期)、第2.8.4節(90度回転) 第2.8.7節(アルベド)		
IBCD(2)	x(+)方向の境界条件		
	=1 真空境界条件		
	=2 反射境界条件		
	=3 周期境界条件 (IBCD(1)=3とする)		
IBCD(3)	y(-)方向の境界条件		
	=1 真空境界条件		
	=2 反射境界条件		
	=3 周期境界条件 (IBCD(4)=3とする)		
	=4 90度回転境界条件 (IBCD(1)=4とする)		
IBCD(4)	y(+)方向の境界条件		
	=1 真空境界条件		

- =2 反射境界条件
- =3 周期境界条件 (IBCD(3)=3とする)

IBCD(5) z(-)方向の境界条件  
 =1 真空境界条件  
 =2 反射境界条件  
 =3 周期境界条件 (IBCD(6)=3とする)

IBCD(6) z(+)方向の境界条件  
 =1 真空境界条件  
 =2 反射境界条件  
 =3 周期境界条件 (IBCD(5)=3とする)

IBCD(7) 対角対称境界条件  
 <0 対角対称境界条件を使用する。  
 =0 対角対称境界条件を使用しない。

[注意]：対角対称境界条件を使用するには、以下の条件が必要である。ただし、  
 XXX(i)とYYY(i)はそれぞれBlock-3-4とBlock-3-5で入力する値である。

NX=NY、(XXX(i)=YYY(i), i=1~NX)、IBCD(1)=IBCD(3)

y軸座標系の選択(IOPT(11))により、対角対称線は下図のようになる。



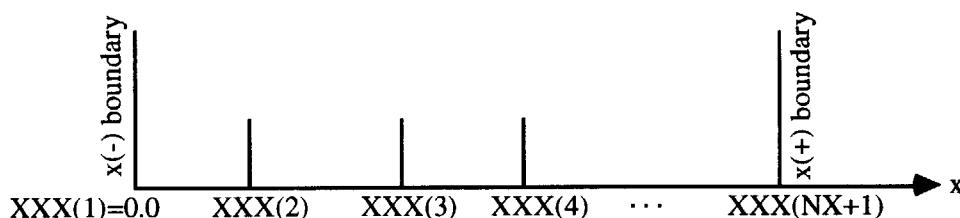
[詳細]：第4.2.10節の(3)、第2.8.6節

Block-3-4 x方向ノード分割位置 (NX\*NY\*NZ≠0の場合に必要) /NX/

(XXX(i+1), i=1,NX)

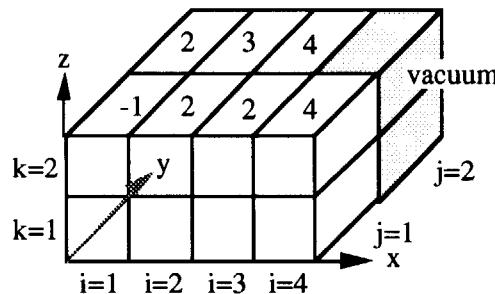
x方向ノード分割位置を原点からの距離(cm)で入力する。

原点XXX(1)=0.0の入力は不要である。



Block-3-5	y方向ノード分割位置 (NX*NY*NZ≠0の場合に必要) (YYY(j+1), j=1,NY)	/NY/
	y方向ノード分割位置を原点からの距離(cm)で入力する。 原点YYY(1)=0.0の入力は不要である。	
Block-3-6	z方向ノード分割位置 (NX*NY*NZ≠0の場合に必要) (ZZZ(k+1), k=1,NZ)	/NZ/
	z方向ノード分割位置を原点からの距離(cm)で入力する。 原点ZZZ(1)=0.0の入力は不要である。	
Block-3-7	ノード毎の物質番号 (NX*NY*NZ≠0の場合に必要) (((IDMAT(i,j,k), i=1,NX), j=J1,J2), k=1,NZ)	/NX*NY*NZ/
	IOPT(11)=0 (標準座標系)の場合 : J1=NY, J2=1 IOPT(11)=1 (反転座標系)の場合 : J1=1, J2=NY  ノード位置(i,j,k)に配置する物質番号を与える。入力順は例えば標準座標系を使用する場合には以下のようになる。  IDMAT(1, NY, 1)、IDMAT(2, NY, 1)、IDMAT(3, NY, 1)、… IDMAT(NX, NY, 1)、 IDMAT(1, NY-1, 1)、IDMAT(2, NY-1, 1)、IDMAT(3, NY-1, 1)、… IDMAT(NX, NY-1, 1)、 IDMAT(1, NY-2, 1)、IDMAT(2, NY-2, 1)、IDMAT(3, NY-2, 1)、… IDMAT(NX, NY-2, 1)、 ⋮  IDMAT(1, 1, 1)、IDMAT(2, 1, 1)、IDMAT(3, 1, 1)、… IDMAT(NX, 1, 1)、 IDMAT(1, NY, 2)、IDMAT(2, NY, 2)、IDMAT(3, NY, 2)、… IDMAT(NX, NY, 2)、 ⋮  IDMAT(1, NY, NZ)、IDMAT(2, NY, NZ)、IDMAT(3, NY, NZ)、… IDMAT(NX, NY, NZ)	
	[注意] : 物質番号はセクション-4で登録する断面積セットの入力順に対応する。 飛び番号(本計算ケースでは使用しない物質)があっても構わない。ただし、 IDMAT ≤NMAT(=Block-4-2で入力する物質数)とする。 計算体系から除外する真空ノードに対してはゼロ値を入力する。対角対称境界条件(BCD(7)<0)を使用し、計算の対象外(i<j)となるノードに対しては入力値に関係なくコード内部でゼロ値が与えられる。 黒体ノードには負の物質番号を与える。全エネルギー群について断面積を使用しない完全黒体に対しても、何らかの物質番号(負)を与えておく必要がある(反射体物質などなんでも良い)。 例えば下図(左)のように標準座標系を使用する場合で、z軸方向に同じ物質がある体系の入力は下図(右)のようになる。	

IOPT(11)=0, NX=4, NY=2, NZ=2



## Input Data for IDMAT

2	3	4	0
-1	2	2	4
2	3	4	0
-1	2	2	4

/ k=1  
/ k=2

[注意]：反転座標系を使用する場合(IOPT(11)=1)でも、IDMATの見た目の入力は変わらないが(第4.2.13節参照)、y軸の向きが反転するため、y軸方向のノード分割位置(Block-3-5)と境界条件(IBCD(3)とIBCD(4))の入力が標準座標系を使用する場合と異なることに注意する。

[詳細]：第4.2.13節(反転座標)、第4.2.11節の(2)(真空ノード)、第4.2.11節、第4.2.12節の(4)(黒体)

## Block-3-8

メッシュオーバーレイ (NX&lt;0の場合に必要)

/7/

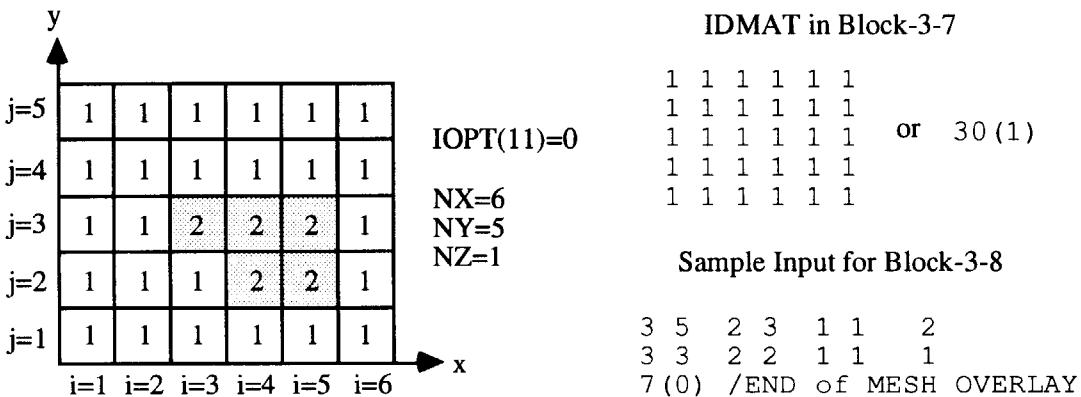
LXYZ(1)～LXYZ(6)の何れかに0を入力するまで( $\prod_{i=1}^6 LXYZ(i)=0$ )、  
Block-3-8を繰り返し入力する。

- LXYZ(1) x方向オーバーレイ始点ノード位置
- LXYZ(2) x方向オーバーレイ終点ノード位置
- LXYZ(3) y方向オーバーレイ始点ノード位置
- LXYZ(4) y方向オーバーレイ終点ノード位置
- LXYZ(5) z方向オーバーレイ始点ノード位置
- LXYZ(6) z方向オーバーレイ終点ノード位置
- LXYZ(7) オーバーレイ物質番号

[注意]：メッシュオーバーレイは、Block-3-7で入力した物質番号のマップを上書き修正する機能である。

LXYZ(1)≤i≤LXYZ(2)、LXYZ(3)≤j≤LXYZ(4)、LXYZ(5)≤k≤LXYZ(6)で囲まれるノード領域が物質番号LXYZ(7)で置き換えられる。

Block-3-8の繰り返しにおいてノードの位置が重なる場合には、後で入力したものが優先(上書き)される。例えば下図(左)のようなx-y面2次元計算で、物質番号2をメッシュオーバーレイにより入力する一例を下図(右)に示す。



### 5.3.4 物質指定 (セクション-4)

【本セクションに対する入力は各計算ケース毎に常時必要】

Block-4-1 セクション名 (常時必要) /A8/  
KEYWOD 8文字'%MATNAME'を入力する。

Block-4-2 物質数 (常時必要) /1/  
NMAT Block-4-3で入力する物質(断面積セット)の数

Block-4-3 物質指定 (常時必要) /A8,4,1/  
Block-4-3をNMAT回繰り返し入力する(m=1~NMAT)。  
計算に当面使用しない物質が含まれていても構わない。

MTNAME(m) 8文字の物質名  
[注意] : Block-2-5で指定したMACROファイル中に存在するP0断面積メンバーの名前(8文字)を入力する。

断面積をセクション-6で入力する場合(IOPT(4)=1)には、ここで指定する名前のメンバーがMACROファイル中に作成される。一旦、MACROファイルに作成されたメンバーは、以降の計算ケースにおいてメンバー名を指定することにより同じ断面積を利用することができます。

なお、Block-4-3の中には' & から始まるコメントを挿入してはならない。

[詳細] : 第7.3節

IDXYZ(m,1) x方向拡散係数の指定  
=0 断面積セットのD1位置の拡散係数を使用する。  
=1 断面積セットのD1位置の拡散係数を使用する(=0と同じ)。  
=2 断面積セットのD2位置の拡散係数を使用する。  
[注意] : 断面積セットは、2種類の方向依存拡散係数(D1, D2)を収納する。SRAC

コードにより得られる巨視的断面積ファイルを使用する場合には、D1とD2はSRACで使用する格子形状やオプションに依存する<sup>25)</sup>。断面積を標準入力として与える場合(IOPT(4)=1)には、D1とD2をセクション-6で入力する。

[詳細]：第7.3.節の(2)

IDXYZ(m,2) y方向拡散係数の指定

- =0 断面積セットのD1位置の拡散係数を使用する。
- =1 断面積セットのD1位置の拡散係数を使用する(=0と同じ)。
- =2 断面積セットのD2位置の拡散係数を使用する。

IDXYZ(m,3) z方向拡散係数の指定

- =0 断面積セットのD1位置の拡散係数を使用する。
- =1 断面積セットのD1位置の拡散係数を使用する(=0と同じ)。
- =2 断面積セットのD2位置の拡散係数を使用する。

IDXYZ(m,4) バックリング用の拡散係数の指定

- =0 断面積セットのD1位置の拡散係数を使用する。
- =1 断面積セットのD1位置の拡散係数を使用する(=0と同じ)。
- =2 断面積セットのD2位置の拡散係数を使用する。

[注意]：バックリングオプションを使用しない場合(IOPT(8)=0)には、本項の入力は無効である。

[詳細]：第4.2.12節の(3)

XJFIS(m) 1核分裂当たりに放出される熱エネルギー(J/fission)

- =0.0 XJFIS(m)=1.0とする。
- >0.0 入力値を使用する。

[注意]：中性子束や出力密度のレベルを炉の熱出力で規格化する際に使用される(4.2.75式参照)。POWER=-2, -3, -4(Block-2-13)の場合や、非核分裂性物質に対しては、本項の値は意味を持たない。

通常、200MeV程度の一定値として知られるが、厳密には核分裂核種や核分裂エネルギーに依存する。例えばORIGEN-2コード<sup>33)</sup>では、以下の式が使用されている。

$$XJFIS(Z,A) = 1.60219E-19[J/MeV] \times \{ 1.29927E-3 (Z^2 \sqrt{A}) + 33.21 \}$$

ここで、Zは原子番号、Aは質量数であり、例えばU-235とPu-239のXJFISは以下の値で与えられる。

U-235 : 3.233E-17 (201.78MeV)

Pu-239 : 3.376E-17 (210.68MeV)

混合物の値は、例えばSRACコードの格子燃焼計算<sup>25)</sup>により構成核種の核分裂率を重みとする平均値を得ることができる。

[詳細]：第4.2.10節

#### [セクション-4の入力例]

```
%MATNAME
3 / NMAT
FUELA010 1 1 2 0 0.0      / Fuel      (Benoist-D)
BLKTA010 1 1 2 0 0.0      / Blanket   (Benoist-D)
REFLA010 0 0 0 0 0.0      / Reflector
```

### 5.3.5 初期値参照ケースの指定(セクション-5)

【本セクションに対する入力はIOPT(2)=2の場合に必要】

Block-5-1	セクション名(常時必要)	/A8/
KEYWOD	8文字'%GESSID'を入力する。	

Block-5-2	参照ケース名(常時必要)	/A4/
GESSID	初期値を参照するケース名	

[注意]：Block-2-6で指定するEggファイルの中から、GESSID(4文字)から始まる名前の中性子束メンバーと中性子流メンバーを読みとり、反復計算の初期値として利用する。エネルギー群数や幾何形状が異なる計算ケースは参照することができない。また随伴問題でIOPT(3)の符号が異なる計算ケースも参照できない。NEMの次数が異なっていても参照することができる。

Eggファイル中に該当する参照ケースが見つからない場合には、自動的にコールドスタート(IOPT(2)=0)と同じ扱いとなる。

Block-2-2で指定するCASEIDとGESSIDが同じ場合には、リスタート計算と同じ扱いとなり、Eggファイルのメンバーは更新される。

[詳細]：第4.2.6節の(2)、第7.4節

### 5.3.6 断面積値入力(セクション-6)

【本セクションに対する入力はIOPT(4)=1の場合に必要】

Block-6-1	セクション名(常時必要)	/A8/
KEYWOD	8文字'%XSINPUT'を入力する。	

Block-6-2

巨視的断面積(當時必要)

/6+NG/

Block-6-2を群数(NG)回繰り返し入力する( $g=1 \sim NG$ )。更にNGセットのBlock-6-2を物質指定(Block-4-3)の入力順に従って物質数回繰り返し入力する( $m=1 \sim NMAT$ )。

SIGP(g,m)  $v\Sigma_f$  : 巨視的核分裂生成断面積( $\text{cm}^{-1}$ )SIGF(g,m)  $\Sigma_f$  : 巨視的核分裂断面積( $\text{cm}^{-1}$ )SIGA(g,m)  $\Sigma_a$  : 巨視的吸収断面積( $\text{cm}^{-1}$ )SIGX(g,m)  $\chi$  : 核分裂スペクトル

SIGD1(g,m) D1 : 拡散係数(cm)

SIGD2(g,m) D2 : 拡散係数(cm)

(SIGS(g,g',m), $g'=1,NG$ ) $\Sigma_{g \rightarrow g'}$  : 第g群から $g'$ (=1~NG)群への巨視的散乱断面積( $\text{cm}^{-1}$ )

[注意] : Block-4-3で与えた断面積メンバーがMACROファイル中に既に存在している場合には、これに対応するm番目の断面積入力において最初のデータ(SIGP(1,m))に負の値を入力する。2番目以降のデータに対しても入力は必要であるが値は任意で良い。これを怠るとMACROファイル中の同名メンバーを保護するためにエラーとなる(第7.3節)。

拡散方程式を解く際には $\Sigma_f$ ではなく $v\Sigma_f$ が使用されるが、出力分布の計算には $\Sigma_f$ が使用される(4.2.78式参照)。

D1とD2は方向依存の拡散係数<sup>12)</sup>を使用するためのものであり、x,y,z各方向及びバックリングに対する拡散係数との対応は、Block-4-3の入力に従う。

自群散乱断面積 $\Sigma_{g \rightarrow g}$ の値も入力を必要とするが、計算には全く使用されない。

### [セクション-5の入力例]

```
%XSINPUT
& NG=2, XS Sample for IAEA Benchmark (See Table 3.5.1)
& P      F      A      X      D1     D2    Sg->1   g->2
  0.0    0.0    0.01   1.0    1.5    1.5    0.0    0.02 /g=1,Fuel-1
  0.135  0.135  0.08   0.0    0.4    0.4    0.0    0.0   /g=2,Fuel-1
& for next material
  0.0    0.0    0.01   1.0    1.5    1.5    0.0    0.02 /g=1,Fuel-2
  0.135  0.135  0.085  0.0    0.4    0.4    0.0    0.0   /g=2,Fuel-2
& for material already existing in MACRO file
 -1.0  7(0.0)  / g=1, dummy input to skip makeing a member
  8(0.0)        / g=2, dummy input
```

### 5.3.7 編集領域(Edit-Region)の指定 (セクション-7)

【本セクションに対する入力はIOPT(6)=1の場合に必要】

Block-7-1 セクション名 (常時必要) /A8/

KEYWOD 8文字'%EDITRGN'を入力する。

Block-7-2 Edit-Regionの数 (常時必要) /1/

NERGN Block-7-3で定義するEdit-Regionの数。

NERGN=0とすると、前の計算ケースで使用したEdit-Regionの定義をそのまま使用し、この場合にはBlock-7-3の入力は不要である。

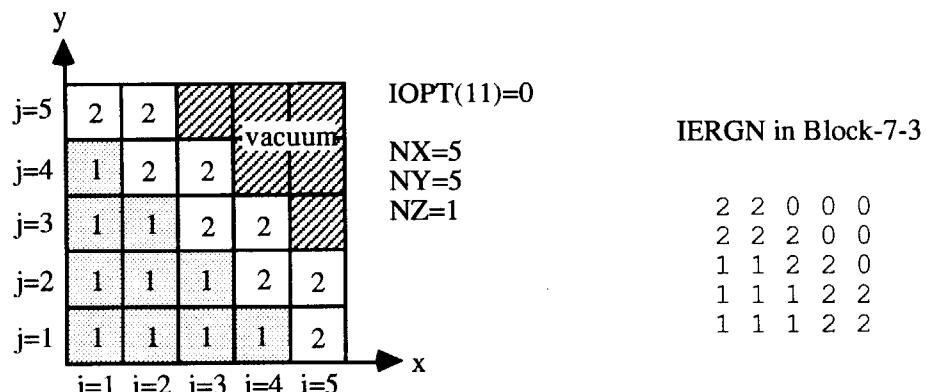
Block-7-3 ノード毎のEdit-Region番号 (NERGN>0の場合に必要) / NX\*NY\*NZ /

((IERN(i,j,k), i=1,NX), j=J1,J2), k=1,NZ)

IOPT(11)=0 (標準座標系)の場合 : J1=NY, J2=1

IOPT(11)=1 (反転座標系)の場合 : J1=1, J2=NY

ノード位置(i,j,k)に対するEdit-Region番号(0～NERGN)を与える。入力方式は、Block-3-7の物質番号の入力に準ずる(下図参照)。



Edit-Regionは中性子束や出力分布の領域平均値をプリント出力する対象領域である。この指定により、例えば炉心部とブランケット平均値、水平方向または軸方向分布などを容易に編集することができる。

[注意]：編集の対象としないノード、あるいは真空ノードや対角対称境界条件を使用し計算の対象外( $i < j$ )となるノードに対してはゼロ値を与える。

IERN(i,j,k)には0または1からNERGNまでの連続番号を入力し、飛び番号は許されない。

Block-3-2でNX\*NY\*NZ=0とする場合にも、実際のNX\*NY\*NZに対応する数の入力が必要である。

[詳細]：第4.2.14節

### 5.3.8 バックリング入力(セクション-8)

【本セクションに対する入力はIOPT(8)=1の場合に必要】

Block-8-1 セクション名(常時必要) /A8/

KEYWOD 8文字'%BUCKLNG'を入力する。

Block-8-2 バックリング(常時必要) /NG/

(BSQ(g), g=1,NG)  
群毎のバックリング値(cm<sup>-2</sup>)

[詳細]：第4.2.12節の(3)

### 5.3.9 黒体の入力(セクション-9)

【本セクションに対する入力はIOPT(9)>0の場合に必要】

Block-9-1 セクション名(常時必要) /A8/

KEYWOD 8文字'%BLACKNS'を入力する。

Block-9-2 黒体定数(常時必要) /1, NG/

Block-9-2をIOPT(9)=N回繰り返し入力する。Nは黒体のタイプ数である。

M 黒体のタイプ番号

黒体のタイプ番号は、物質番号の入力(Block-3-7)に負の値を与えたものである。Mは正の値でも負の値でも構わない。

(BKCST(g, |M|), g=1,NG)

各エネルギー群における黒体表面での外挿定数を入力する。

[注意]：完全黒体と見なす場合には外挿定数に0.5(または0.4692)を与える。

ゼロ値を与えたエネルギー群に対しては、黒体の扱いとはならず物質番号|M|の断面積が使用される。ただし、黒体の扱いをするエネルギー群の下群にゼロ値を入力してはならない。

[詳細]：第4.2.11節、第4.2.12節の(4)、第2.8.5節

[セクション-9の入力例]

```
%BLACKNS
& Material Map for Block-3-7
& 1 1 1 1
& 1 -2 1 1      NG=5, IOPT(9)=2
& 1 1 -3 1
& 1 1 1 1
```

```

&
-2 0.0 0.0 0.5 0.5 0.5 / Black for g=3-5 for Materail 2
-3 0.0 0.0 0.0 0.5 0.5 / Black for g=4-5 for Materail 3

```

### 5.3.10 外部源の入力 (セクション-10)

【本セクションに対する入力はIOPT(1)=1の場合に必要】

Block-10-1	セクション名 (常時必要)	/A8/
KEYWOD	8文字'%EXTSOUR'を入力する。	
Block-10-2	外部源のタイプ数 (常時必要)	/1/
NSOU	Block-10-3～Block-10-5で入力する外部源のタイプ数。	

以下のBlock-10-3～Block-10-5をNSOU回繰り返し入力する(n=1～NSOU)。

Block-10-3	外部中性子源の強度 (常時必要)	/1/
SINTEN(n)	n番目タイプの外部源における中性子生成密度 (個/cm <sup>3</sup> /sec)	
Block-10-4	外部源中性子スペクトル (常時必要)	/NG/
(SPECTR(g, n), g=1,NG)	n番目タイプの外部中性子源のエネルギースペクトル	
[注意] :	スペクトルの値はコード内部で1.0に規格化されるため、必ずしも総和を1.0にする必要はない。	
Block-10-5	n番目タイプの外部中性子源のノード配置 (常時必要)	/6/
LXYZ(1,n)～LXYZ(6,n)	の何れかに0を入力するまで( $\prod_{i=1}^6 LXYZ(i,n)=0$ )、 Block-10-5を繰り返し入力する。	
LXYZ(1,n)	外部源(n)を配置するx方向始点ノード位置	
LXYZ(2,n)	外部源(n)を配置するx方向終点ノード位置	
LXYZ(3,n)	外部源(n)を配置するy方向始点ノード位置	
LXYZ(4,n)	外部源(n)を配置するy方向終了ノード位置	
LXYZ(5,n)	外部源(n)を配置するz方向始点ノード位置	
LXYZ(6,n)	外部源(n)を配置するz方向終点ノード位置	
[注意] :	メッシュオーバーレイ(Block-3-8)と同様に、LXYZ(1,n)≤i≤LXYZ(2,n)、 LXYZ(3,n)≤j≤LXYZ(4,n)、LXYZ(5,n)≤k≤LXYZ(6,n)で囲まれるノード領域にn番目タイプの外部中性子源が配置される。	
	Block-10-5の繰り返しにおいて、ノードの位置が重なる場合には、後で 入力したものが優先(上書き)される。	

外部中性子源は、等方かつノード内分布は一様として扱われる。外部源を配置するノードに対しても何らかの物質(断面積)を指定する必要がある。

[詳細]：第4.2.1節

#### [セクション-10の入力例]

```
%EXTSOUR
& Material Map for Block-3-7
& 1 1 1 NG=3, NX=3, NY=3, NZ=1, IOPT(1)=1, IOPT(11)=0
& 3 3 1 MAT2: the 1st external source
& 2 3 1 MAT3: the 2nd external source
&
2 / NSOU (Block-10-2)
2.0E6 / SINTEN for the 1st source
0.8 0.2 0.0 / SPECTRUM for the 1st source
1 1 1 1 1 / 1st source position
0 0 0 0 0 / end of Block-10-5 for the 1st source
1.0E6 / SINTEN for the 2nd source
0.8 0.2 0.0 / spectrum of the 2nd source
1 2 2 2 1 1 / 2nd source position
2 2 1 1 1 1 / 2nd source position
0 0 0 0 0 0 / end of Block-10-5 for the 2nd source
```

#### 5.3.11 不連続因子の入力(セクション-11)

【本セクションに対する入力はIOPT(12)=1の場合に必要】

Block-11-1	セクション名(常時必要)	/A8/
KEYWOD	8文字'%DISCONT'を入力する。	

以下のBlock-11-2～Block-11-4を、Block-11-2のMEMNAMに8文字のブランクを入力するまで繰り返し入力する。なお、本セクションの中には' & 'から始まるコメントを挿入してはならない。

Block-11-2	不連続因子のメンバーノーと回転オプションの指定(常時必要)	/A8, 1/
------------	-------------------------------	---------

MEMNAM	8文字の不連続因子メンバーノー
--------	-----------------

[注意]：Block-2-5で指定するMACROファイル中の不連続因子メンバー(第7.3節参照)を使用する場合には、MACROファイル中に存在する不連続因子メンバーの名前(8文字)を入力する。

不連続因子をBlock-11-3で入力する場合には、MEMNAMに'INPUT-DC'を入力する。

[詳細]：第7.3節の(5)、第2.7節

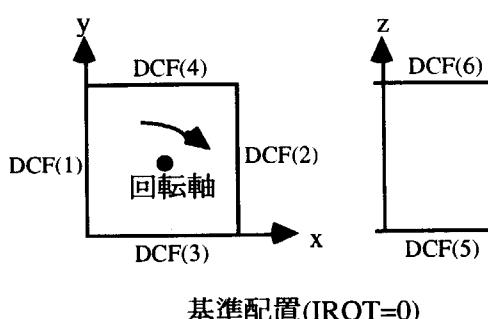
## IROT

不連続因子データの回転オプション

 $\pm 0$  回転なし( $=+4$ ) $\pm 1$  90度回転 $\pm 2$  180度回転 $\pm 3$  270度回転 $\pm 4$  360度回転

[注意]：MEMNAMにブランクを与えて、本セクションの入力を終了させる場合にも、IROTに便宜的に任意の値を与える(入力データ数整合のため)。

ファイルまたは標準入力データで与える不連続因子は、6つの面に対して、x-、x+、y-、y+、z-、z+面の順に与えられる。これらを体系中の各ノード表面に割り当てる際に、IROTによりz方向を回転軸として時計回りに回転させることができる。また、IROT<0により、z-面とz+面の不連続因子を反転させることができる。下図(左)は、IROT=0の場合における6つの不連続因子(DCF(1)～DCF(6))とノード表面との対応を示し、IROTにより下表(右)のような対応に変わる。



IROT=	0,4 / -4	1 / -1	2 / -2	3 / -3
x-面	DCF(1)	DCF(3)	DCF(2)	DCF(4)
x+面	DCF(2)	DCF(4)	DCF(1)	DCF(3)
y-面	DCF(3)	DCF(2)	DCF(4)	DCF(2)
y+面	DCF(4)	DCF(1)	DCF(3)	DCF(1)
z-面	DCF(5) / (6)	DCF(5) / (6)	DCF(5) / (6)	DCF(5) / (6)
z+面	DCF(6) / (5)	DCF(6) / (5)	DCF(6) / (5)	DCF(6) / (5)

Block-11-3-1

x-面不連続因子 (MEMNAM='INPUT-DC'の場合に必要)

/NG/

(DCF(g,1), g=1,NG)

x-面における各エネルギー群の不連続因子

[注意]：Block-11-3-1～Block-11-3-6では、IR OT(Block-11-2)による回転や反転を考慮しない条件(IROT=0)で不連続因子を与える。また、ゼロ値を与えるとデフォルト値(1.0)がセットされる。

Block-11-3-2

x+面不連続因子 (MEMNAM='INPUT-DC'の場合に必要)

/NG/

(DCF(g,2), g=1,NG)

x+面における各エネルギー群の不連続因子

Block-11-3-3	y-面不連続因子 (MEMNAM='INPUT-DC'の場合に必要) (DCF(g,3), g=1,NG) y-面における各エネルギー群の不連続因子	/NG/
Block-11-3-4	y+面不連続因子 (MEMNAM='INPUT-DC'の場合に必要) (DCF(g,4), g=1,NG) y+面における各エネルギー群の不連続因子	/NG/
Block-11-3-5	z-面不連続因子 (MEMNAM='INPUT-DC'の場合に必要) (DCF(g,5), g=1,NG) z-面における各エネルギー群の不連続因子	/NG/
Block-11-3-6	z+面不連続因子 (MEMNAM='INPUT-DC'の場合に必要) (DCF(g,6), g=1,NG) z+面における各エネルギー群の不連続因子	/NG/
Block-11-4	不連続因子のノード割り当て (常時必要) LXYZ(1)～LXYZ(6)の何れかに0を入力するまで( $\prod_{i=1}^6 LXYZ(i)=0$ )、 Block-11-4を繰り返し入力する。  LXYZ(1) 不連続因子を適用するx方向始点ノード位置 LXYZ(2) 不連続因子を適用するx方向終点ノード位置 LXYZ(3) 不連続因子を適用するy方向始点ノード位置 LXYZ(4) 不連続因子を適用するy方向終点ノード位置 LXYZ(5) 不連続因子を適用するz方向始点ノード位置 LXYZ(6) 不連続因子を適用するz方向終点ノード位置  [注意] : Block-11-2、Block-11-3で定義した不連続因子がLXYZ(1)≤i≤LXYZ(2)、 LXYZ(3)≤j≤LXYZ(4)、LXYZ(5)≤k≤LXYZ(6)で囲まれる領域内のノード に適用される。 不連続因子を使用する場合には、全ての群、ノード、表面に対して、 先ず1.0の不連続因子が初期値としてセットされる。その後、ファイル 読み込みまたは標準入力された不連続因子の値がBlock-11-4で指定する ノードに対して上書き変更される。一旦ノードに割り当てられた不連 続因子は、後から入力する不連続因子により何度も上書き変更する ことができる。	/6/

## [セクション-11の入力例]

```

&
&      Y          IOPT(1)=1, IOPT(11)=0
& 3 | B   B   B'  NG=2, NX=3, NY=3, NZ=4
& 2 | A   A   w  A,B : Fuel Assembly Type
& 1 | A   A   w  B' and B" are surrounded by B
& + -----> x
&     1   2   3
&
%DISCONT      / Block-11-1
TYPEBDCF    0   / Block-11-2 for Type-B Nodes
1 3 1 3 1 4 / Block-11-4 Set DCF of Type-B for all Nodes
6(0)          / End of Block-11-4
TYPEADCF    0   / Block-11-2 for Type-A Nodes
1 2 1 2 1 4 / Overwriting DCF for Type-A
6(0)          / End of Block-11-4
TYPEBDCF    1   / Block-11-2 (90-Rotation for Type-w )
3 3 1 2 1 4 / Block-11-4
6(0)          / End of Block-11-4
INPUT-DC    0   / Modification of DCF for Type-B' by Input
2(1.1)        / Block-11-3-1 (for x- surface)
2(1.1)        / Block-11-3-2 (for x+ surface)
2(1.1)        / Block-11-3-1 (for y- surface)
2(1.1)        / Block-11-3-2 (for y+ surface)
2(1.1)        / Block-11-3-1 (for z- surface)
2(1.1)        / Block-11-3-2 (for z+ surface)
3 3 3 3 1 4 / Block-11-4
6(0)          / End of Block-11-4
0   / End of Section-11 (8 Blanks & Dummy IROT)

```

## 6. 実行用シェルスクリプトと入力例

## 6.1 実行用シェルスクリプト

MOSRA-Lightは、UNIXまたはこれと同等なオペレーティングシステム(例えばLinuxやFreeBSD)を使用する計算機で実行することができる。なお、具体的なインストールの方法については付録で述べる。MOSRA-Light実行用シェルスクリプトの一例を以下に示す。

### 【シェルスクリプト例1】

```

#!/bin/csh
#
#####
#
# << run MOSRA-Light >>
#
# Keisuke OKUMURA
# E-mail okumura@mike.tokai.jaeri.go.jp
#
#####
#
# Fortran logical unit usage (allocate if you need)
#
# [ ]:important files for users.
#
# [ 5] text:80 standard input
# [ 6] text:137 standard output, monitoring message
# 49 device internally used to access bnyary file(PDS)
# 95 text:80 (DTLIST), scratch for input list
# [99] text:137 calculated results
#
=====
#
alias mkdir mkdir
alias cd cd
alias rm rm
alias cat cat
#
===== Set by user =====
#
# CASE : case name which is refered as names of output files
# ODR : directory name in which output data will be stored
# INPT : standard input file
# MACRO : directory name of MACRO file (PDS file)
# EGG : directory name of MOSRA output file (PDS file)
# LM : load module
#
set MOSRA_DIR = $HOME/MOSRA-Light.....@
set CASE      = Test.....@
set INPT       = $HOME/input.dat.....@

```

```

set ODR      = $MOSRA_DIR/smpl/outp.....(d)
set MACRO    = $MOSRA_DIR/smpl/pds/MACRO.....(e)
set EGG      = $MOSRA_DIR/smpl/pds/EGG-$CASE.....(f)
set LM       = $MOSRA_DIR/bin/MOSRA-L.lm.....(g)
#
#===== Change if you like =====
#
set DATE     = `date +%b%d.%H.%M.%S`
set WKDR     = $HOME/MOSRAtmp.$CASE.$DATE.....(h)
mkdir $WKDR
#
if (! -e $MACRO) then
  mkdir $MACRO
endif
if (! -e $EGG) then
  mkdir $EGG
endif
#
#===== File Allocation =====
#
# setenv MACRO2   $HOME/MOSRA-Light/smpl/pds/MACRO
# setenv EGG2     $HOME/MOSRA-Light/smpl/pds/EGG
setenv MACRO1  $MACRO.....(j)
setenv EGG1    $EGG.....(k)
setenv fu99    $ODR/$CASE.MFT99.$DATE.....(l)
set OUTLST   = $ODR/$CASE.MFT06.$DATE.....(m)
#
#===== Execution of MOSRA-Light code =====
#
cd $WKDR
$LM < $INPT >& $OUTLST.....(n)
#
#===== Remove scratch files =====
#
cd $HOME
rm -r $WKDR.....(o)
#
#===== Remove PDS files, if you do not need =====
#
# Egg & MACRO files are necessary for the utility code More-MOSRA.
# Keep them if necessary. ]
# rm -r $MACRO
# rm -r $EGG
#
#=====

```

(a) MOSRA-Lightコードシステムのトップディレクトリの所在

(b) ジョブのケース名(任意)の指定。

ジョブのケース名は作業用ディレクトリや出力ファイル名の一部として参照される(①, ②, ③, ④ 参照)。

- ⑤ 標準入力データファイルの指定
- ⑥ 計算結果(テキストファイル)を収納する既存ディレクトリの指定
- ⑦ MACROファイル(ディレクトリ)の指定
 

MACROファイルは、MOSRA-Lightが使用するバイナリー断面積データを複数のサブファイル(メンバー)として収納するディレクトリである(第7.2節、第7.3節参照)。SRACコードで作成したMACROファイルをここで指定することができる。断面積データをMOSRA-Lightの標準入力から与える場合にも、ここで指定するMACROファイルの中にメンバーが作成される。
- ⑧ Eggファイル(ディレクトリ)の指定
 

Eggファイルは、MOSRA-Lightの計算結果(バイナリーデータ)を複数のメンバーとして収納するディレクトリである(第7.4節参照)。得られたEggファイルは、別のMOSRA-Lightの計算やユーティリティコードMore-MOSRA(第4.2.16節参照)に利用することができる。
- ⑨ 使用するMOSRA-Lightのロードモジュールの指定
- ⑩ MOSRA-Lightが実行される作業用ディレクトリの指定
 

ワークステーション(EWS)などでは、ジョブの実行によりスクラッチファイル(例えば`fort.機番`)がここで指定するディレクトリに残される。上記の例では、複数のジョブの実行によりスクラッチファイルの干渉が生じないように、ジョブのケース名と時刻を付けた名前のディレクトリをユーザーのホームディレクトリ上に作成している。ジョブが正常に終了した時点でこのディレクトリは⑪において消去される。なお、共有の大型計算機では、スクラッチファイルの命名や消去をシステムが適切に管理するものが多く、このような場合には⑩及び⑪は特に意味を持たない。
- ⑪ MACROファイル及びEggファイルの新規作成
 

⑩及び⑪で指定するMACROまたはEggファイルが存在しない場合には、ここで新規にディレクトリが作成される。上記の例では、EggファイルはMOSRA-Light/smpl/pds配下にEGG-Testとして作成される。
- ⑫ MACROファイルに対する環境変数の指定
 

上記の例では`setenv`コマンドにより環境変数MACRO1に⑪で指定するMACROファイルの所在(フルパス)を定義している。これはMOSRA-Lightが使用する断面積ファイルのアロケーションの役割を果たす。MOSRA-Lightの標準入力では、使用するMACROファイルに対応する環境変数名(例えばMACRO1)を計算ケース毎に入力する(第5.3.2節のBlock-

2-5参照)。従って、ここで指定する環境変数名はユーザーの任意である。また、MOSRA-Lightの繰り返し計算機能により複数の計算ケースを異なるMACROファイルを使用して実行する場合には、下記に示すように複数のMACROファイルのパスを予め異なる環境変数に定義しておくと良い。

```
例： setenv MACRO1 $HOME/smpl/pds/MACRO-2G
      setenv MACRO2 $HOME/smpl/pds/MACRO-4G
      setenv MACRO3 $HOME/smpl/pds/MACRO-8G
```

#### ⑯ Eggファイルに対する環境変数の指定

MACROファイルと同様に、バイナリー出力ファイル(メンバー)の収納先となるEggファイルのパスを環境変数に定義する。Eggファイルに対する環境変数名もMOSRA-Lightの標準入力で計算ケース毎に指定するため(第5.3.2節のBlock-2-6参照)、ここで指定する環境変数名はユーザーの任意である。また、複数のEggファイルのパスを定義しても良い。

#### ① 計算結果のプリント出力ファイルの指定

MOSRA-Lightの主要な計算結果は99番のファイルに出力される。①は99番にプリント出力されるテキストデータを実際のファイルに割り当てるための定義である。上記の例では、例えばTest.MFT99.Feb09.11.22.33(日本語利用環境では表示が異なる)と言う名前のファイルが①で指定するディレクトリ配下に作成される。最後の部分にジョブ実行時の日付と時刻(2月9日11時22分33秒)を付けているが、出力ファイル名の命名はユーザーの自由である。

なお、何らかの理由で99番が利用できない計算機においては、MOSRA-Lightのメインプログラムのみの修正で容易に使用する機番を変更することができる。このような場合には、環境変数名をfu##(##は変更した機番)とする。

#### ⑮ 標準出力ファイル(6番)の指定

標準出力(6番)には、入力データのエコー、MOSRA-Lightの実行ステップのモニタリング情報、エラーメッセージなどがプリント出力される。エラーが発生した場合には、先ずこちらの出力結果を見ると良い。

#### ⑯ ロードモジュールの実行

カレントディレクトリを⑯で指定するディレクトリに移し、MOSRA-Lightを実行させる。実行時間の計測コマンド(timeまたはtimex)が利用できる場合には、⑯を下記のように変更するとMOSRA-Light実行の経過時間、システム消費時間などの情報を出力することができる。

```
例： timex $LM < $INPT >& $OUTLST
```

## ⑥ スクラッチファイルの消去

⑪で作成したスクラッチファイル用のディレクトリを消去する。MOSRA-Lightで使用するスクラッチファイルは95番(標準入力の複製)のみである。

## ⑦ MACROファイルとEggファイルの消去

MOSRA-Lightの実行が終了すると、⑥及び⑪で指定したディレクトリ配下に複数のメンバーファイルが作成される。上記の例ではコメント化してあるが、MACROまたはEggファイルが不要であるならば、これらを最後に消去しておく。

ユーザー好みの問題ではあるが、MOSRA-Lightシステムに添付してあるサンプル入データは全て実行用シェルスクリプトと一緒にしてある。例えば次のようなものである。

## 【シェルスクリプト例2】

```
#!/bin/csh
#
#----- Execution of MOSRA-Light code -----
#
cd $WKDR
cat - << END_DATA | $LM >& $OUTLST.....@  

%TIMESET  
10.0          / CPULIM (min.)  

%CONTROL  

TEST  

IAEA 3-D BENCHMARK PROBLEM (CASE0:20CM*20CM*20CM), NODE:1311  

EIGEN VALUE PROBLEM, 2-GROUPS, WITH DYADONAL SYMMETRIC OPTION  

MACRO1          / PDS NAME FOR MACRO XS FILE IN SHELL-SCRIPT  

EGG1           / PDS NAME FOR MOSRA OUTPUT FILE IN SHELL-SCRIPT  

2 1            / NG, NGT  

0 0 0 1 0 1 0 0 0 0 0 2 0 0 5(0) / GENERAL CONTROL  

6 1 0 4 0 0 1 0 0 0           / PRINT CONTROL  

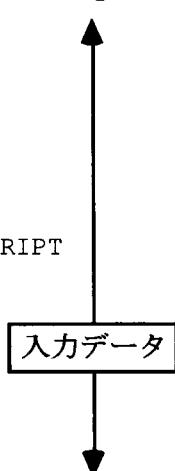
  : (省略)  

%CONTROL  

      / END OF ALL INPUT DATA  

END_DATA.....@r  

#
#----- Remove scratch files -----
#
cd $HOME
rm -r $WKDR
#
#----- Remove PDS files, if you do not need -----
```



```
#  
# Egg & MACRO files are necessary for the utility code More-MOSRA.  
# Keep them if necessary. ] ..... (P)  
# rm -r $MACRO  
# rm -r $EGG  
#  
#=====
```

シェルスクリプト例1との違いは④から⑤までの部分である。④の部分では、⑤のEND\_DATA手前までの入力データ部分をcatコマンドにより読み書きし、パイプ処理によってMOSRA-Lightロードモジュールの標準入力として与えている。この様にすることにより実行用シェルスクリプトと入力データを1つのファイルに保存しておくことができ、実行時のファイル割り当てや使用したロードモジュール名などの記録として便利である。ただし、この場合に注意すべき事は、入力データ部分のコメントなどに文字'\$'を使用してはならないことである。これはシェルスクリプトの中では\$から始まる文字列はシェル変数と見なされるためである。この場合、シェル変数が未定義であるとしてスクリプトエラーとなる。

## 6.2 入力データの例

本節では、ユーザーが入力データを作成する上で参考となるように、幾つかの具体的な問題について入力データの例を示す。

### 6.2.1 IAEA3次元ベンチマーク

IAEAの3次元ベンチマーク問題<sup>24)</sup>については第3.5.1節において述べた通りである。以下の図表に問題の仕様や結果がまとめられている。

計算体系	: Fig.3.5.1
2群断面積	: Table 3.5.1
参照解及びMOSRA-Lightの計算結果	: Fig.3.5.2

なお、問題の仕様には拡散計算に必要な生成断面積( $v\Sigma_f$ )は与えられているが、核分裂断面積( $\Sigma_f$ )は与えられていない。本来、出力分布は(4.2.78)式のように核分裂断面積を使用して計算すべきであるが、ベンチマーク問題では便宜的に生成断面積を使用して次の式で評価することになっている。

$$q'''^k = \sum_{g=1}^G v\Sigma_{f,g}^k \Phi_{g,0}^k \quad (6.2.1)$$

主な出力項目には、中性子増倍率の他に軸方向に平均化された燃料集合体(I)毎の出力分布 $\bar{q}'''^I$ があり、これを炉心全体の平均値が1.0になるように規格化して出力する。

$$\bar{q}'''^I = \frac{1}{V^I} \sum_{k \in I} q'''^k V^k = \frac{1}{V^I} \sum_{k \in I} \sum_{g=1}^G v\Sigma_{f,g}^k \Phi_{g,0}^k V^k \quad (6.2.2)$$

この問題に対するMOSRA-Lightの入力データ例を以下に示す。これは、4次のNEMを使用して各方向のメッシュ幅を20cmに設定した計算例である(反射境界面に接するノードは10cm)。また、対角対称境界条件を使用し実際の計算は1/8炉心で行っている。更にEdit-Regionのオプションを使用して燃料集合体単位の出力分布を編集している。

#### 【入力データ例1】

```
.....
%TIMESET
10.0          / CPULIM (min.)
%CONTROL
TEST
IAEA 3-D BENCHMARK PROBLEM (CASE0:20CM*20CM*20CM), NODE:1311
EIGEN VALUE PROBLEM, 2-GROUPS, WITH DYADONAL SYMMETRIC OPTION
```

```

MACRO1          / PDS NAME FOR MACRO XS FILE IN SHELL-SCRIPT
EGG1           / PDS NAME FOR MOSRA OUTPUT FILE IN SHELL-SCRIPT
2 1           / NG, NGT
0 0 0 1 0  1 0 0 0 0  0 0 2 0 0  5(0) / GENERAL CONTROL
6 1 0 4 0  0 1 0 0 0           / PRINT CONTROL
-1.0          / Power(W) IF -1.0, Power = Fissile Volume
&   POWER =  0.0 : NO NORMALIZATION (PRODUCTION RATE = K-EFF)
&   POWER = -1.0 : NORMALIZED TO FISSILE VOLUME (POWER FORM FACTOR)
&   POWER = -2.0 : NORMALIZED TO ABSORPTION RATE = 1.0
&   POWER = -3.0 : NORMALIZED TO FISSION RATE = 1.0
&   POWER = -4.0 : NORMALIZED TO PRODUCTION RATE = 1.0
%GEOMSET      / GEOMETRY INPUT SECTION
9 9 19          / NX, NY, NZ
2 1 2 1 1 1 -1           / B.C. (X-, X+, Y-, Y+, Z-, Z+), IBCD(7)<0:Diagonal Sym.
10.0 8*20.0        / XXX
10.0 8*20.0        / YYY
20.0 18*20.0        / ZZZ
1( 4 4 4 4 0 0 0 0 0 0    & PLANE-1( BOTTOM REFLECTOR)
 4 4 4 4 4 4 0 0 0
 4 4 4 4 4 4 4 0 0
 4 4 4 4 4 4 4 4 0
 4 4 4 4 4 4 4 4 0
 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 )
13( 4 4 4 4 0 0 0 0 0 0    & PLANE-2-14 (NO PARTIAL INSERTED CR)
 1 1 1 4 4 4 0 0 0
 2 2 1 1 1 4 4 0 0
 2 2 2 2 1 1 4 4 0
 3 2 2 2 3 1 1 4 0
 2 2 2 2 2 2 1 4 4
 2 2 2 2 2 2 1 1 4
 2 2 2 2 2 2 2 1 4
 3 2 2 2 3 2 2 1 4 )
4( 4 4 4 4 0 0 0 0 0 0    & PLANE-15-18 (PARTIAL INSERTED CR)
 1 1 1 4 4 4 0 0 0
 2 2 1 1 1 4 4 0 0
 2 2 2 2 1 1 4 4 0
 3 2 2 2 3 1 1 4 0
 2 2 2 2 2 2 1 4 4
 2 2 3 2 2 2 1 1 4
 2 2 2 2 2 2 2 1 4
 3 2 2 2 3 2 2 1 4 )
1( 4 4 4 4 0 0 0 0 0 0    & PLANE-19(TOP REFLECTOR)
 4 4 4 4 4 4 0 0 0
 4 4 4 4 4 4 4 0 0
 4 4 4 4 4 4 4 4 0
 5 4 4 4 5 4 4 4 4 0
 4 4 4 4 4 4 4 4 4 4
 4 4 5 4 4 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4
 5 4 4 4 5 4 4 4 4 4 )
%MATNAME
 5 / NMAT (Number of Materials whose XSs are read)
MAT1A010  0 0 0 0 0.0 / 1 : Outer Fuel
MAT2A010  0 0 0 0 0.0 / 2 : Inner Fuel

```

```

MAT3A010 0 0 0 0 0.0 / 3 : Inner Fuel + Control Rod
MAT4A010 0 0 0 0 0.0 / 4 : Reflector
MAT5A010 0 0 0 0 0.0 / 5 : Reflector + Control Rod
%XSINPUT
0.0 0.0 0.01 1.0 1.5 1.5 0.1922 0.02
0.135 0.0675 0.08 0.0 0.4 0.4 0.0 0.7533 / SIGP,F,A,X,D1,D2,S MAT1
0.0 0.0 0.01 1.0 1.5 1.5 0.1922 0.02
0.135 0.0675 0.085 0.0 0.4 0.4 0.0 0.7483 / SIGP,F,A,X,D1,D2,S MAT2
0.0 0.0 0.01 1.0 1.5 1.5 0.1922 0.02
0.135 0.0675 0.13 0.0 0.4 0.4 0.0 0.7033 / SIGP,F,A,X,D1,D2,S MAT3
0.0 0.0 0.0 0.0 2.0 2.0 0.1267 0.04
0.0 0.0 0.01 0.0 0.3 0.3 0.0 1.1011 / SIGP,F,A,X,D1,D2,S MAT4
0.0 0.0 0.0 0.0 2.0 2.0 0.1267 0.04
0.0 0.0 0.055 0.0 0.3 0.3 0.0 1.0561 / SIGP,F,A,X,D1,D2,S MAT5
%EDITRGN
38 / NERGN WHEN IOPT(6)=1 : NUMBER OF E-REGION
1( 0 0 0 0 0 0 0 0 & IERGN FOR PLANE-1
  0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 )
17( 0 0 0 0 0 0 0 0 & IERGN FOR PLANE-2-18
  0 0 0 0 0 0 0 0
  0 0 0 0 0 38 0 0
  0 0 0 0 0 35 36 37 0
  0 0 0 0 31 32 33 34 0
  0 0 0 25 26 27 28 29 30
  0 0 18 19 20 21 22 23 24
  0 10 11 12 13 14 15 16 17
  1 2 3 4 5 6 7 8 9 )
1( 0 0 0 0 0 0 0 0 & IERGN FOR PLANE-19
  0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 )
%CONTROL
/ END OF ALL CASE
.....
```

## 補足：

核分裂断面積の入力において、 $v$ 値に2.0を仮定して  $\Sigma_{f,g} = v \Sigma_{f,g} / 2.0$  としているが、規格化される出力分布には影響しない。

自群散乱断面積は拡散計算には使用されないため任意ではあるが、ここでは便宜的に次の式により与えている。

$$\Sigma_{s,g \rightarrow g} = \Sigma_{t,g} - \sum_{g \neq g'} \Sigma_{s,g \rightarrow g'} \approx \frac{1}{3D_g} - \sum_{g \neq g'} \Sigma_{s,g \rightarrow g'} \quad (6.2.3)$$

### 6.2.2 IAEA2次元ベンチマーク

IAEAの2次元ベンチマーク問題<sup>24)</sup>は前節の3次元問題を2次元化した問題である。計算体系はFig.3.5.1の3次元炉心において、軸方向中心位置( $z=190\text{cm}$ )で水平断面をとったX-Y体系である(Fig.6.2.1参照)。また、軸方向への中性子の漏洩はバックリングで扱い、全群についてバックリング値は一定( $B^2=8.0E-5\text{cm}^{-2}$ )とする。2群断面積は3次元問題と同じもの(Table 3.5.1)を使用する。

MOSRA-Lightは2次元や1次元問題に特化したプログラムを有していないため、この問題に対しては $z$ 軸方向に少なくとも1ノードをとり、その両側( $z\pm$ 面)に反射境界条件を与えて扱う。この場合 $z$ 方向のメッシュ幅は任意であるが、安定な収束を期待するならば他方向のメッシュ分割幅とほぼ同等な長さを与える方が好ましい。これは、1つの方向に突出したノードはその応答行列の性質が悪くなるためである。ここでは、反射境界面に接するノードを除きメッシュ幅を20cmとする4次のNEMによる計算例を示す。

このベンチマーク問題自体は単純なものであるが、ここではMOSRA-Lightの入力オプションの具体的な利用法を説明するため、問題を拡張して以下の2ケースの計算を連続して行うものとする。

#### ケース名：STND(標準ケース)

- ・ 本来の2次元ベンチマーク問題
- ・ 1/4炉心体系(敢えて対角対称境界条件は使用しない)
- ・ 通常の中性子束計算(forward flux)の後で、オプションとして随伴中性子束も計算させる。
- ・ Edit-Regionオプションを使用して集合体出力分布を編集する。この際、対角対称な位置にある集合体には同じEdit-Region番号を与える。

#### ケース名：A150(摂動ケース)

- ・ 制御棒挿入領域の熱群吸収断面積をもとの値の1.5倍とする。
- ・ 通常の中性子束計算と随伴中性子束を計算する。
- ・ 前ケース(STND)と断面積は変わるが幾何形状は同じため、オプションにより幾何形状の入力をスキップさせる。(第5.3.3節 Block-3-2参照)。
- ・ 前ケース(STND)とEdit-Regionの指定は同じで良いため、オプションによりEdit-Region番号の入力をスキップさせる。(第5.3.7節 Block-7-2参照)。

この計算では固有値計算を合計4回行うことになる。以下はMOSRA-Lightの入力例である。

## 【入力データ例2】

```

%TIMESET
10.0          / CPULIM (min.)
%CONTROL
STND
IAEA 2-D BENCHMARK PROBLEM WITH BUCKLING (20CM*20CM*20CM)
No Perturbation Case
MACRO          / PDS NAME FOR MACRO XS FILE IN SHELL-SCRIPT
EGG            / PDS NAME FOR MOSRA OUTPUT FILE IN SHELL-SCRIPT
2 1            / NG, NGT
0 0 2 1 0  1 0 1 0 2  0 0 -1 2 0  5(0)  / GENERAL CONTROL
6 1 0 4 0  0 0 0 0 0  / PRINT CONTROL
1.0E-5 9(0.0)  / EPSCV
-1.0          / Power(W) IF -1.0, Power = Fissile Volume
%GEOMSET      / GEOMETRY INPUT SECTION
9 9 1          / NX, NY, NZ
2 1 2 1 2 2 0  / B.C.(X-, X+, Y-, Y+, Z-, Z+), IBCD(7)<0:Diagonal Sym.
10.0 8*20.0   / XXX
10.0 8*20.0   / YYY
20.0          / ZZZ (DUMMY NODE FOR 2-D CALCULATION)

4 4 4 4 0 0 0 0 0
1 1 1 4 4 4 0 0 0
2 2 1 1 1 4 4 0 0
2 2 2 2 1 1 4 4 0
3 2 2 2 3 1 1 4 0
2 2 2 2 2 2 1 4 4
2 2 2 2 2 2 1 1 4
2 2 2 2 2 2 2 1 4
3 2 2 2 3 2 2 1 4  / IDMAT

%MATNAME
4 / NMAT (Number of Materials whose XSs are read)
STNDA010 0 0 0 0 0.0 / 1 : Outer Fuel
STNDA020 0 0 0 0 0.0 / 2 : Inner Fuel
STNDA030 0 0 0 0 0.0 / 3 : Inner Fuel + Control Rod
STNDA040 0 0 0 0 0.0 / 4 : Reflector

%XSINPUT
0.0 0.0 0.01 1.0 1.5 1.5 0.1922 0.02
0.135 0.0675 0.08 0.0 0.4 0.4 0.0 0.7533 / SIGP,F,A,X,D1,D2,S MAT1
0.0 0.0 0.01 1.0 1.5 1.5 0.1922 0.02
0.135 0.0675 0.085 0.0 0.4 0.4 0.0 0.7483 / SIGP,F,A,X,D1,D2,S MAT2
0.0 0.0 0.01 1.0 1.5 1.5 0.1922 0.02
0.135 0.0675 0.13 0.0 0.4 0.4 0.0 0.7033 / SIGP,F,A,X,D1,D2,S MAT3
0.0 0.0 0.0 0.0 2.0 2.0 0.1267 0.04
0.0 0.0 0.01 0.0 0.3 0.3 0.0 1.1011 / SIGP,F,A,X,D1,D2,S MAT4

%EDITRGN
38     / NERGN WHEN IOPT(6)=1 : NUMBER OF E-REGION
9 17 24 30 0 0 0 0 0
8 16 23 29 34 37 0 0 0
7 15 22 28 33 36 38 0 0
6 14 21 27 32 35 36 37 0
5 13 20 26 31 32 33 34 0
4 12 19 25 26 27 28 29 30
3 11 18 19 20 21 22 23 24
2 10 11 12 13 14 15 16 17
1 2 3 4 5 6 7 8 9  / IERGN WHEN IOPT(6)=1

%BUCKLNG

```

```

2(0.8E-4)      / AXIAL BUCKLING BY GROUP WHEN IOPT(8)=1
%CONTROL
A150
IAEA 2-D BENCHMARK PROBLEM WITH BUCKLING (20CM*20CM*20CM)
Perturbed Case (Thermal Absorption of Matterial3 => *1.5)
MACRO          / PDS NAME FOR MACRO XS FILE IN SHELL-SCRIPT
EGG            / PDS NAME FOR MOSRA OUTPUT FILE IN SHELL-SCRIPT
2 1            / NG, NGT
0 0 2 1 0  1 0 1 0 2  0 0 -1 2 0  5(0)  / GENERAL CONTROL
0 1 0 4 0  0 0 0 0 0  / PRINT CONTROL
1.0E-5 9(0.0)  / EPSCV
-1.0           / Power(W) IF -1.0, Power = Fissile Volume
%GEOMSET       / GEOMETRY INPUT SECTION
0 0 0          / NX, NY, NZ : Geometry is same to that of the previous case
%MATNAME
4 / NMAT (Number of Materials whose XSs are read)
STNDA010  0 0 0 0  0.0 / 1 : Outer Fuel
STNDA020  0 0 0 0  0.0 / 2 : Inner Fuel
A150A030  0 0 0 0  0.0 / 3 : Inner Fuel + Contorl Rod
STNDA040  0 0 0 0  0.0 / 4 : Reflector
%XSINPUT
-1.0 7(0.0) / MAT1 (Member already existing in MACRO)
8(0.0)      / dummy for 2nd group of MAT1
-1.0 7(0.0) / MAT2 (Member already existing in MACRO)
8(0.0)      / dummy for 2nd group of MAT2
& Thermal Absorption XS of MAT3 is multiplied by 1.5 (0.13=>0.195)
0.0 0.0 0.01 1.0 1.5 1.5 0.1922 0.02
0.135 0.0675 0.195 0.0 0.4 0.4 0.0 0.7033 / SIGP,F,A,X,D1,D2,S MAT3
&
-1.0 7(0.0) / MAT4 (Member already existing in MACRO)
8(0.0)      / dummy for 2nd group of MAT4
%EDITRGN
0          / NERGN=0 : E-Region is same to that of the previous case
%BUCKLNG
2(0.8E-4)    / AXIAL BUCKLING BY GROUP WHEN IOPT(8)=1
%CONTROL
/ END OF ALL CASE
.....
```

**補足：**

上記の例では2ケースの計算を行っているが、ケース名にブランクを与えるまで何度も繰り返し計算を行うことができる。

MACROファイルに書き出された断面積メンバーは、別のMOSRA-Lightによる計算やSRAC-CITATIONによる計算などに利用することができる(第7.3節参照)。

両計算ケースによりEggファイルに書き出されるメンバーを使用して、ユーティリティコード(More-MOSRA)により、1次摂動または厳密摂動の計算を行うことができる。ただし、More-MOSRAにより摂動計算を行うには、ノード毎の断面積をEggファイルに出力するオプション(IOPT(14)=2/6)を指定しておく必要がある。

MOSRA-LightによるIAEA2Dベンチマークの計算結果(ケース名STND)は、VENTUREコードによる参考解(無限小メッシュ外挿値)<sup>24)</sup>と比較してFig.6.2.2に示してある。

### 6.2.3 多群格子計算

これまで、2群断面積を標準入力として与えるベンチマーク計算の例を示した。しかし、エネルギー群数や物質数(異なる断面積セット数)が多い場合には、標準入力から断面積を与えるのは不便である。ここでは、40群の2次元格子計算を例として、SRACコードにより作成される巨視的断面積セット(バイナリーデータ)を利用する例を示す。断面積を指定するには、物質を代表する8文字のメンバー名を指定するだけで良い。MACROファイルのバイナリーデータの形式をSRACコードの形式(第7.3節参照)に合わせれば、SRAC以外のコードにより作成される断面積もMOSRA-Lightに利用することができる。

計算体系をFig.6.2.3に示す。これはピッチ1.26cmの燃料ピンセルで構成される30×30(計算上は15×15)の無限格子体系で、中心部には高富化度のMOX燃料が配置され、そのまわりをUO<sub>2</sub>燃料が取り囲んでいる。この体系に対する中性子増倍率と出力分布(Fig.6.2.3のA-MとA-Dの位置)を計算する。

この問題はOECD/NEA/NSCのWPPR (the Working Party on Physics of Plutonium Recycling)において取り上げられたベンチマーク問題<sup>34)</sup> の一部である。MOX燃料セルとUO<sub>2</sub>燃料セルの40群均質化断面積はSRACコードの107群衝突確率計算により作成する。以下はSRACコードの入力データの一例である。入力の詳細については、文献25を参照されたい。

#### 【SRACコード入力データ例】

```

HMXN
NEACRP PWR BENCHMARK PROBLEM . ZERO BUCKLING : H-MOX CELL : UNVOIDED
1 1 1 1 2   1 4 3 -2 1   0 0 0 0 2   0 1 0 0 0
1.0000E-30
/home/okumura/SRAC/SRACLIB-JDL32/pds/pfast   Old   File
/home/okumura/SRAC/SRACLIB-JDL32/pds/pthml   O     F
/home/okumura/SRAC/SRACLIB-JDL32/pds/pmcrs   O     F
/home/okumura/WPPR/UFAST                      Scratch Core
/home/okumura/WPPR/UTHERMAL                   S     C
/home/okumura/WPPR/UMCROSS                    S     C
/home/okumura/WPPR/MACROWRK                  S     C
/home/okumura/MACRO-SRAC                     New   C
/home/okumura/WPPR/FLUX                       S     C
/home/okumura/WPPR/MICREF                     S     C
61 46 17 23 / CONDENSED TO 40-GROUP
61(1)      /
46(1)      /
9(5) 8(2)   / CONDENSED FAST     17 GROUP STRUCTURE
23(2)      / CONDENSED THERMAL 23 GROUP STRUCTURE

4 7 7 3 1   1 7 0 0 0   5 0 8 15 0   0 45 0       / PIJ CONTROL
0 50 50 5 5 0 0.0001 0.00001 0.001 1.0 10. 0.5 /
1 1 1 2 3 3 3 / R-T
3(1)      / X-R
1 2 3      / M-R
0.0 0.23642 0.33436 0.40950 0.47500 0.56468 0.641956 0.63000 / RX

```

```

3      / NMAT
PIN1XP1X  0  9   300.0   0.8190  0.0  / 1 MOX-FUEL
XU050001  2  0   7.90000E-05
XU080001  2  0   1.92000E-02
XPU80001  2  0   4.90000E-05
XPU90001  2  0   1.99000E-03
XPU00001  2  0   7.53000E-04
XPU10001  2  0   2.06000E-04
XPU20001  2  0   1.40000E-04
XAM10001  2  0   9.83000E-05
XO060001  0  0   4.50000E-02
CLD1XC1X  0  2   300.0   0.1310  0.0  / 2 CLADDING
XZRN0001  0  0   3.68000E-02
XFEN0001  0  0   7.23000E-05
MOD1XM1X  0  3   300.0   1.000   0.0  / 3 MODERATOR (0%VOID)
XO060001  0  0   2.47000E-02
XH01H001  0  0   4.94000E-02
XAL70001  0  0   3.31000E-03
0      / PEACO

```

**UO2N**

NEACRP PWR BENCHMARK PROBLEM . ZERO BUCKLING : UO2 CELL : UNVOIDED

```

1 1 1 1 2   1 4 3 -2 1   0 0 0 0 2   0 1 0 0 0
1.0000E-30

```

```

4 7 7 3 1   1 7 0 0 0   5 0 8 15 0   0 45 0       / PIJ CONTROL

```

```

0 50 50 5 5 5 0   0.0001 0.00001 0.001 1.0 10. 0.5 /

```

```

1 1 1 2 3 3 3 / R-T

```

```

3(1) / X-R

```

```

1 2 3 / M-R

```

```

0.0 0.23642 0.33436 0.40950 0.47500 0.56468 0.641956 0.63000 / RX

```

```

3      / NMAT

```

```

PIN2XP2X  0  3   300.0   0.8190  0.0  / 1 UO2-FUEL

```

```

XU050001  2  0   7.65000E-04

```

```

XU080001  2  0   2.21000E-02

```

```

XO060001  0  0   4.57000E-02

```

```

CLD2XC2X  0  2   300.0   0.1310  0.0  / 2 CLADDING

```

```

XZRN0001  0  0   3.68000E-02

```

```

XFEN0001  0  0   7.23000E-05

```

```

MOD2XM2X  0  3   300.0   1.000   0.0  / 3 MODERATOR (0%VOID)

```

```

XO060001  0  0   2.47000E-02

```

```

XH01H001  0  0   4.94000E-02

```

```

XAL70001  0  0   3.31000E-03

```

```

0      / PEACO

```

.....

上記のSRACによる計算の結果、MACROファイル(/home/okumura/MACRO-SRAC)が作成され、配下には以下のような7個の断面積メンバーが含まれる(第7.3節参照)。

```

CONTA000
HMXNA010   HMXNA01N   HMXNA01Z
UO2NA010   UO2NA01N   UO2NA01Z

```

このMACROファイルを利用してMOSRA-Lightによる40群拡散計算を実施する。以下は

MOSRA-Lightの入力データ例である。

### 【入力データ例3】

```

0 16 0 0 0 0 0 0 0 0 0 0 0 0 0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 / IERGN
%CONTROL
    / END OF ALL CASE
.....

```

### 6.2.4 その他の入力例

前節までに3つのMOSRA-Lightの入力例を紹介したが、MOSRA-Lightのファイルの中(MOSRA-Light/smpl/shr配下)にはこれら以外の入力例も収納している。何れもシェルスクリプトと一体化させた入力データ(第6.1節参照)としており、MOSRA-Lightをインストールしたユーザーは、ほとんど内容を変更することなく実行することができる。ここでは、入力例の一覧と概略説明だけを述べる(例題を特徴付ける部分を下線で示した)。同種の問題の入力を作成する場合に参考にして頂きたい。

#### (1) Test.sh (動作確認用)

- IAEA3Dベンチマーク問題(第6.2.1節の例1と同じ)
- 2群固有値計算、メッシュ幅20cm、対角対称オプション
- 断面積は標準入力
- 他のIAEA3次元ベンチマーク計算のためMACROファイルが保存される。

#### (2) IAEA3D1.sh

- IAEA3Dベンチマーク問題(メッシュ効果)
- 2群固有値計算、メッシュ幅20cm、対角対称オプションなし
- 断面積はファイル読み込み(Test.shで作成されるMACROファイルを使用するため、Test.shを先に実行しておく必要がある)。

#### (2) IAEA3D2.sh

- IAEA3Dベンチマーク問題(メッシュ効果)
- 2群固有値計算、メッシュ幅10cm、対角対称オプションなし
- Test.shで作成されるMACROファイルを使用する。

#### (3) IAEA3D4.sh

- IAEA3Dベンチマーク問題(メッシュ効果)
- 2群固有値計算、メッシュ幅5cm、対角対称オプションなし
- Test.shで作成されるMACROファイルを使用する。
- リスタート計算用のEggファイルを保存する。
- スカラーアルゴリズムでは、かなりの計算時間を必要とする(計算時間の上限を変更すること)。

## (4) IAEA3D4R.sh

- IAEA3D4.shのリスタート計算
- Test.shで作成されるMACROファイルを使用する。
- IAEA3D4.shで作成されるEggファイルを使用する。

## (5) IAEA2D.sh

- IAEA2Dベンチマーク問題(第6.2.2節の例2と同じ)
- メッシュ幅20cm、バックリング使用の2次元計算
- 2群中性子束と随伴中性子束の計算
- 非摂動体系(標準ケース)と摂動体系の連続ケース計算
- More-MOSRAによる摂動計算に必要なメンバーをEggファイルに出力する。

## (6) FixSou.sh

- IAEA3Dベンチマーク問題(IAEA3D1.sh)をベースとした固定源問題
- 未臨界体系とするため、全ての核分裂断面積をもとの値の半分としている。

## (7) Black.sh

- IAEA2Dベンチマーク問題において、制御棒挿入領域を内部黒体の扱いとした問題
- 内部黒体では第1群を断面積で扱い、第2群のみを完全吸収黒体として扱う。

## (8) MultG.sh

- OECD/NEA/NSC(WPPR)のベンチマーク問題に対する多群2次元格子計算(第6.2.3節の例3と同じ)
- 熱群反復計算を含む40群(高速17群+熱群23)計算
- SRACコードにより作成される断面積ファイル(MACRO)を使用する。MACROファイルはSRAC.shにより予め作成しておく。

## (9) SRAC.sh

- MultG.shに使用する巨視的断面積をSRACコードの衝突確率計算により作成する。
- 衝突確率計算：107群正方格子モデル、実効共鳴断面積の計算：超詳細群衝突確率法
- 2種類の正方格子( $UO_2$ セル、MOXセル)に対する格子内で均質化した40群縮約断面積をMACROファイルに出力する。
- JENDL-3.2ライブラリー使用

以上のサンプル問題に対して、ベクトル及びスカラーアルゴリズムに必要とされる計算コスト(計算時間、可変配列長)の例を、付録-B.3のTable B.3.1に示してある。

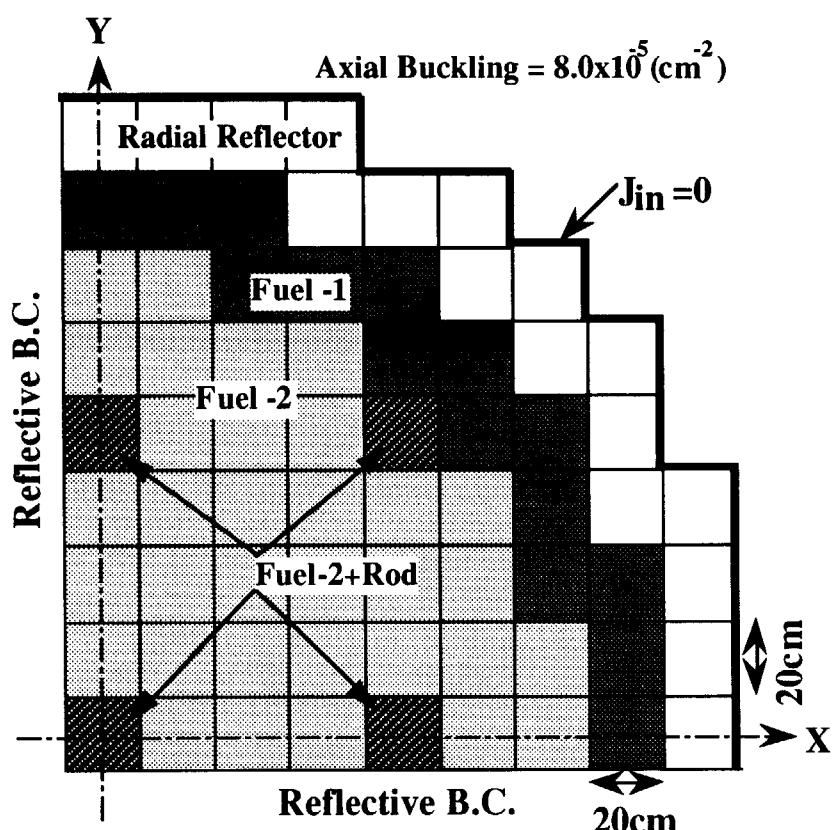


Fig.6.2.1 Core geometry of IAEA 2-dimensional benchmark problem

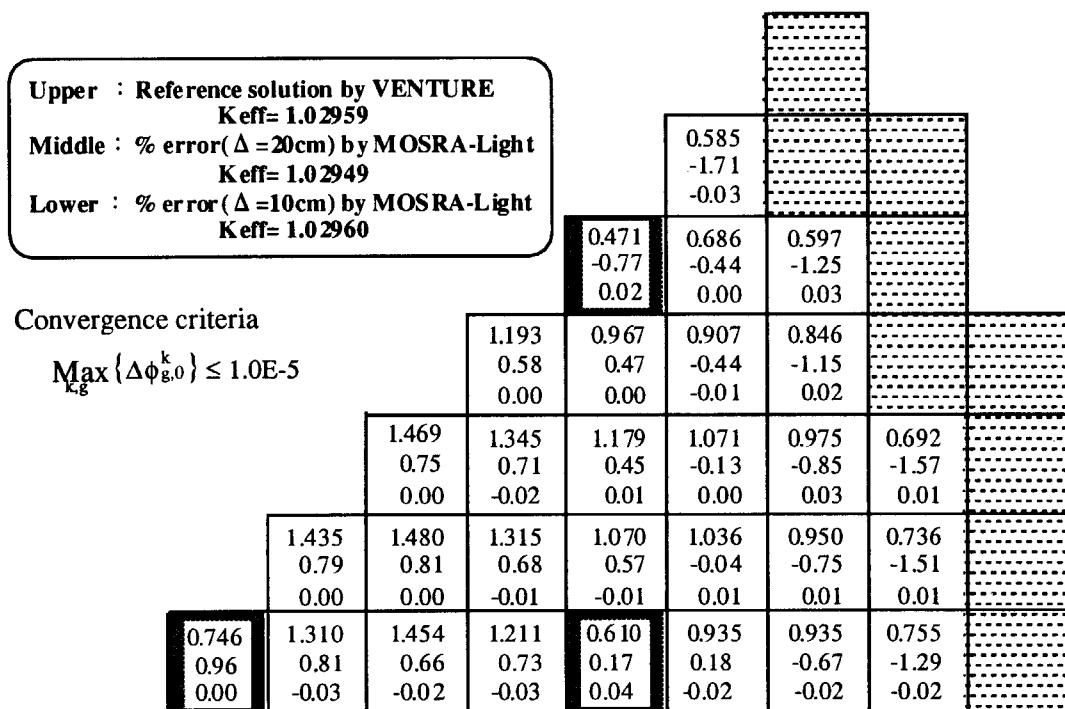


Fig.6.2.2 Comparison of k-eff and assembly power distribution in IAEA 2D benchmark

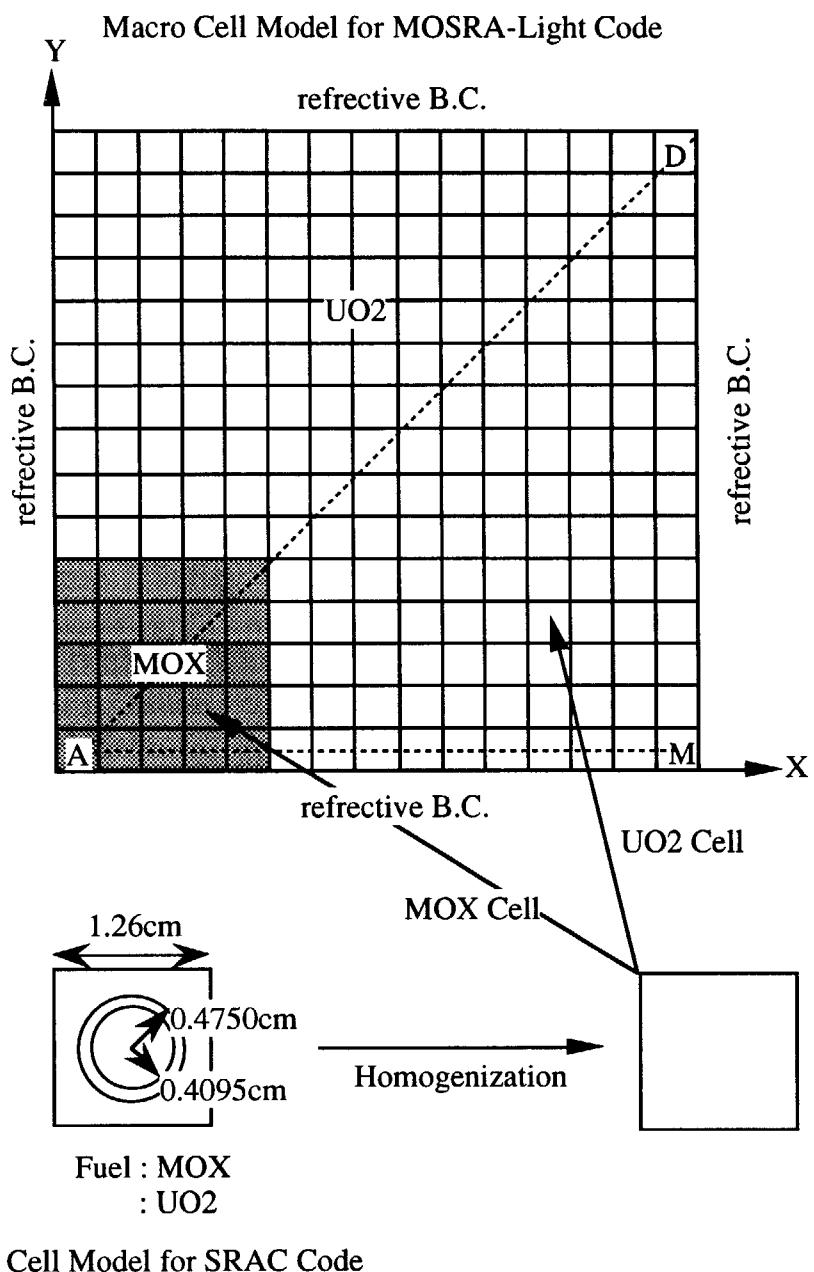


Fig.6.2.3 Geometry of multi-group lattice calculation<sup>34)</sup> for sample input of MOSRA-Light

## 7. 入出力ファイル

### 7.1 入出力機番とプリント出力ファイル

MOSRA-Lightの入出力ファイルには、通常の順編成データセットファイル(PSファイル)と階層構造を有するバイナリーファイル(PDSファイル)との2種類がある。これらのファイルはプログラムにより定義された入出力機番により読み書きが行われる。MOSRA-Lightは他コードへの移植が容易なモジュラーコードとしての性格上、ファイルのI/Oに使用する入出力機番を最小限にするとともに、これらをメインプログラムのみで容易に変更できるようにしている。MOSRA-Lightが使用する入出力機番は以下の5つである。

機番	ファイル形式(レコード長)	データの内容
5	テキスト(80)	標準入力
6	テキスト(137)	標準出力(計算ステップのモニター等)
45	バイナリー	PDSファイルの各種メンバー
95	テキスト(80)	スクラッチ(標準入力の複製)
99	テキスト(137)	計算結果

機番5から与えられる標準入力は一旦機番95のスクラッチファイルに複写され、実際の入力データの読み込みはこのスクラッチファイルから行われる。

MOSRA-Lightでは複数のバイナリーファイル(メンバー)を利用するが、これらは全て1つの機番(45)を窓口としてオープンとクローズを繰り返すことによりアクセスされる。詳細については第7.2節以降で述べる。

標準出力(6番)には、入力データのエコーバックの他、使用したサブルーチン名、積算計算時間、PDSファイルのアクセス状況、エラーメッセージ、デバッグ情報などが、時系列的にプリントされる。これらの情報量はオプションの指定(例えばBlock-2-9のIPRN(3))により多段階に制御することができる。6番のファイル内容はいわば計算ステップの監視情報であり、計算が異常終了した場合などにその原因を解明するのに役立つ。

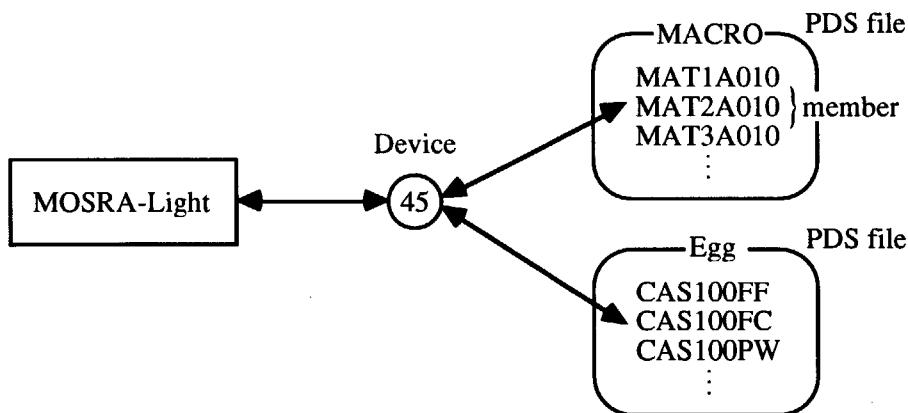
中性子増倍率、中性子束分布、及び出力分布などの主要な計算結果は99番のファイルにプリント出力される。ただし、入力のデフォルト設定では中性子増倍率以外の情報は特に指定がない限り書き出さない。

全ての計算が正常に終了すると、6番及び99番のファイルの最後に以下のような一文がプリントされる。逆にプリントされない場合には、何らかの理由により計算が中断したものと考えて良い(APPENDIX-B.1参照)。

"===== END OF MOSRA CALCULATION ====="

## 7.2 バイナリーファイル(PDS)の構造

MOSRA-Lightが使用するバイナリーデータセットは、下図に示すように一階の階層構造を有しており、1つのディレクトリ内にメンバーと呼ばれる複数のバイナリーファイルが存在する。データの読み込みと書き出しをする際には、その都度1つの機番(45)を介してファイルのオープンとクローズが行われる。



上の図の例では、MACRO及びEggがPDSファイルであり、UNIX上では単なるディレクトリにすぎない。これらの中には8文字の名前を持つ複数のメンバーが存在し、メンバー名がそのデータ内容を表している。例えば、MAT1A010～MAT3A010は物質単位の巨視的断面積セットであり、CAS100FFとCAS100FCは、それぞれCAS1と言う名の計算ケースで得られた中性子束分布と部分中性子流などの情報を収納している。収納されるデータの内容はメンバーのタイプにより異なるが、具体的な内容については次節以降で述べるものとし、ここではメンバーファイルの一般的な構造について説明する。メンバーには次のような特徴がある。

- ・単精度実数型と整数型が混在するデータを収納する。
- ・データの先頭には、常に収納するデータの数(word)が記録されている。

例えば、1つのメンバーに実数型データA(1)～A(100)と整数型データN(1)～N(50)が収納されている場合には、下図のような1次元データの構造を持つことになる。

LENG	A(1)	A(2)	A(3)	...	A(100)	N(1)	N(2)	...	N(50)
------	------	------	------	-----	--------	------	------	-----	-------

この場合、LENG=150であるが、LENGを含めたデータ総長は151(words)である。こうしたバイナリーデータは、例えば次のようなプログラムで読み込むことができる。

```
C-----
C Read data from member TEST0000 in current directory
  DIMENSION A(100),N(50),WORK(200),IWORK(200)
  EQUIVALENCE (WORK(1),IWORK(1))
  OPEN(UNIT=45,FILE='TEST0000',FORM='UNFORMATTED',ACCESS=SEQUENTIAL)
  READ(45) LENG, (WORK(I),I=1,LENG)
  DO 100 I=1,100
    A(I)=WORK(I)
100 CONTINUE
  DO 200 I=1,50
    N(I)=IWORK(I+100)
200 CONTINUE
C-----
```

モジュラーコードであるMOSRA-Lightには基本的な計算機能しか与えられていない。ユーザリティコードであるMore-MOSRAがこれを補う役割を果たすが、ユーザー独自の計算結果の編集を望む場合には、MACROファイルやEggファイルに格納される必要なメンバーの内容を上記のようなプログラムで読み出して利用して頂きたい。

逆にメンバーにデータを書き込む場合には、例えば以下のようなプログラムを使用する。

```
C-----
C Write data to member TEST0000 in current directory
  DIMENSION A(100),N(50),WORK(200),IWORK(200)
  EQUIVALENCE (WORK(1),IWORK(1))
  DO 100 I=1,100
    WORK(I)=A(I)
100 CONTINUE
  DO 200 I=1,50
    IWORK(I+100)=N(I)
200 CONTINUE
  LEMG=150
  OPEN(UNIT=45,FILE='TEST0000',FORM='UNFORMATTED',ACCESS=SEQUENTIAL)
  WRITE(45) LENG, (WORK(I),I=1,LENG)
C-----
```

なお、SRACコードも同様なPDSファイルを採用しており、SRACが出力する巨視的断面積ファイル(MACRO)中のメンバーの内容も上記のようにして読み書きすることができる。

MOSRA-Lightを実行する上では、メンバー内部のデータの詳細やこれらのデータにアクセスするプログラミングに精通している必要は無いが、少なくともどのようなデータがどのメンバーに収納されているかは知っておいて頂きたい。

### 7.3 MACROファイルの内容

MACROファイルはMOSRA-Lightで使用する巨視的断面積セットを収納する。SRACコードにより出力される巨視的断面積ファイル(MACRO)をそのままMOSRA-LightのMACROファイルとして利用することができる。例えば、SRACコードのMACROファイルには通常以下のような8文字の名前のメンバーが含まれる。

メンバー名	内容
CONTe000	エネルギー群構造に関する情報
caseebx0	均質化断面積(P0)
caseebxN	(n,2n)反応巨視的断面積
caseebxZ	遅発中性子データ
case-tag	SRACにおける格子計算のケース名(4文字)
e-tag	エネルギー範囲を表す1つの英字(e=F:高速群、e=A:全群) 通常はe=Aであるが、SRACによる格子計算を高速群のみで行った場合にはe=Fとなる。
b-tag	格子燃焼計算における時間ステップを表す1つの英数字(0,1,2,.....,9,A,B.....Z) 燃焼計算を行わない場合にはb=0となる。
x-tag	格子内の均質化領域を表す1つの英数字(1,2,...,9,A,B...Z) 通常は格子全体を1つの均質化領域に指定するためx=1である。

例えば、第6.2.3節のSRAC計算の入力例では、"HMXN"と言う計算ケースによりMACROファイルには、CONTA000、HMXNA010、HMXNA01N、HMXNA01Zの4つのメンバーが作成され、引き続き同じエネルギー群構造で行われる計算ケース"UO2N"では、UO2NA010、UO2NA01N、UO2NA01Zの3つのメンバーが作成される。

#### 【MACROファイル利用上の留意点】

- 1) SRACの入力オプションによっては、P1断面積や燃焼情報など上記以外のメンバーが含まれることがあるが、MOSRA-Lightには関係しない。MOSRA-Lightの計算に直接必要となるのはP0均質化断面積メンバー(caseebx0)のみである。また、(n,2n)反応断面積メンバー(caseebxN)がMACROファイル中に存在する場合にはこれを使用して(n,2n)反応の断面積補正が行われる(第4.2.9節参照)。同メンバーが存在しない場合には補正は行われない。
- 2) SRACの計算においては、少数群に縮約された巨視的断面積はMACROファイルに出力され、詳細群の断面積はMACROWRKファイルに出力される。MOSRA-LightはMACROWRKファイルのメンバー(第4.2.8節参照)も利用できるが、e-tagが'A'のメンバーに限られる。高速群と熱群が別々に解かれた格子計算の場合には、高速群断面積(e-Tag=

'F')と熱群断面積(e-Tag='T')が別のメンバーに収納される。このような断面積メンバーは MOSRA-Lightで直接利用することはできない。最も簡単な対処法は、詳細群構造と同じ群構造で群縮約を行い、MACROWKファイルと同じ断面積をMACROファイルに出力することである。非均質物質の断面積も均質化領域を物質毎に指定すればMACROファイルに出力することができる。

- 3) MOSRA-Lightに使用する巨視的断面積を標準入力から与える場合(第5.3.6節参照)には、入力された断面積データはユーザーが指定する名前のメンバーとしてMACROファイルに作成される。このようにして作成された断面積メンバーは、別途行うMOSRA-Lightの計算に利用できる他、SRACコードにも利用することができる。ただし、この場合には SRACの計算に必要とされるエネルギー群構造に関するメンバー(CONTe000)は作成されない。このため、ユーザーがこのメンバー自分で作成するか、もしくは同じエネルギー群数のSRAC計算により出力される同メンバーを複写する必要がある。また、(n,2n)反応断面積メンバー(caseebxN)と遅発中性子データに関するメンバー(caseebxZ)もMOSRA-Lightでは作成されない。このため、SRACのCITATIONモジュールでは(n,2n)反応補正がスキップされる他、1点炉動特性パラメータの計算機能は利用できない。
- 4) MOSRA-LightによりMACROファイルに新しい断面積メンバーを作成することはできるが、既存のメンバーの上書きは許していない。必要ならばユーザーがrmコマンドにより不要なメンバーを消去する。

以下にMACROファイルのメンバーに収納される具体的なデータ内容を示す。記号 / / 内に各メンバーが収納するデータ数(word)を表示するが、これにはメンバーの先頭に常時記録されているデータ数自体(第7.2節参照)は含まれていない。

(1) エネルギー群構造に関するメンバー (CONTe000) /2\*(ng+1)/

ng	エネルギー群数
(W(g),g=1,ng)	重み付きレサジー幅(群縮約に使用される漸近スペクトル)
(E(g),g=1,ng+1)	エネルギー群境界(eV)

[Note]

- ・ MOSRA-Lightが参照するのはngのみである。MACROファイル中にこのメンバーが存在する場合には、MOSRA-Lightの入力で与えるエネルギー群数(NG)とngが一致しない場合にはエラーとなる。

(2) 均質化P0断面積メンバー(caseebx0)

$$/10*ng + \sum_{g=1}^{ng} LGV(g)/$$

1 LSS(g) 散乱断面積データの中の自群散乱位置

2 LGT(g) 散乱断面積のデータ長

3  $\Sigma_u(g)$  中性子速度(u)に関する定数

$$\Sigma_u(g) \equiv \sqrt{\frac{1}{E_g(eV)}} \quad (7.3.1)$$

$$u(g) = \sqrt{\frac{2}{m_n}} \frac{1}{\Sigma_u(g)} = \frac{13831.7}{\Sigma_u(g)} \text{ (m/sec)} \quad (7.3.2)$$

4  $\Sigma_f(g)$  核分裂断面積5  $v\Sigma_f(g)$  生成断面積6  $\Sigma_t(g)$  トータル断面積または輸送断面積7  $\chi(g)$  核分裂スペクトル

8 D1(g) 1番目の拡散係数

9 D2(g) 2番目の拡散係数

10  $\Sigma_a(g)$  吸收断面積

以下に長さLGVの散乱断面積データが続く。

1	$\Sigma_{g \rightarrow g1}$	$g1=g-LSS(g)+1$
2	$\Sigma_{g \rightarrow g1+1}$	
3	$\Sigma_{g \rightarrow g1+2}$	
-	-	
-	-	
LSS(g)	$\Sigma_{g \rightarrow g}$	
-	-	
-	-	
LGV(g)	$\Sigma_{g \rightarrow g2}$	$g2=g+LGV(g)-LSS(g)$

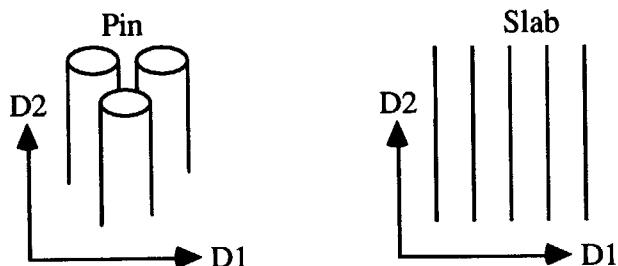
以上のデータが各群毎に、 $g=1 \sim ng$ 回繰り返される。

## [Note]

- $\Sigma_u(g)$ は、第g群の中性子速度(u)を計算するための擬似的な断面積であり、MOSRA-Light

の計算には使用されないが、SRACおよびMore-MOSRAにおいて1点炉動特性パラメータの計算に使用される。

- トータル断面積 $\Sigma_t(g)$ は拡散コードであるMOSRA-Lightには使用されない。
- SRACにより作成される拡散係数としては通常D1を使用するが、格子計算のオプションで方向依存拡散係数<sup>10)</sup>を指定すると、衝突確率計算によりD1とD2に異なる値が与えられる(下図参照)。これらのD1とD2をMOSRA-Lightの計算座標(x/y/z)のどの方向に割り当てるかはMOSRA-Lightの入力で指定する。



[ SRACにより作成される方向依存拡散係数 ]

(3) (n,2n)反応断面積メンバー(caseebxN)

$$/10*ng + \sum_{g=1}^{ng} LGV(g)/$$

このメンバーは(n,2n)反応断面積のマトリックスをP0断面積メンバー(caseebx0)と同じ構造で収納する。但し、散乱断面積データ以外の $\Sigma_u(g) \sim \Sigma_a(g)$ の位置には、便宜的にゼロ値が充てられる。

[Note]

- (n,2n)反応断面積はMOSRA-LightやSRAC-CITATIONにおいて(n,2n)反応を考慮するために除去断面積の補正に使用される(第4.2.9節参照)。メンバーが無い場合には補正は行われない。

(4) 遅発中性子に関するメンバー(caseebxZ)

$$/3*ng*IFM/$$

$$(\Sigma^{BVF}(i,g), i=1, IFM), g=1, ng)$$

遅発中性子割合に関するパラメータ(7.3.3式)

$$(\chi_d(i,g), i=1, IFM), g=1, ng)$$

遅発中性子スペクトル

$$(\Sigma^{BVL}(i,g), i=1, IFM), g=1, ng)$$

遅発中性子先行核の崩壊定数に関するパラメータ(7.3.4式)

## [Note]

- IFMは遅発中性子ファミリーの群数で、通常IFM=6であるが均質化格子内に重水が含まれる場合にはIFM=15である。
- $\Sigma_{i,g}^{BVF}$  と  $\Sigma_{i,g}^{BVL}$  は以下の式で定義される遅発中性子に関する量である。これらはMOSAR-Lightでは使用されないが、More-MOSRA及びSRAC-CITATIONによる1点炉動特性パラメータの計算に使用される。

$$\Sigma_{i,g}^{BVF} = \sum_m \left\{ \beta_m^i v_{d,g,m} \sigma_{f,g,m} N_m \right\} \quad (7.3.3)$$

$$\Sigma_{i,g}^{BVL} = \sum_m \left\{ \beta_m^i v_{d,g,m} \sigma_{f,g,m} N_m / \lambda_m^i \right\} \quad (7.3.4)$$

- $\beta_m^i$  : 核種mのi番目familyの規格化遅発中性子割合 ( $\sum \beta_m^i = 1.0$ )  
 $v_{d,g,m}$  : g群における核種mの核分裂により発生する遅発中性子発生数  
 $\sigma_{f,g,m}$  : g群における核種mの微視的核分裂断面積 (barn)  
 $N_m$  : 核種mの個数密度 ( $n/cm^3$ )  
 $\lambda_m^i$  : 核種mのi番目familyの遅発中性子先行核の崩壊定数 (1/sec)

遅発中性子データをこのような形で整理しているのは、 $\Sigma_{i,g}^{BVF}$  と  $\Sigma_{i,g}^{BVL}$  を通常の断面積と同様にして均質化や群縮約をすることができるためである。More-MOSRAでは、MOSRA-Lightがoutputする中性子束分布( $\phi_{g,0}^k$ )と隨伴中性子束分布( $\phi_{g,0}^{*k}$ )を使用して実効遅発中性子割合( $\beta_{eff}$ )、遅発中性子先行核の実効崩壊定数( $\lambda_{eff,i}$ )、即発中性子寿命( $l$ )及び生成時間( $\Lambda$ )を以下の式で計算する。

$$\beta_{eff} = \sum_i \beta_{eff,i} = \sum_i \left[ \frac{\sum_k \int_{V_k} \sum_g \phi_{g,0}^{*k} \chi_{d,g,i}^k \sum_{g'} \Sigma_{i,g'}^{BVF,k} \phi_{g',0}^k dv}{\sum_k \int_{V_k} \sum_g \phi_{g,0}^{*k} \chi_g^k \sum_{g'} v \Sigma_{f,g}^k \phi_{g',0}^k dv} \right] \quad (7.3.5)$$

$$\frac{1}{\lambda_{eff,i}} = \frac{\sum_k \int_{V_k} \sum_g \phi_{g,0}^{*k} \chi_{d,g,i}^k \sum_{g'} \Sigma_{i,g'}^{BVL,k} \phi_{g',0}^k dv}{\sum_k \int_{V_k} \sum_g \phi_{g,0}^{*k} \chi_{d,g,i}^k \sum_{g'} \Sigma_{i,g'}^{BVF,k} \phi_{g',0}^k dv} \quad (7.3.6)$$

$$l = \frac{\sum_k \int_{V_k} \sum_g \phi_{g,0}^{*k} \frac{1}{u_g^k} \phi_{g,0}^k dv}{\sum_k \int_{V_k} \sum_g \phi_{g,0}^{*k} \frac{\chi_g^k}{k_{eff}} \sum_{g'} v \Sigma_{f,g}^k \phi_{g',0}^k dv} = k_{eff} \Lambda \quad (7.3.7)$$

## (5) 不連続因子メンバー(任意8文字)

/6\*ng/

(DCF(g,1),g=1,ng) x-面の第g群不連続因子

(DCF(g,2),g=1,ng) x+面の第g群不連続因子

(DCF(g,3),g=1,ng) y-面の第g群不連続因子

(DCF(g,4),g=1,ng) y+面の第g群不連続因子

(DCF(g,5),g=1,ng) z-面の第g群不連続因子

(DCF(g,6),g=1,ng) z+面の第g群不連続因子

## [Note]

- 不連続因子メンバーはSRACによっては直接作成されない。ユーザーが格子計算で得られる中性子束分布を編集して上記の形式でMACROファイルのメンバーとして作成する必要がある。MOSRA-Lightでは、不連続因子を標準入力で与えられる他、MACROファイルの不連続因子メンバーを指定して読み込むことができる(第5.3.11節参照)。

## 7.4 Eggファイルの内容

EggファイルにはMOSRA-Lightの主要な計算結果がいくつかのメンバーに分けられて収納される。これらのメンバーは、リストアートファイルや別の計算ケースの初期推定値として利用できる。また、付属のユーティリティコード(More-MOSRA)により、MOSRA-Lightには無い機能の計算をすることができる(第4.2.16節参照)。Eggファイルには以下のメンバーが出力される。ただし、入力データにおいてIOPT(14)に負の値を与える場合には全てのメンバーの書き出しがスキップされる。

メンバー名	内容	IOPT(14)≥0 以外の出力条件
caseGEOM	幾何形状情報、入力オプション	常時
case00FF	中性子束(高次を含む)	IOPT(3)  ≠ 1
case00AF	随伴中性子束(高次を含む)	IOPT(3)  > 0
case00FJ	部分中性子流	IOPT(3)  ≠ 1
case00AJ	随伴計算における部分中性子流	IOPT(3)  > 0
case00PW	出力分布	IOPT(3)  ≠ 1
case00XS	ノード毎の巨視的断面積	IOPT(3)  ≠ 1, IOPT(14)=2/6
case00AJ	ノード毎の不連続因子	IOPT(3)  ≠ 1, IOPT(14)=4/6, IOPT(12)≠0

case-tag : MOSRA-Lightにおける4文字の計算ケース名(第5.3.2節のBlock-2-2参照)

例えば、第6.2.1節のMOSRA-Light計算の入力データ例1では、"TEST"と言う計算ケースによりEggファイルには、TESTGEOM、TEST00FF、TEST00FJ、TEST00PWの4つのメンバーが作成される。

以下にEggファイルのメンバーに収納される具体的なデータ内容を示す。記号/ /内に各メンバーが収納するデータ数(word)を表示するが、これにはメンバーの先頭に常時記録されているデータ数自体(第7.2節参照)は含まれていない。

### (1) 幾何形状情報に関するメンバー (caseGEOM)

/ 44+2\*(NX\*NY\*NZ)+7\*NOD+(NX+NY+NZ) /

1-20	IOPT(1-20)	入力Block-2-8で与える一般制御変数
21	NG	エネルギー群数(Block-2-7)
22	NGT	熱エネルギー群数(Block-2-7)
23	NX	x方向ノード分割数(Block-3-2)
24	NY	y方向ノード分割数(Block-3-2)
25	NZ	z方向ノード分割数(Block-3-2)

NX、NY、NZは実際の分割数であり、前の計算ケース参照やメッシュ

		オーバーレイなどのオプション指示に使用するゼロ値または負値は実際の値に置き換えられる。
26	NMAT	物質数(Block-4-2)
27	NOD	真空ノードまたは対角対称ノードを除くノード数(=中性子束を計算するノード総数=NRED+NREDB+NREDV+NBLK+NBLKB+NBLKV)
28	NRED	外部境界または黒体に接しない赤色ノード(黒体は除く)
29	NREDB	外部境界または黒体に接する赤色ノード
30	NREDV	黒体として扱われる赤色ノード
31	NBLK	外部境界または黒体に接しない黒色ノード(黒体は除く)
32	NBLKB	外部境界または黒体に接する黒色ノード
33	NBLKV	黒体として扱われる黒色ノード NRED～NBLKVは境界分離チェックカードスウェーブ(第3.4.2節参照)のために必要な変数であり計算結果としての本質的な意味は無い。
34	NERGN	Edit-Regionの数(Block-7-2) 前の計算ケースのEdit-Regionを参照するためにNERGN=0を入力した場合にも実際の値(前ケースの指定値)がセットされる。Edit-Regionを使用しないケース(IOPT(6)=0)ではNERGN=0である。
35-41	IBCD(1-7)	境界条件 (Block-3-3)

引き続き以下のデータが収納される。

(IDMAT(i,j,k),i=1,NX),j=1,NY),k=1,NZ)

(i,j,k)の位置にあるノードに割り当てられた物質番号(Block-3-7,3-8)  
内部黒体を含むノードには負の値が、真空ノードなど中性子束が計算されない領域にはゼロ値が与えられる。

(IX(n),n=1,NOD) n番目のノードのx方向位置(1≤IX≤NX)  
1次元のノード番号(n)は、ベクトル計算のためにプログラム内部で付けられるものである(Fig.3.4.2参照)。必要であればMOSRA-Lightのプリント制御入力(IPRN(1))によりプリントすることができる。また、後述のNXYZ(i,j,k)によっても知ることができる。

(IY(n),n=1,NOD) n番目のノードのy方向位置(1≤IY≤NY)

(IZ(n),n=1,NOD) n番目のノードのz方向位置(1≤IZ≤NZ)

(NXYZ(i,j,k),i=1,NX),j=1,NY),k=1,NZ)

(i,j,k)位置のノードに与えられた1次元のノード番号(1≤NXYZ≤NOD)  
必然的にNXYZ(IX(n),IY(n),IZ(n))=nである。

IHLK(n),n=1,NOD n番目ノードの6つの面の回りの状態を表す6桁の整数(3.4.1式参照)

(XXX(i),i=1,NX+1) XXX(1)=0.0cmから始まるx方向ノード分割境界座標(Block-3-4)

(YYY(i),i=1,NY+1) YYY(1)=0.0cmから始まるy方向ノード分割境界座標(Block-3-5)

(ZZZ(i),i=1,NZ+1) ZZZ(1)=0.0cmから始まるz方向ノード分割境界座標(Block-3-6)

(DELX(n),n=1,NOD) n番目ノードのx方向ノード幅(cm)

(DELY(n),n=1,NOD) n番目ノードのy方向ノード幅(cm)

(DELZ(n),n=1,NOD) n番目ノードのz方向ノード幅(cm)

## (2) 中性子束メンバー(case00FF) /2+7\*NOD\*NG/

1 AKEFF 実効中性子増倍率(固有値)  
固定源問題の計算ケースでは1.0が与えられる。

2 FACT 中性子束の規格化因子(第4.2.10節参照)

引き続き以下のデータが収納される。

((FLXA0(n,g),n=1,NOD),g=1,NG)  
n番目ノードの第g群に対するノード平均中性子束( $n/cm^2/s$ )  
FACTにより規格化された値 ( $\Phi_{g,0}^n = FACT * \phi_{g,0}^n$ )

((FLXX1(n,g),n=1,NOD),g=1,NG)  
n番目ノードの第g群に対するx方向1次のモーメント中性子束  
FACTにより規格化された値 ( $FACT * \phi_{g,x1}^n$ )

((FLXY1(n,g),n=1,NOD),g=1,NG)  
n番目ノードの第g群に対するy方向1次のモーメント中性子束  
FACTにより規格化された値 ( $FACT * \phi_{g,y1}^n$ )

((FLXZ1(n,g),n=1,NOD),g=1,NG)  
n番目ノードの第g群に対するz方向1次のモーメント中性子束  
FACTにより規格化された値 ( $FACT * \phi_{g,z1}^n$ )

((FLXX2(n,g),n=1,NOD),g=1,NG)  
n番目ノードの第g群に対するx方向2次のモーメント中性子束  
FACTにより規格化された値 ( $FACT * \phi_{g,x2}^n$ )

((FLXY2(n,g),n=1,NOD),g=1,NG)

n番目ノードの第g群に対するy方向2次のモーメント中性子束  
FACTにより規格化された値 (FACT\* $\phi_{g,y2}^n$ )

((FLXZ2(n,g),n=1,NOD),g=1,NG)

n番目ノードの第g群に対するz方向2次のモーメント中性子束  
FACTにより規格化された値 (FACT\* $\phi_{g,z2}^n$ )

[Note]

- 2次のNEMによる計算の場合にも(4.2.43)式と(4.2.44)式に基づく高次モーメント中性子束が与えられる。これにより、2次のNEMにより得られた収束解を4次のNEMの初期値として利用することができる(第4.2.5節の(2)及び第4.2.6節の(2)参照)。

(3) 随伴中性子束メンバー(case00AF)

/2+7\*NOD\*NG/

随伴問題の場合もメンバーの内容は同じであるが、随伴中性子束はエネルギー群を逆転した形で収納される。これはエネルギー群を反転して解く方が外部反復計算の収束速度が速いためである(第4.2.2節参照)。ただし、入力オプション(IOTP(3)<0)によって、エネルギー群を逆転させないで解く場合には、随伴中性子束は通常の中性子束の場合と同様に収納される。

1 AKEFF 随伴計算における実効中性子増倍率(固有値)

2 FACT =1.0

引き続き以下のデータが収納される。ただしIOTP(3)>0の場合には(g1=NG, g2=1)であり、IOTP(3)<0の場合には(g1=1, g2=NG)とする。

((FLXA0(n,g),n=1,NOD),g=g1,g2)

n番目ノードの第g群に対するノード平均随伴中性子束

((FLXX1(n,g),n=1,NOD),g=g1,g2)

n番目ノードの第g群に対するx方向1次のモーメント随伴中性子束

((FLXY1(n,g),n=1,NOD),g=g1,g2)

n番目ノードの第g群に対するy方向1次のモーメント随伴中性子束

((FLXZ1(n,g),n=1,NOD),g=g1,g2)

n番目ノードの第g群に対するz方向1次のモーメント随伴中性子束

((FLXX2(n,g),n=1,NOD),g=g1,g2)

n番目ノードの第g群に対するx方向2次のモーメント随伴中性子束

((FLXY2(n,g),n=1,NOD),g=g1,g2)

n番目ノードの第g群に対するy方向2次のモーメント随伴中性子束

((FLXZ2(n,g),n=1,NOD),g=g1,g2)

n番目ノードの第g群に対するz方向2次のモーメント随伴中性子束

[Note]

- ・2次のNEMによる計算の場合にも(4.2.43)式と(4.2.44)式に基づく高次モーメント随伴中性子束が与えられる。
- ・エネルギー群を逆転させないオプション(IOP(3)<0)を使用すると、共通の編集用プログラムにより中性子束メンバーと随伴中性子束メンバーのデータを読み込める利点がある。しかし、このオプションはエネルギー群数が多い場合ほど収束速度を遅くさせるため推奨しない。なお、More-MOSRAはIOP(3)の符号に関係なく利用できる。
- ・随伴中性子束の収束解を別の随伴問題の初期値として参照する場合には、IOP(3)の符号を合わせておく必要がある。

(4) 中性子流メンバー(case00FJ)

/12\*NOD\*NG/

((CINXP(n,g),n=1,NOD),g=1,NG)

n番目ノードのx+面に流入する第g群の部分中性子流 ( $J_{g,x+}^{in,n}$ )

((CINYP(n,g),n=1,NOD),g=1,NG)

n番目ノードのy+面に流入する第g群の部分中性子流 ( $J_{g,y+}^{in,n}$ )

((CINZP(n,g),n=1,NOD),g=1,NG)

n番目ノードのz+面に流入する第g群の部分中性子流 ( $J_{g,z+}^{in,n}$ )

((CINXN(n,g),n=1,NOD),g=1,NG)

n番目ノードのx-面に流入する第g群の部分中性子流 ( $J_{g,x-}^{in,n}$ )

((CINYN(n,g),n=1,NOD),g=1,NG)

n番目ノードのy-面に流入する第g群の部分中性子流 ( $J_{g,y-}^{in,n}$ )

((CINZN(n,g),n=1,NOD),g=1,NG)

n番目ノードのz-面に流入する第g群の部分中性子流 ( $J_{g,z-}^{in,n}$ )

((COTXP(n,g),n=1,NOD),g=1,NG)

n番目ノードのx+面から流出する第g群の部分中性子流 ( $J_{g,x+}^{out,n}$ )

((COTYP(n,g),n=1,NOD),g=1,NG)

n番目ノードのy+面から流出する第g群の部分中性子流 ( $J_{g,y+}^{out,n}$ )

((COTZP(n,g),n=1,NOD),g=1,NG)  
 $n$ 番目ノードのz+面から流出する第g群の部分中性子流( $J_{g,z+}^{out,n}$ )

((COTXN(n,g),n=1,NOD),g=1,NG)  
 $n$ 番目ノードのx-面から流出する第g群の部分中性子流( $J_{g,x-}^{out,n}$ )

((COTYN(n,g),n=1,NOD),g=1,NG)  
 $n$ 番目ノードのy-面から流出する第g群の部分中性子流( $J_{g,y-}^{out,n}$ )

((COTZN(n,g),n=1,NOD),g=1,NG)  
 $n$ 番目ノードのz-面から流出する第g群の部分中性子流( $J_{g,z-}^{out,n}$ )

## [Note]

- 部分中性子流は中性子束メンバーのFACTにより規格化された値として出力される。

(5) 随伴中性子流メンバー(case00AJ) /12\*NOD\*NG/

中性子流メンバー(case00FJ)と同じ構造であるが、随伴中性子束メンバーの場合と同様に  
 $IOPT(3)>0$ の場合にはエネルギー群を逆転した形で収納される。

(6) 出力分布メンバー(case00PW) /3+NOD/

- |   |       |                                  |
|---|-------|----------------------------------|
| 1 | POWER | 熱出力(Wt)または中性子束の規格化指定(POWER≤0の場合) |
| 2 | VOLFS | 核分裂領域体積(cm <sup>3</sup> )        |
| 3 | AVPOW | 炉心平均出力密度(全ての核分裂領域の平均値)           |

引き続き以下のデータが収納される。

(POWDT(n),n=1,NOD)  
 ノード毎の出力密度

## [Note]

- POWERには入力指定(Block-2-12)によって与えた値が収納される。
- 出力密度(AVPOW, POWDT)の単位はPOWERの符号によって異なる。例えば、炉心熱出力の入力(POWER)を正の値で与えた場合には、出力密度の単位は[W/cm<sup>3</sup>]で与えられる。POWER=-1(炉心熱出力を核分裂領域体積に規格化)とする場合には、AVPOW=1.0となり出力分布はAVPOWに対する相対値として出力される(第4.2.10節参照)。

- ・随伴問題のみの計算ケースでは本メンバーは出力されない。

(7) 巨視的断面積メンバー(case00XS) /NOD\*NG\*(8\*+NG)/

MACROファイルの巨視的断面積は物質単位に1つのメンバーとして与えられるが、本メンバーはノード毎の断面積に整理して出力される。断面積の値はノード毎に割り当てられた物質に対するMACROファイルのメンバーのものと同じである。

((SIGA(n,g),n=1,NOD),g=1,NG)

n番目ノードに対する第g群の吸収断面積 ( $\Sigma_{a,g}^n$ )

((SIGR(n,g),n=1,NOD),g=1,NG)

n番目ノードに対する第g群の除去断面積 ( $\Sigma_{r,g}^n$ )

((SIGF(n,g),n=1,NOD),g=1,NG)

n番目ノードに対する第g群の核分裂断面積 ( $\Sigma_{f,g}^n$ )

((SIGP(n,g),n=1,NOD),g=1,NG)

n番目ノードに対する第g群の生成断面積 ( $v\Sigma_{f,g}^n$ )

((SIGX(n,g),n=1,NOD),g=1,NG)

n番目ノードに対する第g群の核分裂スペクトル ( $\chi_g^n$ )

((SIGDX(n,g),n=1,NOD),g=1,NG)

n番目ノードに対する第g群のx方向拡散係数 ( $D_{x,g}^n$ )

((SIGDY(n,g),n=1,NOD),g=1,NG)

n番目ノードに対する第g群のy方向拡散係数 ( $D_{y,g}^n$ )

((SIGDZ(n,g),n=1,NOD),g=1,NG)

n番目ノードに対する第g群のz方向拡散係数 ( $D_{z,g}^n$ )

((SIGS(n,g,g'),n=1,NOD),g=1,NG),g'=1,NG)

n番目ノードに対する第g群から第g'群への散乱断面積 ( $\Sigma_{s,g \rightarrow g'}^n$ )

#### [Note]

- ・除去断面積はバッククリングによる疑似吸収と(n,2n)反応の補正が加えられたもの(4.2.118式)である。
- ・随伴問題のみの計算ケースでは本メンバーは出力されない。これは随伴問題では断面積の配置が便宜的に入れ替えられて計算が行われるためである(第4.2.2節参照)。

## (8) 不連続因子メンバー(case00DC)

/6\*NOD\*NG/

((DCXP(n,g),n=1,NOD),g=1,NG)

n番目ノードのx+面に対する第g群の不連続因子

((DCXN(n,g),n=1,NOD),g=1,NG)

n番目ノードのx-面に対する第g群の不連続因子

((DCYP(n,g),n=1,NOD),g=1,NG)

n番目ノードのy+面に対する第g群の不連続因子

((DCYN(n,g),n=1,NOD),g=1,NG)

n番目ノードのy-面に対する第g群の不連続因子

((DCZP(n,g),n=1,NOD),g=1,NG)

n番目ノードのz+面に対する第g群の不連続因子

((DCZN(n,g),n=1,NOD),g=1,NG)

n番目ノードのz-面に対する第g群の不連続因子

Eggファイルのメンバーを出力する(IOPT(14)≥0)と、MOSRA-Lightのメモリー使用量や計算時間が増大する。不要なメンバーの出力は極力控える方が好ましい。

caseGEOM～case00PWのメンバーはデフォルト入力(IOPT(14)=0)によりEggファイルに出力される。プリント出力ファイルを消去してしまった場合やEdit-Regionを変更してプリントする場合に、これらのメンバーを使用して再計算をすることにより、短時間で計算結果を得ることができる。

MACROファイルのメンバーは上書きから保護されているが、Eggファイル内のメンバーは保護されていない。例えば同じケース名を使用するリスタート計算では、メンバーの内容は更新される。リスタート計算でも古いメンバーを敢えて保存しておきたい場合には、計算ケース名を変えて初期値参照コールドスタートのオプション(IOPT(2)=2)を利用する。

## 参考文献

- 1) Fowler T. B., Vondy D. R. and Cunningham G. W. : "Nuclear Reactor Analysis Code ;CITATION", ORNL-TM-2496 (1969).
- 2) Vondy D. R., Fowler T. B. and Cunningham G. W. : "VENTURE: A Code Block for Solving Multigroup Neutronics Problems Applying the Finite-Difference Diffusion-Theory Approximation to Neutron Transport, Version III", ORNL-5062/R1 (1977).
- 3) Delp D. L., Fischer D. L., Harriman J. M. and Stedwell M. J. : "FLARE, A Three-Dimensional Boiling Water Reactor Simulator", GEAP-4598, General Electric Company (1964).
- 4) Gupta N. K. : "Nodal Methods for Three-Dimensional Simulators", Prog. Nucl. Energy, 7, 127 (1981).
- 5) Greenman G., Smith K. S., and Henry A. F. : "Recent Advances in an Analytic Nodal Method for Static and Transient Reactor Analysis", Proc. Computational Methods in Nuclear Engineering, p.3-49, Williamsburg, VA, 23-25 April, ANS (1979).
- 6) Finnenmann B., Bennewitz F. and Wagner M. R. : "Interface Current Techniques for Multidimensional Reactor Calculations", Atomkernenergie, **30**, 123 (1977).
- 7) Smith K. S., Ver Planck D. M. and Edenius M. : "QPANDA: An Advanced Nodal Method for LWR Analyses", Trans. Am. Nucl. Soc., **50**, 532 (1985).
- 8) Lawrence R. D. and Dorning J. J. : "A Nodal Green's Function Method for Multidimensional Neutron Diffusion Calculations", Nucl. Sci. Engr. **76**, 218 (1980).
- 9) Lawrence R. D. : "Progress in Nodal Methods for the Solution of the Neutron Diffusion and Transport Equations", Prog. Nucl. Energy, **17**, 271 (1986).
- 10) 竹田敏一 : "軽水炉における新しい出力分布計算法", 原子力工業, **33**, 58 (1987).
- 11) 竹田敏一, 牛尾直史 : "核計算法の研究成果と炉心設計法の進歩 (II-1 ノード法・CCCP 法)", 日本原子力学会誌, **39** [1], 3 (1997).
- 12) Benoist P. : "Theorie du Coefficient des Diffusion des Neutrons dans un Reseau Comportant des Cavites", CEA-R 2278 (1964).
- 13) Duderstadt J. J. and Hamilton J. L. : "Nuclear Reactor Analysis", John Wiley & Sons, Inc., New York, p.143 (1976). または、成田正邦、藤田文行 共訳 : "原子炉の理論と解析(下)", 現代工学社, 東京, p.142 (1982).
- 14) Noh J. M. and Cho N. Z. : "A New Approach of Analytic Basis Function Expansion to Neutron Diffusion Nodal Calculation", Nucl. Sci. Eng. **116**, 165 (1994).
- 15) Cho N. Z. and Noh J. M. : "Analytic Function Expansion Method for Hexagonal Geometry", Nucl. Sci. Eng. **121**, 245 (1995).
- 16) Smith K.S. : "Assembly Homogenization Techniques for Light Water Reactor Analysis", Prog. Nucl. Energy, **17**, 303 (1986).

- 17) Weinberg A. M. and Wigner F. P. : "The Physical Theory of Neutron Chain Reactor", University of Chicago Press., p.264 (1958).
- 18) 曽野浩樹, 中野佳洋, 山根義宏, 三澤毅, 横尾健司 : "日本原子力研究所・研究炉JRR-3M-の燃料領域照射筒内部の詳細スペクトル計算", JAERI-Research 95-059 (1995).
- 19) Azmy Y. Y. : "Multiprocessing for Neutron Diffusion and Deterministic Transport Methods", Prog. Nucl. Energy, **31**, 317 (1997).
- 20) Kim Y. H. and Cho N. Z. : "Parallel Solution of the Neutron Diffusion Equation with the Domain Decomposition Method on a Transputer Network.", Nucl. Sci. Eng. **114**, 252 (1993).
- 21) Fuentes E. and Turinsky P. J. : "Parallel Implementation of Integral Transport Methods", Nucl. Sci. Eng. **121**, 277 (1995).
- 22) Uematsu M. : "Parallel Processing of Two-Dimensional Sn Transport Calculation", Nucl. Sci. Eng. **125**, 300 (1997).
- 23) Kirk B. L. and Azmy Y. Y. : "A New Communication Scheme for the Neutron Diffusion Nodal Method in a Distributed Computing Environment", Proc. of Topical Meeting on Advances in Reactor Physics, Knoxville, TN April 11-15, **1**, 298 (1994).
- 24) Argonne Code Center : "Benchmark Problem B601 (11. multi-dimensional (x-y-z) LWR model)", ANL-7416, Suppl. 2, Argonne National Laboratory (1977).
- 25) 奥村啓介, 金子邦男, 土橋敬一郎 : "SRAC95;汎用核計算コードシステム", JAERI-Data/Code 96-015 (1996).
- 26) Lawrence R. D. : "Perturbation Theory Within the Framework of a Higher Order Nodal Method", Trans. Am. Nucl. Soc., **46**, 402 (1984).
- 27) Kim T. K. and Kim C. H. : "Solution of Mathematical Adjoint Equation for a Higher Order Nodal Expansion Method", Nucl. Sci. Eng. **123**, 381 (1996).
- 28) 中原康明, 藤村統一郎 : "原子炉核計算コードにおける反復解法の収束加速アルゴリズム", JAERI-M 5590 (1974).
- 29) Wagner M. R. : "GAUGE: A Two-Dimensional Few Group Neutron Diffusion-Depletion Program for A Uniform Triangular Mesh", GA-8307, Gulf General Atomic (1968).
- 30) 藤村統一郎 : 私信
- 31) Koebke K. and Hetzelt L. : "On the Reconstruction of Local Homogeneous Neutron Flux and Current Distributions of Light Water Reactors from Nodal Schemes", Nucl. Sci. Eng. **91**, 123 (1985).
- 32) Rempe K. R, Smith K. S. and Henry A.F. : "SIMULATE-3 Pin Power Reconstruction : Methodology and Benchmarking", Nucl. Sci. Eng. **103**, 334 (1989).
- 33) Groff A. G. : "ORIGEN-2 A Revised and Updated of the Oak Ridge Isotope Generation and Depletion Code", ORNL-5621 (1989).
- 34) Lutz D., Bernnat W., Hesketh K. and Sartori E. : "Physics of Plutonium Recycling", Vol.3 (Benchmark on Void Reactivity Effect in PWRs), OECD/NEA (1995).

## 付録-A MOSRA-Lightコードのインストール

### A.1 会話型インストーラ

MOSRA-Lightはベクトル計算機用に開発したコードであるが、UNIX(LinuxやFreeBSDを含む)をOSとするほとんどの計算機で利用可能である。使用言語は原則としてFORTRAN77であるが、パーソナル計算機においては計算の本質と関係ない部分(日付、時刻、計算時間の取得)においてC言語を使用している。これまでに当研究室で利用可能な以下の計算機で動作確認を行っている。現在のところ並列計算には対応していない。

#### 【大型共用計算機】

FACOM/VPP-500(ベクトル)、FACOM/VPP-300(ベクトル)、FACOM/AP-3000(スカラ)、NEC/SX3(ベクトル)、NEC/SX4(ベクトル)、IBM/SP2(スカラ)、HITACHI/SR2201(スカラ)

#### 【ワークステーション】

SUN/SPARC 20、SUN/UltraSPARC、HP-9000/735、IBM/AIX RS/6000

#### 【パーソナルコンピュータ】

PC/AT (Pentium-II, Linux + f2c+gcc または Linux + g77)

MOSRA-Lightコードのファイル一式は、テープまたはFTPにより提供される(正規に利用手続きをしたユーザーに限る)。テープには、tar形式によるアーカイブファイル(MOSRA-Light.tar)が含まれる。以下のコマンドによりテープから、MOSRA-Light.tarをディスクに書き出す。

```
tar -cvf /dev/rst# MOSRA-Light.tar
```

なお、テープDevice番号(#)は環境によって異なる(システム管理者に確認)。MOSRA-Light.tarをインストール先のディレクトリ(任意)上に置き、次のコマンドにより解凍する。

```
tar -xvf MOSRA-Light.tar
```

これにより、MOSRA-Lightと言う名前のディレクトリが作成され、配下にインストーラ、ソースプログラム、サンプル入力などのファイルが収納される。後述のインストーラを起動する前であれば、トップディレクトリ(MOSRA-Light)の名前を変更することもできる。例えば、

```
mv MOSRA-Light MOSRA-Light98
```

これによりバージョンの異なるMOSRA-Lightを同居させることができるが、特に不都合が

無ければ、ユーザーのホームディレクトリ上にMOSRA-Lightの名前で置くことを勧める(サンプル問題を実行する際、シェルスクリプトの内容を変更する必要が無い)。以下では、トップディレクトリの名前を'MOSRA-Light'のままにしたものとして説明を続ける。

MOSRA-Lightディレクトリ配下のファイル構成の概略を以下に示す。より詳細な情報はインストーラのヘルプ機能で表示可能である。

```
MOSRA-Light : top directory
  -@PunchMe (Installer for MOSRA-Light)
  -bin   : directory for MOSRA-Light load modules
  -obj   : directory for object modules and Makefile for MOSRA-Light
  -smpl  : directory for sample problems
    -shr    : directory for shell-script and input data
    -outp   : directory for output data (text)
    -pds    : directory for input/output data (binary)
  -src   : directory for source programs
    -difxyz : directory for main part of MOSRA-Light
    -pdsacs : directory for PDS access part
    -common  : directory for common part with other codes
    -extnl   : directory for system-dependent part
    -inc     : directory for include files
  -tmp   : directory for temporary files (work directory)
  -tool  : directory for installation tools
    -install : directory for MOSRA-Light installer
    -permit   : directory for tools to permit files to other users
    -else    : directory for other tools (source conversion, etc.)
  -more  : directory for utility programs (More-MOSRA)
    -@MorePunchMe (Installer for More-MOSRA)
```

### 【インストーラ：@PunchMe】

MOSRA-Lightディレクトリの直下に会話型インストーラ(@PunchMe)を用意している。インストーラはC-shellで書かれたコマンドである。これを起動することにより、MOSRA-Lightコードを容易にインストールすることができる。@PunchMeは以下の機能を持つ。

- 1) ヘルプ機能(コード概要、ファイル構成、インストール手引き、入力の手引き(英文)、開発担当者連絡先、バージョン番号・最終修正日)の表示
- 2) プリプロセッサ(計算機種に依存するソースプログラムの生成、ベクトル制御文の書き換え、Makefileの作成)。
- 3) プログラムのインストール(ロードモジュール作成、カスタムインストール：コンパイルオプションの変更、フルソースプログラムの生成等)
- 4) プログラムのアンインストール

@PunchMeを実行すると、以下のメインメニューが現れ上記の機能を選択することができる。

```
+++++++
Wellcome to MOSRA-Light installation conductor. (by K.Okumura)
Usual installation process ;
(1) Read Help ==> Menu No.1
(2) Execute Pre-processor ==> Menu No.2
(3) Install Load Module of MOSRA-Light ==> Menu No.3

++++++< Main Menu >+++++
+
+ 1 : Help & version No. +
+ 2 : Menu for pre-processor +
+ 3 : Menu for installation of MOSRA-Light (after pre-processor) +
+ 4 : Menu for uninstallation of MOSRA-Light +
+ q : bye-bye +
+
+++++
Set number ==>
```

インストールの一般的な手順は、メインメニューにおいて先ずヘルプ情報を読み(1を選択)、プリプロセッサを起動させて機種依存性への対処を行い(2を選択)、ロードモジュールを作成する(3を選択)。これらの機能により新規に作成されたロードモジュール等のファイルは、アンインストーラにより削除することができる。インストールに失敗して作業をやり直す際や、ファイルを初期化して別の機種にインストールする際にアンインストーラを使用する(4を選択)。以下に1~4の選択により現れるサブメニューを示す。サブメニューから出る(qを選択)とメインメニューに戻る。以下にメインメニューから入ることができるサブメニューを示す。

#### 【サブメニュー：ヘルプ】

```
=====< Help Menu >=====
*
* 1 : about the MOSRA-Light code *
* 2 : about file structure *
* 3 : about installation *
* 4 : about contact to consultants *
* 5 : output all contents of help in file (help.txt) *
*
* 6 : output users input manual (man.txt) *
* 7 : show current version number *
* q : bye-bye *
=====
Set number ==>
```

ヘルプメニューの1番~4番は、それぞれの項目についての情報を画面に表示する。5番はそれらの内容をMOSRA-Lightディレクトリ配下にファイル(help.txt)として出力する。6番は入力マニュアル(英文)をファイル(man.txt)に出力する。7番はMOSRA-Lightコードのバージョン番号と最終更新日を表示する。

## 【サブメニュー：プリプロセッサ】

```
=====
Pre-processor will generate system-dependent source programs
and Makefile for load module of MOSRA-Light.
===== < Pre-processor Menu > =====
*
* q : bye-bye
* h : Help
*
* Do you know who I am ?
*
* 1 : You don't know who I am, but you know I have a
*      FORTRAN compiler.
* 2 : I am Sun Sparc or it's relations.
* 3 : I am HP-9000 or it's relations.
* 4 : I am FACOM-VPP(Vector) or it's relations.
* 5 : I am NEC-SX3(Vector) or it's relations.
* 6 : I am IBM AIX RISC SYSTEM/6000 or it's relations.
* 7 : I am Personal Computer with Linux system.
*
=====
Set number ==>
```

プリプロセッサでは、ユーザーが使用している計算機システムを選択する。ここで選択により、システムに依存したプログラム(ファイルの初期化、日時、計算時間の取得など)に関するビルトインルーチンの呼び出し)がMOSRA-Light/src/extnl配下に展開される。また、ベクトル計算機の場合には、ベクトル制御文が変換される(第A.2節参照)。更に、MOSRA-Light/obj配下に、機種に依存したコンパイルドライバ(例えばf77)と推奨するコンパイルオプションを使用したMakefile(ロードモジュール作成・更新のためのスクリプト)が作成される。

メニューに無い計算機の場合には、とりあえず類似の計算機を選択するか、もしくは1番を選択しておく。ただし、1番を選択した場合には、日時や計算時間の取得ルーチンはダミールーチン(固定値を出力)で置き換えられる。日時の取得は単なる計算実行日時を表示させるためのもので計算の本質には関係ない(固定値は筆者の誕生日)。計算時間については、CPU時間の制限による計算の中止機能が無効になる。共有計算機などでシステムが管理する計算時間の制限により計算が中断されると、リスタート計算用のファイルが出力されないなどの弊害を生ずる。CPU時間を計測する適当な代替関数がFORTRANに用意されている場合には、プリプロセッサを起動した後で、MOSRA-Light/src/extnl配下に作成されるサブルーチン(uclckm.f)を書き換えると良い(方法はサブルーチン内部に記述)。ただし、アンインストールを行うと元のダミールーチンに戻される。適当なFORTRANの代替関数が無い計算機の場合でCコンパイラがある場合には、PC用に用意しているC言語のルーチン(MOSRA-Light/src/extnl/p c-unix/\*.c)を利用することもできる。この場合には、ユーザーがMakefile(MOSRA-Light/obj/Makefile)を適切に書き換えてからロードモジュールを作成する必要がある。

何らかの原因により、プリプロセッサによりMakefileが作成されない場合には、MOSRA-Light/tmp/mkmk.outlstの内容を見て原因を調べる。

## 【サブメニュー：インストール】

```
=====
Load modules of MOSRA-Light code can be installed here.
If installation failed, go to the custom-installation menu.
```

```
=====< Installation Menu >=====
*
* 1 : Installation of MOSRA-Light load module
* 2 : Menu for custom-installation of MOSRA-Light
* h : Help
* q : bye-bye
*
```

```
=====
Set number ==>
```

インストールメニューの1番は、プリプロセッサにより作成されたMOSRA-Light/obj/Makefileを使用してロードモジュールを作成(make)する。makeはバックグラウンドジョブで実行され、MOSRA-Light/bin配下にロードモジュール(MOSRA-L.lm)が作成される。ほとんどの場合、以上でインストール作業は終了する。Makefileやシェルコマンドに馴染みがないユーザーでも、インストーラの指示に従えばプログラムをインストールできるはずである。

何らかの原因でMOSRA-Light/bin配下にロードモジュールができない場合には、makeの結果(MOSRA-Light/tmp/ProgInst.outlist)をcatコマンド等(viはラインが長すぎて利用できないことがある)で確認する。

2番を選択すると、Makefileのコンパイルドライバやコンパイルオプションを変更することができるが、Makefileに精通しているユーザーは、直接MOSRA-Light/obj/Makefileを変更してmakeしても良い。2番やMakefileの修正は、原則としてロードモジュールがうまく作成できない場合や実行エラーが生じてプログラムを修正する必要がある場合に行う。プログラムの修正は以下の手順で行う。

- 1) プリプロセッサを起動してMOSRA-Light/obj/Makefileを作成する。
- 2) MOSRA-Light/src配下にあるソースプログラムを修正する。特にシステムに依存するソースに問題がある場合にはsrc/extnl配下のプログラムを変更する。
- 3) 必要があれば、Makefileを修正する。
- 4) 以下のコマンドによりロードモジュールを作成する。

```
cd MOSRA-Light/obj <-
make <-
```

ロードモジュールが作成できたら、動作確認用のサンプル問題(MOSRA-Light/smpl/shr/Test.sh)を実行する。この計算はごく短時間(数秒～数十秒)で終了するはずである。これに対する計算結果の出力例はMOSRA-Light/smpl/outp配下に収納されている。diffコマンドなどにより結果を比較し、動作確認を行う。

## 【サブメニュー：アンインストール】

```
===== < Uninstallation Menu > =====
*
*   1 : Uninstallation of pre-processed data
*   2 : Uninstallation of MOSRA-Light load module and object *
*   3 : Uninstallation for all (initialization)
*   q : bye-bye
*
=====
Set number ==>
```

アンインストールメニューは、プリプロセッサメニュー やインストールメニューで作成されたファイルを削除するためのものである。メニューの1番は、プリプロセッサにより作成された、システム依存のソースプログラム、Makefile、その他の一時作業ファイルを削除する。ユーザーが追加・修正したソースやMakefileも削除される可能性があるため、ユーザーが作成した重要なファイルは予めMOSRA-Lightディレクトリの外へ退避させておく。2番は、インストールメニューで作成されたオブジェクトモジュール、ロードモジュール等を削除するが、プリプロセッサで作成されたファイルは残される。3番は、MOSRA-Lightファイルを初期化(インストール前の状態に戻す)するためのもので、1番と2番の機能に加え、MOSRA-Light/tmp、MOSRA-Light/outp、MOSRA-Light/smpl/pds配下にユーザーが作成した計算結果なども削除される。この場合にも、ユーザーが作成し保存しておく必要があるファイルはMOSRA-Lightディレクトリの外へ移動させることを勧める。なお、ヘルプメニューで作成されるhelp.txtとman.txtはアンインストーラでは削除されない。また、MOSRA-Light/smpl/shr配下にユーザーが作成したシェルスクリプトも保護される。

## A.2 ベクトル制御指示行の変更

一般にベクトル制御指示行(第3.2.1節の(5)参照)はシステムまたは計算機メーカに固有の文法で、ソースプログラム中にコメント行として書き込まれる。MOSRA-Lightのソースプログラムでは、FACOMが定める文法に基づき記述しており、他メーカのベクトル計算機に対しては単なるコメントにすぎず何の効力も持たない。MOSRA-Lightが使用しているベクトル制御指示行は以下の2種類である。

## 1) \*VOCL LOOP, NOVREC

本ベクトル制御指示行の下に位置するDOループは回帰演算を含まない(ベクトル化可能である)ことをコンパイラに指示する。

## 【FACOM例】

```
*VOCL LOOP, NOVREC
DO 100 I=1,1000
.
100 CONTINUE
```

## 2) \*VOCL STMT, IF(%) (ただし#はIF文の真率(%)を表す1から99までの整数値)

本ベクトル制御指示行は、DOループの中においてIF分岐を行う文の前に配置し、そのIF文が真である確率(%)を指定する。これに基づきIF分岐を含むDOループの適切なベクトル処理をコンパイラに指示する。一般的に言えばIF分岐を含むDOループはベクトル化の対象とならないが、コンパイラによってはマスク付き演算などの技法によりベクトル処理するものもある。ただし、マスク処理のための前準備などが内部的に行われるため、IF分岐を含まないDOループに比べ効率的では無い。また、IF文の真率が実状と異なる場合にはかえって処理を遅くすることもある。

FACOM以外のベクトル計算機でMOSRA-Lightを高速に実行するためには、上記のベクトル制御指示行を、使用するベクトル計算機の文法に基づき書き換える必要がある。ベクトル制御指示行はgrepコマンドなどにより検索することができるが、これを用いて計算機タイプ毎に検索とプログラムの書き換えを手作業で行うのは効率的では無い。

そもそも、MOSRA-Lightのソースプログラムはベクトル制御指示行に多くは頼らないプログラミングとしている。すなわち、ほとんどの部分はIF文を含まないベクトル長の長い単純なDO文で構成され、ベクトル制御文を書き換えなくても、自動ベクトルコンパイルの機能だけである程度の高速化は実現できる。また、上記の2)の指示行は高速化の本質的な部分には使用していないため無視しても構わない。しかし、チェックカードスウェープをベクトル処理させるには、1)の制御指示行はぜひ書き換えて使用したいところである。1)に示したベクトル制御指示行はベクトル計算機には最も基本的な機能であり、表現こそ異なるものの全てのベクトル計算機が同じ機能を持っている。例えば、NECのベクトル計算機(SXシリーズ)の場合には、1)に対応する指示行は次のように表現される。

## 【NEC例】

```
*vdir nodep
  DO 100 I=1,1000
  :
  100 CONTINUE
```

すなわち、MOSRA-LightのFACOM以外のベクトル計算機への移植は極めて単純であり、

## \*VOCL LOOP, NOVREC

で始まる行を、使用するベクトル計算機の同じ機能『回帰演算を含まないDO文であることの指示』を表すベクトル制御指示行で置き換えてやれば良い。

この指示行を含むサブルーチンは、MOSRA-Light/src/difxyz配下にしか含まれていない。インストーラでは、以下のような処理によりベクトル制御指示行を自動的に書き換えている。

- 1) オリジナルのソースプログラムをdifxyz.sxに複写する。
- 2) catコマンドによりdifxyz.sx配下のサブルーチンを1つのファイルにかためる。  
例：cat \*.f > tmp.sx

- 3) tmp.sxに含まれるベクトル制御指示行を目的の計算機の制御指示行で書き換える。  
インストーラではawk言語(MOSRA-Light/tool/else/VPconv/VPPtoSX/vc.awk参照)を使用しているが、sedやFORTRANプログラムを使用しても良い。
- 4) fsplitコマンドによりtmp.sxをサブルーチン単位に分解する。  
例：**fsplit tmp.sx**
- 5) Makefileを使用してロードモジュールを作成する。

実際にはMOSRA-Lightのベクトル制御指示行は僅かに20行未満であり、手作業による修正でも十分対処可能である。しかし、このような作業を汎用的に自動化しておくと後にソースプログラムに追加や修正が加えられた場合に便利である。プリプロセッサのメニューに現れるベクトル計算機に対しては、このようなベクトル制御指示行の書き換えをインストーラが自動的に行うが、メニューに現れないベクトル計算機の場合にはユーザーがこれを行う必要がある。

### A.3 可変長配列サイズの変更

プログラム実行時に可変長配列のメモリーサイズ(インクルードファイルで設定)に不足が生じると、エラーメッセージに現在の設定値と必要なメモリーサイズが表示される。なお、必要サイズは使用するサブルーチン単位でチェックを行っているため、何度も修正を要求されるかもしれない。このような場合には、外部反復計算回数の上限を1回に設定して計算を一旦終了させると良い。メモリーサイズの修正が要求される場合には、以下の手続きによりロードモジュールを再作成する。

- 1) MOSRA-Light/obj配下のインクルードファイル(ZMEMORYまたはZMAXMT)に記述してあるパラメータ値(配列の上限)を修正する。
- 2) ロードモジュールを別の名前で作成する場合には、Makefileの中の作成するロードモジュール名(MOSRA-L.lm)を変更する。
- 3) MOSRA-Light/obj配下でmakeする(make ↵)。

プリプロセッサを起動した後では、MOSRA-Light/obj配下にインクルードファイル(ZMEMORY、ZMAXMT)が作成される。ZMEMORYは1次元配列のサイズをパラメータ文で定義しており、以下のような内容である。

```
PARAMETER ( IMEMOR = 4000000)
```

インストール時の設定ではメモリサイズを400万ワードに設定しており、これを越える計算の場合にはパラメータ文の値を変更する。また、物質数の入力(第5.3.4節のNMAT)が100を

越える場合には、インクルードファイルZMAXMTも変更する必要がある。これは、以下のようなパラメータ文により物質数の上限を定義している。

**PARAMETER ( MAXMT=100 )**

MOSRA-Lightは物質数に関しても可変長配列の設計としているが、断面積のメンバーネーム(8文字)を収納するための文字型配列だけは移植上のトラブルを避けるため可変配列に組み込んでいない。

なお、プリプロセッサを再起動したりアンインストールをすると、変更したインクルードファイルの設定は初期の値に戻される。恒久的に変更する場合には、インクルードファイルのオリジナルであるMOSRA-Light/src/inc配下の同名ファイルの内容を変更する。

#### A.4 原研計算機での利用

原研の計算機上には、現在のところVPP-500とAP-3000の以下のディレクトリ上にMOSRA-Lightを置いてある。ただし、今後の原研の組織変更や管理上の都合によりディレクトリを変更することがある。ファイルが見つからない場合やトラブルの際には遠慮無くご連絡して頂きたい(okumura@mike.tokai.jaeri.go.jp)。

**VPP-500 : /dg02/ufs02/j9347/MOSRA-Light  
AP-3000 : /dg02/g0435/j9347/MOSRA-Light**

なお、メッシュ数の多い大規模な計算を行う際には、ベクトル計算機であるVPP-500の使用を推奨する。

## 付録-B トラブル対策

### B.1 エラーによる計算の中止

プログラムが計算を正常に終了しない場合には、以下に述べる(1)～(4)の原因が考えられる。それぞれの場合についてユーザーの対処の仕方を述べる。

#### (1) 自由形式入力データの不整合

自由形式(第5.1節参照)による入力データが読み込まれる際、そのデータ形式がプログラムが要求する形式(文字型、整数型、実数型)と異なっていたり、1つの入力ブロックのデータ数に過不足が生じる場合にエラーメッセージを出力して計算は中断される。例えば以下ののようなメッセージが出力される。

#### 【例1】入力データ形式の不整合

```
<<< ERROR STOP (REAM0 ) >>>
*** INVALID DATA AS INTEGER : <19.0>
LINE: 9 9 19.0           / NX, NY, NZ
```

Note：実数型が要求される場合に整数型で入力しても許容されるが( $10=>10.0$ )、その逆は許されない。

#### 【例2】入力データ数の不足

```
<<< ERROR STOP (REAM0 ) >>>
MORE DATA REQUIRED (REQUIRED      10 INPUT      9 )
LINE( 0 0 0 1 0    1 0 0 0    / GENERAL CONTROL      )
```

例2は、本来10個の入力が要求される入力ブロックにおいて9個の入力しか与えなかつた場合のエラーメッセージである。行をまたがって10個の入力をしているつもりでも、プロックの終了を指示するコメント挿入指示記号' / 'を途中に入れている場合には同じエラーとなる(正しくは' & 'を使用する)。

同じデータ形式の入力ブロックが続く場合には、データ数に過不足があつても入力ブロック単位ではプログラムがデータの誤りを検知できない場合がある。このような場合、エラーメッセージは実際に誤った入力があるラインよりも後ろのラインでデータの不整合を指摘することがある(多くの場合がそうである)。したがって、ユーザーはエラーが指摘されたラインより遡ってデータをチェックする必要がある。チェックする範囲が拡大しないようにプログラムは%から始まる入力セクション名を確認している。

**【例3】入力セクション名の不整合**

```
<<< ERROR STOP (INPT04) >>>
INPUT SECTION NAME SHOULD BE (%MATNAME). WRONG NAME (1 1 1 1 ) WAS
DETECTED.
```

この例では、プログラムは入力セクション名が%MATNAMEになっていないことを指摘しているが、これはむしろ前の入力セクションにデータが正しく与えられていないために生じたエラーと考えて良い。

ブロック単位でプログラムが入力ミスを検知し易くするために、各入力ブロックの最後に記号'/'を挿入することを推奨する。

**(2) 入力データの理論的不整合**

例えば、例4は幾何形状が非対称であるにも関わらず、対角対称境界条件の使用を指示した場合のエラーメッセージである。理論的な入力データの不整合はできる限り検知してメッセージを出力するようしているが、必ずしも全てを網羅しているわけではないため、ユーザーも入念な入力データのチェックを心がけて頂きたい。

**【例4】幾何形状入力の理論的不整合**

```
<<< ERROR STOP (INPTA3) >>>
INVALID INPUT FOR NODE SIZE
NODE SIZE FOR X, Y DIRECTION ARE NOT SYMMETRIC
```

**(3) 計算の続行ができなくなる場合**

計算の途中経過によっては、計算を続行するのに必要なデータが得られなかった、続行できない、あるいは続行しても意味が無いなどの理由により計算が中断することがある。例えば、計算に必要な可変長配列のメモリーが途中で不足した場合や、中性子増倍率の値がユーザーが入力で与える有効範囲を超えた(発散した)場合などである。

**【例5】可変長配列サイズの不足**

```
<<< ERROR STOP (ADDRES) >>>
LACK OF MEMORY SIZE, CHANGE MEMORY SIZE IN MAIN ROUTINE
CURRENT MEMORY SIZE = 2000
REQUIRED MEMORY SIZE = 3119
```

計算の続行が可能でも、プログラムがユーザーに注意を促す場合には警告メッセージが

出力される。例えば、計算時間などの制限により収束解が得られなかつた場合や、負の中性子束が得られた場合等である。これらの場合、プログラムの実行は続行され、リストファイルなどは作成される。

#### 【例6】未収束に対する警告

```
<<< CAUTION      (ITERAT)  >>>
OUTER ITERATION STOPPED, BECAUSE OF SHORT CPU TIME
YOU CAN RESTART FOR THIS CASE WITH EGG FILE
```

#### 【例7】負またはゼロ値の中性子束に対する警告

```
<<< CAUTION      (CHKFLX)  >>>
THERE ARE INVALID FLUXES (= < 0) EXCEPT IN BLACKNESS NODES
NUMBER OF INVALID FLUXES ARE :    10
YOU CAN PRINT ALL OF THEM BY IPRN(2)=2
```

エラーメッセージと警告メッセージは、プリント量が膨大な場合でも容易に検索できるように同じフォーマットで出力される。メッセージは99番と6番のファイルに出力されるが6番にしか出力されないものもある。プログラムが最後まで行われた場合には、両ファイルに以下のような計算の終了メッセージが現れる。これが現れない場合には、6番のファイルに書き出されたエラーメッセージの内容を見て対処すると良い。

#### 【プログラムの終了メッセージ】

```
USED      MAIN MEMORY (WORDS) =     229240
CURRENT  MAIN MEMORY (WORDS) =   4000000
```

```
===== END OF MOSRA CALCULATION =====
```

#### (4) プログラムが予期しない中断

MOSRA-Lightのプログラムが計算を意図的に中断する場合には、これまでの例に示したようなエラーメッセージが必ずプリント出力される。所定のフォーマットのメッセージが現れないで計算が中断している場合には、プログラムが予期しない原因によるものである。例えば、システムが設定する計算時間の制限により中断する場合、ディスク容量がフルになってファイルを書き込めない場合、プログラムのバグにより不正なメモリーアクセスが生じた場合などである。このような場合には通常システムに固有のエラーメッセージが標準エラー出力ファイルに書き出される。エラーメッセージの内容が理解できない場合には、先ずシステム管理者にお問い合わせ願いたい。その結果、プログラムの修正が必要と判断される場合には、計算機環境とエラーの状況を筆者(okumura@mike.tokai.jaeri.go.jp)までご連絡頂ければ幸いである。

## B.2 反復計算が収束しない場合の対処

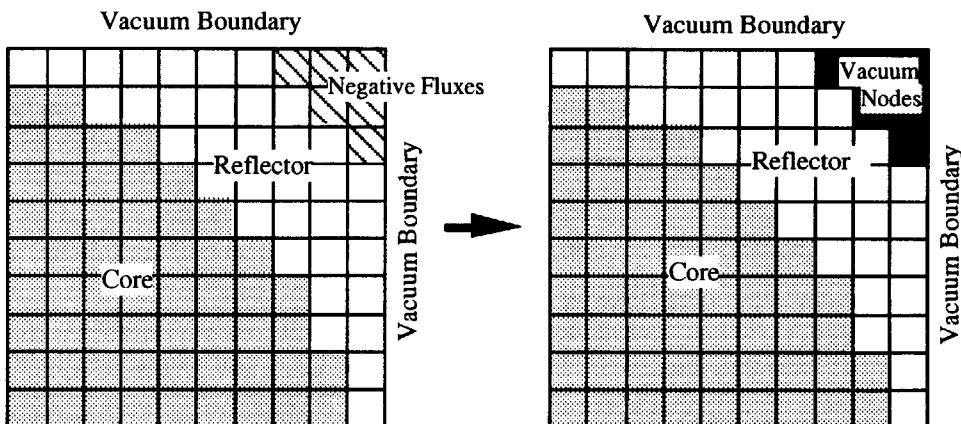
問題によっては、反復計算が収束しなかったり負の中性子束が得られることがある。原因はいろいろ考えられるが、一般に巨視的断面積の値とノードサイズによって決まる拡散方程式の差分化行列式の性質により収束性は左右される。反復計算の収束性が悪くなる問題は以下のような場合である。

- 1) 各方向のノード幅が極端に不揃いな場合
- 2) 中性子束の分布が急峻である場合
- 3) 炉心から遠く離れた反射体など、中性子源が無く真空境界近傍で中性子束レベルが大きく低下しているノードが存在する場合
- 4) 詳細メッシュを使用している場合で内部反復計算の回数が不十分な場合。

ノードの各方向の幅をあまり不揃いにすると、応答行列を作成する場合の逆行列の計算精度が落ち、これが収束性を悪化させる1つの原因となる。1次元計算や2次元計算を行う場合にも、両側に反射境界を使用する方向のノード幅は任意ではあるが、他の方向のノード幅になるべく揃えておく方が好ましい。

急峻な中性子束分布が予想される場合には、ノード幅をあまり大きく設定しない方が好ましい。これは、ノード内の中性子束分布が多項式の表現能力の範囲を越え、結果として不適切な部分中性子流や負の中性子束を発生させるためである。

同様な理由により厚い反射体の真空境界近傍で負の中性子束が発生することがある(左図参照)。



このような場合、負の中性子束が現れる反射体ノードは炉心の特性にほとんど影響を与えない。簡単な対処の仕方は、収束判定をポイント中性子束ではなくポイント中性子源(デフォルト設定)で行い負の中性子束を許容して収束させることである。あるいは右図のように炉心の特性に影響しない範囲で問題となる反射体ノード(全てである必要は無い)を真空ノードに置き換えて計算から除外しても良い。しかし、特に反射体部分の中性子束分布を目的と

して計算を行う場合には、やはり問題となる部分のノード幅を小さくして対処する必要がある。もともと粗メッシュノード法は反射体領域の精度を軽視した手法と言えるため、そうした目的にはいずれにせよ詳細なメッシュを使用する方が好ましい。なお、負の中性子束は最終的にプログラムによりゼロ値に置き換えられて出力される。

粗いノード幅を使用する場合には、内部反復の回数は比較的少なくても良いが、詳細な幅を使用するほど多くの内部反復計算が必要になる。これを怠ると収束しなくなることがある。このため、MOSRA-Lightは収束性が悪いとプログラムが判断した場合には、内部反復計算の収束判定因子と反復回数の上限値を動的に厳しくするようにしている(第4.2.7節参照)。収束制御パラメータにコード設定値を使用しない場合には、問題に応じて適切な設定をする必要がある(厳しくしすぎると計算時間は遅くなる)。

また、筆者の経験では、より複雑な計算を行う4次のNEMよりは、2次のNEMの方が収束性は安定している。

### B.3 計算コストの節約

#### (1) メモリーの節約

コードモジュールが必要とするメモリーサイズは、使用する計算機やコンパイラ及びコンパイルオプションに強く依存する。使用する計算機の総メモリー容量が不十分な場合には、これによりMOSRA-Lightが使用できる可変長配列のメモリーサイズが制限され、大規模な計算はできなくなる。メモリーの増設が可能な場合には、これにより対処するのが望ましいが、不可能な場合には、使用する計算機、コンパイルラー、コンパイルオプションなどを変えてみるのも1つの手である。

MOSRA-Lightは、ベクトル計算機や並列計算機のような同時処理が行われる計算機での高速化を意図して、メモリーを贅沢に使用するプログラミングとなっている。最たる例は、メモリーアクセスの競合を避けるため、同じ断面積が与えられるノードに対しても別個に断面積データを持たせている点である(第3.2.1節の(6)参照)。これは、単一プロセッサのスカラーレ計算機には何の御利益も無い。それほど大きなメモリー容量を持っていない計算機を利用するユーザーには、以下の点に配慮してメモリーの節約をすると良い。

- 1) 大雑把に言って、可変長配列の使用メモリーはノード総数と群数の積に比例して増加する。粗メッシュ、対角対称計算、真空ノードなどの利用によりノード総数を減らすことが効果的である。
- 2) ノード総数が同じ場合には、2次のNEMに比べて4次のNEMの方がより多くのメモリーを消費する。これは、高次のパラメータを収納するためであり約3~4割増となる。精度的に許容できるメッシュサイズであれば、無理に4次のNEMを使用せず2次のNEMを使用する(計算時間も速くなる)。

3) オプション機能(不連続因子、ファイル出力など)を多く使用するほどメモリーは消費される。特に必要としないオプションの乱用は控える。

メモリーは年々安価となってきており、原研の計算機(AP-3000)も今や1プロセッサあたりの搭載メモリーは2GB(単精度で約5億ワード相当)である。逆にこれほど巨大なメモリーを利用できるようになると、むしろ大量データの演算の高速化の方が優先度が高くなってくる。このような状況を考えると、スクラッチファイルへの一時データ書き出しなどによるメモリーの節約は得策ではなく、ただ時が解決してくれるのを待つ方が賢明に思われる。

## (2) 計算時間の節約

MOSRA-Lightの収束制御パラメータのデフォルト設定はオプションにより選択可能であるが、高速演算よりは反復計算の収束性を重視したものといえる。また、その設定自体も内部反復計算にそれほど計算時間を必要としないベクトル計算機用の設定である。経験により最適な収束パラメータの設定が予測できる場合やスカラー計算機を使用する場合には、ユーザー入力オプションにより収束制御パラメータを適切に調整すると計算時間を節約することができる。また、ベクトル計算機が利用できるユーザーは、ぜひこれをを利用して頂きたい。大規模な問題であるほど、スカラー計算機との速度差は格段に大きくなるためである。MOSRA-Lightコードの設計思想は「小規模な問題はそれなりに速く、大規模な問題をより高速に」なのである。

以下にMOSRA-Lightに付属のサンプル問題(内容については第6.2.4節参照)を実行するのに必要な可変長配列のサイズと4種類の計算機による計算時間の一例を示しておく。

Table B.3.1 Required computation cost for execution of sample problems on several computers.

Sample Problems	Memory Size (words)	Vector Computers		Scalar Computers	
		VPP-500	SX-3	SUN	PC/AT
Test.sh	229240	1.0s.	1.9s.	10s.	15s.
IAEA3D1.sh	412420	1.4s.	2.9s.	18s.	27s.
IAEA3D2.sh	2881200	7.9s.	18.4s.	2m.45s.	3m.51s.
IAEA3D4.sh	23048860	1m.39s.	4m.04s.	1h.10m.30s.	1h.13m.32s.
IAEA3D4R.sh	24807196	----	----	----	----
IAEA2D.sh	22588	2.0s.	2.0s.	6s.	10s.
FixSou.sh	415042	0.8s.	1.1s.	5s.	8s.
Black.sh	21760	0.3s.	0.8s.	1s.	1s.
MultG.sh	1551770	4.6s.	10.6s.	48s.	67s.

VPP-500 : FACOM Vector Computer , SX-3 : NEC Vector Computer ,

SUN : Sun Engineering Work Station with UltraSPARC & Solaris 2.5.1

PC/AT : PC/AT with Pentium-II(266MHz) + Linux(f2c+gcc)

This is a blank page.

# 国際単位系(SI)と換算表

表1 SI基本単位および補助単位

量	名称	記号
長さ	メートル	m
質量	キログラム	kg
時間	秒	s
電流	アンペア	A
熱力学温度	ケルビン	K
物質量	モル	mol
光强度	カンデラ	cd
平面角	ラジアン	rad
立体角	ステラジアン	sr

表3 固有の名称をもつSI組立単位

量	名称	記号	他のSI単位による表現
周波数	ヘルツ	Hz	s <sup>-1</sup>
力	ニュートン	N	m·kg/s <sup>2</sup>
圧力、応力	パスカル	Pa	N/m <sup>2</sup>
エネルギー、仕事、熱量	ジュール	J	N·m
功率、放射束	ワット	W	J/s
電気量、電荷	クーロン	C	A·s
電位、電圧、起電力	ボルト	V	W/A
静電容量	ファラード	F	C/V
電気抵抗	オーム	Ω	V/A
コンダクタンス	ジーメンス	S	A/V
磁束密度	ウェーバ	Wb	V·s
磁束密度	テスラ	T	Wb/m <sup>2</sup>
インダクタンス	ヘンリー	H	Wb/A
セルシウス温度	セルシウス度	C	
光束度	ルーメン	lm	cd·sr
照度	ルクス	lx	lm/m <sup>2</sup>
放射能	ベクレル	Bq	s <sup>-1</sup>
吸収線量	グレイ	Gy	J/kg
線量等量	シーベルト	Sv	J/kg

表2 SIと併用される単位

名 称	記 号
分、時、日	min, h, d
度、分、秒	°, ' , "
リットル	L
トン	t
電子ボルト	eV
原子質量単位	u

$$1 \text{ eV} = 1.60218 \times 10^{-19} \text{ J}$$

$$1 \text{ u} = 1.66054 \times 10^{-27} \text{ kg}$$

表5 SI接頭語

倍数	接頭語	記号
10 <sup>18</sup>	エクサ	E
10 <sup>15</sup>	ペタ	P
10 <sup>12</sup>	テラ	T
10 <sup>9</sup>	ギガ	G
10 <sup>6</sup>	メガ	M
10 <sup>3</sup>	キロ	k
10 <sup>2</sup>	ヘクト	h
10 <sup>1</sup>	デカ	da

(注)

- 表1～5は「国際単位系」第5版、国際度量衡局1985年刊行による。ただし、1eVおよび1uの値はCODATAの1986年推奨値によった。
- 表4には海里、ノット、アール、ヘクタールも含まれているが日常の単位なのでここでは省略した。
- barは、JISでは流体の圧力を表わす場合に限り表2のカテゴリーに分類されている。
- EC閣僚理事会指令ではbar、barnおよび「血圧の単位」mmHgを表2のカテゴリーに入れている

## 換 算 表

力	N( $-10^5$ dyn)	kgf	lbf
1	0.101972	0.224809	
9.80665	1	2.20462	
4.44822	0.453592	1	

粘度  $1 \text{ Pa}\cdot\text{s}(\text{N}\cdot\text{s}/\text{m}^2) = 10 \text{ P}$  (ポアズ)(g/(cm·s))

動粘度  $1 \text{ m}^2/\text{s} = 10^4 \text{ St}$  (ストークス)(cm<sup>2</sup>/s)

圧力	MPa( $=10$ bar)	kgf/cm <sup>2</sup>	atm	mmHg(Torr)	lbf/in <sup>2</sup> (psi)
1	10.1972	9.86923	7.50062 $\times 10^3$	145.038	
力	0.0980665	1	0.967841	735.559	14.2233
	0.101325	1.03323	1	760	14.6959
	1.33322 $\times 10^{-1}$	1.35951 $\times 10^{-3}$	1.31579 $\times 10^{-3}$	1	1.93368 $\times 10^{-3}$
	6.89476 $\times 10^{-2}$	7.03070 $\times 10^{-2}$	6.80460 $\times 10^{-2}$	51.7149	1

エネルギー・仕事・熱量	J( $=10^7$ erg)	kgf·m	kW·h	cal(計量法)	Btu	ft·lbf	eV	1 cal = 4.18605J (計量法)	
								= 4.184J (熱化学)	= 4.1855J (15°C)
	1	0.101972	2.77778 $\times 10^{-7}$	0.238889	9.47813 $\times 10^{-4}$	0.737562	6.24150 $\times 10^{18}$		= 4.1868J (国際蒸気表)
	9.80665	1	2.72407 $\times 10^{-6}$	2.34270	9.29487 $\times 10^{-3}$	7.23301	6.12082 $\times 10^{19}$		
	3.6 $\times 10^6$	3.67098 $\times 10^5$	1	8.59999 $\times 10^5$	3412.13	2.65522 $\times 10^6$	2.24694 $\times 10^{25}$		
	4.18605	0.426858	1.16279 $\times 10^{-6}$	1	3.96759 $\times 10^{-3}$	3.08747	2.61272 $\times 10^{19}$		
	1055.06	107.586	2.93072 $\times 10^{-1}$	252.042	1	778.172	6.58515 $\times 10^{21}$		
	1.35582	0.138255	3.76616 $\times 10^{-7}$	0.323890	1.28506 $\times 10^{-3}$	1	8.46233 $\times 10^{18}$		
	1.60218 $\times 10^{11}$	1.63377 $\times 10^{20}$	4.45050 $\times 10^{-20}$	3.82743 $\times 10^{20}$	1.51857 $\times 10^{-22}$	1.18171 $\times 10^{-39}$	1		

放射能	Bq	Ci	吸収線量	Gy	rad	照射線量	
						C/kg	R
	1	2.70270 $\times 10^{-11}$		100	1	1	3876
	3.7 $\times 10^{10}$	1		0.01	1	2.58 $\times 10^{-11}$	1

線量当量	Sv	rem	線量当量	
			1	100
			0.01	1

(86年12月26日現在)

MOSRA-Light;ベクトル計算機のための高速3次元中性子拡散ノード法コード