

J A E R I - M
87-022

モンテカルロ・コード M C N P のベクトル化

1987年2月

栗田 豊*・菅沼 正之**・樋口 健二・浅井 清

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

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

JAERI-M reports are issued irregularly.

Inquiries about availability of the reports should be addressed to Information Division Department of Technical Information, Japan Atomic Energy Research Institute, Tokaimura, Naka-gun, Ibaraki-ken 319-11, Japan.

©Japan Atomic Energy Research Institute, 1987

編集兼発行 日本原子力研究所
印 刷 嶽高野高速印刷

モンテカルロ・コード M C N P のベクトル化

日本原子力研究所東海研究所計算センター
栗田 豊^{*}・菅沼正之^{**}・樋口健二・浅井 清

(1987年1月29日受理)

中性子、光子、中性子・光子の結合系の三次元輸送計算用の汎用連続エネルギー・モンテカルロ・コード M C N P のベクトル化を VP-100 ベクトル計算機を使用して行った。ベクトル化は粒子の独立的挙動を利用し、イベント・バンク方式と呼ばれる手法を使って行った。

M C N P コードのベクトル化には二つの問題がある。ひとつは改造によって持ち込まれるオーバーヘッドが大きいこと、もうひとつは粒子の死によって並列度が低下するためにベクトル処理効率が低下することである。ゆえに本作業では例外処理の孤立化、粒子の逐次サンプリングによってオーバーヘッドの低減及びベクトル処理効率の改善をはかった。

ベクトル版コードのオリジナル版コードに対する実効性能は約 1.3 倍であった。本報告書では、ベクトル化に際して使用した入力データの概要、ベクトル化の方法及びベクトル化の結果とその評価について述べる。

東海研究所：〒319-11 茨城県那珂郡東海村白方字白根 2-4

* (財)原子力データセンター

** 外来研究員（富士通株式会社）

Vectorization of Monte Carlo Code MCNP

Yutaka KURITA,^{*} Masayuki SUGANUMA,^{**}

Kenji HIGUCHI and Kiyoshi ASAI

Computing Center, Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received January 29, 1987)

Continuous energy Monte Carlo code MCNP, which is used to analyze the three dimensional transport for neutron and/or photon, has been vectorized on the FACOM VP-100 vector processor. Vectorization has been done by using independent behavior of particles, and the event bank method has been applied to control multiple particles' random walks.

In vectorization of MCNP code, there are two problems. One is a large overhead introduced by the program modifications for vectorization. Another is a lowering of vector processing efficiency, since parallelism decreases with time due to the death of the particles. In order to reduce the overhead introduced by the program modifications, processing parts for exceptional events have been isolated from the DO loops. A successive sampling method for source particles has been applied to improve the efficiency of vector processing.

The performance ratio of the vectorized version in the vector mode on the FACOM VP-100 to the original one in the scalar mode is 1.3. In this report, we describe sample input data, summary of MCNP code, vectorization technique and performance evaluation of the vectorized MCNP code.

Keywords: MCNP, Monte Carlo method, Vectorization, VP-100, Supercomputer, Continuous Energy, Nuclear Codes, Vector

* NUCLEAR ENERGY DATA CENTER

** On leave from FUJITSU LTD.

目 次

1.はじめに	1
2.入力データの概要	1
3.MCNPコードの概要	4
3.1 MCNPコードの構造	4
3.2 断面積データと核反応	5
3.3 幾何形状記述法	6
3.4 評価法と分散低減法	7
3.5 モンテカルロ計算の制御構造	7
4.ベクトル化の方法	10
4.1 事象処理用粒子バンクの導入	10
4.2 新しい粒子の逐次サンプリング	12
4.3 粒子数が多い事象の優先処理	13
4.4 例外処理部分の孤立化	13
4.5 事象処理部分のベクトル化	15
4.6 幾何形状分類を伴う処理のベクトル化	15
4.7 ベクトル変数とスカラ変数間のインターフェース整合	17
5.ベクトル化の結果と評価	19
5.1 ベクトル化の結果	19
5.2 結果の評価	21
6.おわりに	24
謝辞	25
参考文献	25
付録A 入力データ	26
付録B 粒子の軌跡	28
付録C コードの木構造	31

CONTENTS

1.	Introduction	1
2.	Sample input data	1
3.	Summary of MCNP code	4
3.1	Program structure of MCNP code	4
3.2	Cross section data and types of nuclear reaction	5
3.3	Geometrical systems of MCNP code	6
3.4	Estimator systems of MCNP code	7
3.5	Control structure of Monte Carlo calculation in MCNP code	7
4.	Techniques for vectorization	10
4.1	Introduction of particle bank for event processing	10
4.2	Successive sampling of new source particles	12
4.3	Effective selection method of even bank	13
4.4	Isolation of processing parts of exceptional events in DO loops	13
4.5	Vectorization of event branch	15
4.6	Vectorization of geometric branch	15
4.7	Interface matching between vectorized variable and scalar variable	17
5.	Performance evaluation of vectorized MCNP code and discussions	19
5.1	Performance evaluation of vectorized MCNP code	19
5.2	Discussions	21
6.	Concluding remarks	24
	Acknowledgements	25
	References	25
APPENDIX A.	Sample input data	26
APPENDIX B.	Trajectories of particles in a geometrical structure	28
APPENDIX C.	Tree structure of MCNP code	31

FIGURES

- Fig. 1 A geometrical scheme of sample input data.
- Fig. 2 The sub system of MCNP code.
- Fig. 3 The tree structure of MCRUN routine.
- Fig. 4 Nuclear reaction types treated in MCNP code.
- Fig. 5 The control structure of random walk control routine HSTORY.
- Fig. 6 An example of vectorization for particle transport processing.
- Fig. 7 The control structure of successive sampling for new source particles.
- Fig. 8 The control structure of effective selection method for event bank.
- Fig. 9 An example of exceptional events isolation from DO loops.
- Fig. 10 An example of vectorization for event branch.
- Fig. 11 The control structure in TALLY, TALLYD for the original version in (a) and vectorized version in (b).
- Fig. 12 An example of interface routine in order to match vectorized subroutine and original one.
- Fig. 13 Transition of active particle number during the random walk for batch source particle sampling method in (a) and successive source particle sampling method in (b).
- Fig. A-1 A sample input data (GOZILLA).
- Fig. B-1 Trajectories of particles in the geometrical structure of top view in (a) and side view in (b).
- Fig. B-2 Trajectories of particles which causes fission reaction in the geometric structure of top view in (a) and side view in (b).
- Fig. C-1 Vectorized routines (underlined) in the tree structure at the top level of MCNP code.
- Fig. C-2 Vectorized routines (underlined) in the tree structure at MCRUN routine level.
- Fig. C-3 Vectorized routines (underlined) in the tree structure at HSTORY routine level.
- Fig. C-4 Vectorized routines (underlined) in the tree structure at STARTP routine level.
- Fig. C-5 Vectorized routines (underlined) in the tree structure at TRACK routine level.
- Fig. C-6 Vectorized routines (underlined) in the tree structure at ACETOT routine level.

- Fig. C-7 Vectorized routines (underlined) in the tree structure at TALLY routine level.
- Fig. C-8 Vectorized routines (underlined) in the tree structure at SURFAC routine level.
- Fig. C-9 Vectorized routines (underlined) in the tree structure at DXTRAN routine level.
- Fig. C-10 Vectorized routines (underlined) in the tree structure at PHOTC routine level.
- Fig. C-11 Vectorized routines (underlined) in the tree structure at COLIDN routine level.
- Fig. C-12 Vectorized routines (underlined) in the tree structure at COLIDP routine level.
- Fig. C-13 Vectorized routines (underlined) in the tree structure at TALLYD routine level.

TABLES

- Table 1 Density of nucldie and the dimension of geometrical structure for sample input data GOZILLA.
- Table 2 Names, functions and FORTUNE cost of vectorized routines.
- Table 3 Names, and functions of interface subroutines for common data area in order to match the vectorized variables and the scalar variables.
- Table 4 Resources required for executing the vectorized and original version of MCNP code.
- Table 5 Summary of FORTUNE cost ratios, vectorization ratios and vector lengths of the vectorized routines.
- Table 6 Vector lengths of typical vectorized routines for algorithm 1 and algorithm 2.
- Table 7 CPU time classified by the calculation type of DO loop.

1. はじめに

中性子、光子の輸送問題の数値解法は、決定論的手法と非決定論的手法に大別される。決定論的手法の代表的な数値解法として Sn 法、直接積分法、有限要素法などがある。他方、非決定論的手法として、モンテカルロ法が挙げられる。

モンテカルロ法は、決定論的手法に比べて、複雑な幾何形状の体系を精度よく表現でき、体系の幾何形状の次元、取り扱うエネルギーの群数に計算の量がほとんど依存しないなどの利点がある。

日本原子力研究所（以下、原研と省略する）においても、高速臨界集合体における実験の解析、J R R - 3 (Japan Research Reactor) の改造における炉心全体の計算、核融合炉工学系の実験の解析、臨界安全解析、そして各種コードの評価などにモンテカルロ法によるコードが使用されて¹⁾いる。

モンテカルロ法とは母集団から代表粒子をサンプリングし、粒子の振る舞いを模擬することによって物理量の計算を行う手法である。すなわち、遮蔽および臨界計算においては粒子が媒質を構成する核種との反応によって吸収されたり、あるいは体系外へ漏れたりするまでの粒子の振る舞いを模擬することによって粒子束、増倍率などの計算を行う。

計算精度を向上させるために、サンプリングする粒子の数を増やすと計算時間がかかるのがモンテカルロ法の短所で、計算時間の短縮はモンテカルロ・コードに残されている大きな課題の一つである。この課題には、二つの側面からのアプローチが行なわれている。一つは、分散低減法の改良を行って計算時間の短縮をはかるアプローチである。もう一つはコードをより高速化の可能性を持つアーキテクチャーのマシン向けに最適化することである。更に進んだアプローチとしてモンテカルロ・コード専用のコンピュータ・アーキテクチャーの提案²⁾³⁾開発も試みられている。

今回、我々はベクトル計算機 FACOM VP-100 向けに汎用の連続エネルギー・モンテカルロ・コード MCNP⁴⁾ (a general Monte Carlo Code for Neutron and Photon transport) を最適化した。本報告書では使用したベクトル化の手法とその結果に重点を置いて述べる。第 2 章では、今回ベクトル化作業のサンプルとして用いた入力データの説明を行い、第 3 章には MCNP コードの概要をまとめた。ベクトル化で用いた手法を第 4 章に、ベクトル化の結果とその評価を第 5 章に、第 6 章に全体のまとめを示す。

2. 入力データの概要

ベクトル化のために使用した入力データは中性子の輸送問題に対するものである。問題の体系は核分裂性物質、反射体及びディテクタの三つの媒質から構成されている。核分裂性物質は U²³⁵, Pu²³⁹ 及び U²³⁸ の混合物である。反射体は C¹² で、ディテクタは He³ で構成されている (He³ 比例計数管)。

問題の幾何形状を Fig. 1 に、その構成要素を Table 1 に示す。半径が 40 cm、高さが 300 cm の円筒形をした核分裂性物質が中央にあり、そのまわりを厚さが 25 cm の四角形の管状の反射体がとり囲んでいる。ディテクタは反射体の外側に配置されている。この入力データでは粒子のランダム・ウォークを追跡する範囲を一辺が 300 cm の立方体の内側に限定している。

1. はじめに

中性子、光子の輸送問題の数値解法は、決定論的手法と非決定論的手法に大別される。決定論的手法の代表的な数値解法として Sn 法、直接積分法、有限要素法などがある。他方、非決定論的手法として、モンテカルロ法が挙げられる。

モンテカルロ法は、決定論的手法に比べて、複雑な幾何形状の体系を精度よく表現でき、体系の幾何形状の次元、取り扱うエネルギーの群数に計算の量がほとんど依存しないなどの利点がある。

日本原子力研究所（以下、原研と省略する）においても、高速臨界集合体における実験の解析、J R R - 3 (Japan Research Reactor) の改造における炉心全体の計算、核融合炉工学系の実験の解析、臨界安全解析、そして各種コードの評価などにモンテカルロ法によるコードが使用されて¹⁾いる。

モンテカルロ法とは母集団から代表粒子をサンプリングし、粒子の振る舞いを模擬することによって物理量の計算を行う手法である。すなわち、遮蔽および臨界計算においては粒子が媒質を構成する核種との反応によって吸収されたり、あるいは体系外へ漏れたりするまでの粒子の振る舞いを模擬することによって粒子束、増倍率などの計算を行う。

計算精度を向上させるために、サンプリングする粒子の数を増やすと計算時間がかかるのがモンテカルロ法の短所で、計算時間の短縮はモンテカルロ・コードに残されている大きな課題の一つである。この課題には、二つの側面からのアプローチが行なわれている。一つは、分散低減法の改良を行って計算時間の短縮をはかるアプローチである。もう一つはコードをより高速化の可能性を持つアーキテクチャーのマシン向けに最適化することである。更に進んだアプローチとしてモンテカルロ・コード専用のコンピュータ・アーキテクチャーの提案²⁾³⁾開発も試みられている。

今回、我々はベクトル計算機 FACOM VP-100 向けに汎用の連続エネルギー・モンテカルロ・コード MCNP⁴⁾ (a general Monte Carlo Code for Neutron and Photon transport) を最適化した。本報告書では使用したベクトル化の手法とその結果に重点を置いて述べる。第 2 章では、今回ベクトル化作業のサンプルとして用いた入力データの説明を行い、第 3 章には MCNP コードの概要をまとめた。ベクトル化で用いた手法を第 4 章に、ベクトル化の結果とその評価を第 5 章に、第 6 章に全体のまとめを示す。

2. 入力データの概要

ベクトル化のために使用した入力データは中性子の輸送問題に対するものである。問題の体系は核分裂性物質、反射体及びディテクタの三つの媒質から構成されている。核分裂性物質は U²³⁵, Pu²³⁹ 及び U²³⁸ の混合物である。反射体は C¹² で、ディテクタは He³ で構成されている (He³ 比例計数管)。

問題の幾何形状を Fig. 1 に、その構成要素を Table 1 に示す。半径が 40 cm、高さが 300 cm の円筒形をした核分裂性物質が中央にあり、そのまわりを厚さが 25 cm の四角形の管状の反射体がとり囲んでいる。ディテクタは反射体の外側に配置されている。この入力データでは粒子のランダム・ウォークを追跡する範囲を一辺が 300 cm の立方体の内側に限定している。

M C N P で使用可能なディテクタでの評価法は二種類あり、ひとつはanalog point detector で、もうひとつはtrack length detector である。これらの二つの評価法を用いて粒子束（中性子束）の計算を行っている。

中性子源は、反射体と核分裂物質の間に位置した点状中性子源とする。全部で 12,000 ヒストリの粒子を母集団からサンプリングし、ランダム・ウォークの追跡を行う。平均の中性子束とその分散の中間結果は 4000 ヒストリ間隔でエネルギー別、セル別、時間別、ディテクタ別に出力される。また、分散の収束の度合を見るために 1000 ヒストリ間隔で、あるセルでの全中性子束と分散を出力し、利用者が指定した中性子評価法及びその使用法が適切であるかどうかの判断材料として提供している。

Table 1 Density of nuclide and the dimension of geometrical structure for sample input data GOZILLA.

媒質番号	媒 質 名	核 種	原子密度	形 状
101	核分裂性物質	$U^{238} : Pu^{239} : U^{235} : O^{16} = 100 : 100 : 1 : 2000$	1.0E22	半径 40 cm, 高さ 300 cm の円柱
103	反射体	C^{12}	1.0E22	一辺が 150 cm, 高さ 300 cm, 厚さ 25 cm の四角形の管状
1~12	ディテクタ	He^3	1.0E22	半径 10 cm, 高さ 300 cm の円柱
102	ボイド			
104				
105	系の境界			一辺が 300 cm の立方体

・原子密度の単位は number of atom/cm³, 1.0E22 は 1.0×10^{22} を意味する。

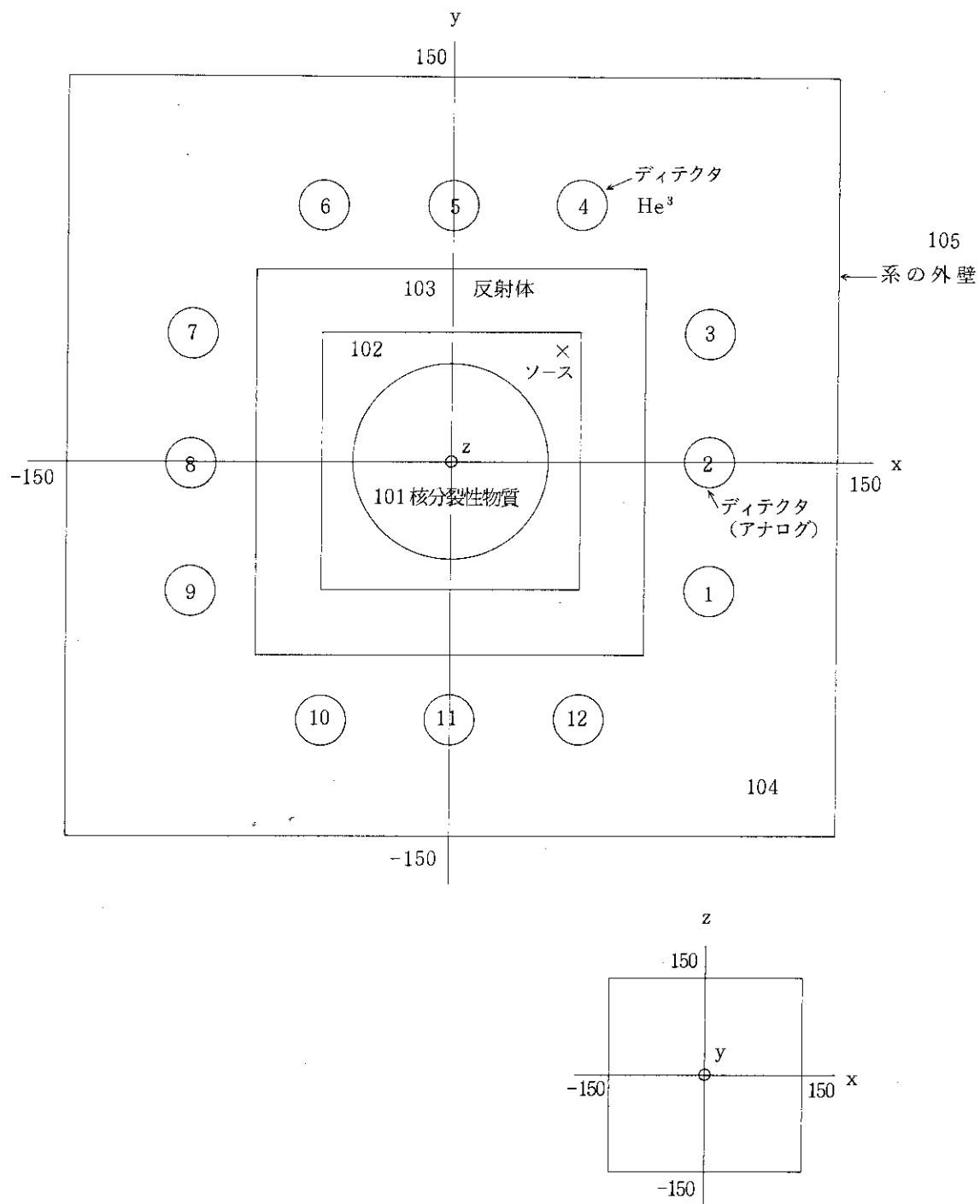


Fig. 1 A geometrical scheme of sample input data.

3. MCNP コードの構造

MCNPコードは、ロスアラモス国立研究所で1970年代の始めに開発されたコードMCNに、光子の輸送も扱えるような改良を加えることによって作成された遮蔽計算、臨界計算ともに行なえる汎用の連続エネルギー・モンテカルロ・コードである。今回のベクトル化の対象になったMCNPコードは第3版で、約40,000ステートメントの規模のコードである。取り扱うことのできるソース分布（エネルギー分布、空間分布）の種類が豊富で、種々の分散低減法が用意されており、問題の体系の記述に対して柔軟性の高い幾何形状記述方式を採用しているのが特徴である。

3.1 MCNP コードの構造

MCNPコードはFig.2に示すように4つの部分から構成されている。IMCNルーチン群は入力データを読み込み、セルの体積計算など粒子の輸送計算の前準備を行う。PLOTルーチン群は体系の幾何形状を会話形式で処理するためのルーチン群である。XACTルーチン群は、断面積ライブリから入力データで指定されたエネルギー範囲の断面積データを読み込むルーチン群であり、MCRUNルーチン群は粒子の輸送計算の実行、中間結果の編集出力などを行なうためのルーチン群である。

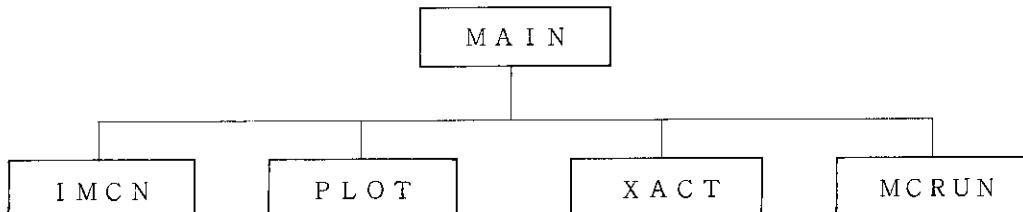


Fig. 2 The sub system of MCNP code.

MCRUNルーチン群の構成をFig.3に示す。OUTPUTルーチン群は各種の統計量を集計し出力する。TRNSPTルーチンは、ヒストリの数、計算時間、ランダムウォークの追跡の途中で幾何学的な矛盾を生じた迷子粒子の数、中間結果の出力間隔、リスタート用データの出力間隔などを制御する。また、HISTORYルーチン群は粒子の輸送計算を行うルーチン群でソース粒子のランダム・ウォークの追跡を行う。なお、HISTORYルーチン群については3.5節で解説する。

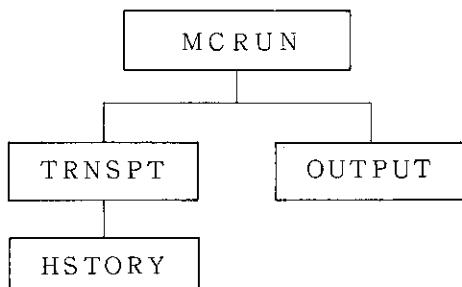


Fig. 3 The tree structure of MCRUN routine.

3.2 断面積データと核反応

モンテカルロ・コードは、連続エネルギー方式と多群エネルギー方式に分類される。MCNPコードは連続エネルギー方式の系列に属する。多群コードの断面積ライブラリである群定数は、その作成過程でENDF/B等の核データ・ライブラリのデータに平均操作を加えて作成する。これに対し、MCNPが使用するACEフォーマットの断面積ライブラリはENDF/B等の核データ・ライブラリから作成する過程での操作が少ないため、高い精度の断面積データを使用することができる。このため多群理論のコードよりも粒子のランダム・ウォークを精密に追跡することができる。ACEフォーマットの断面積ライブラリはNJOYと呼ばれるデータ形式変換コードを使ってENDF/B核データ・ライブラリから作成される。取り扱うことのできるエネルギー範囲は0.00001eV～20.0MeVである。断面積ライブラリはENDF/B核データ・ライブラリ以外からも作成することができ、次のようなものを対象にすることができる。

- ENDF (Evaluated Nuclear Data Library)核データ・ライブラリ
- AWRE (Atomic Weapons Research Establishment) 核データ・ライブラリ

MCNPが扱う中性子の反応の種類をFig.4に示す。中性子の反応では最初に反応のtarget核種が選択される。中性子の入射エネルギーがanalog captureの閾値よりも大きいか否かでanalog captureか、ウエイトの低減によって捕獲を行うimplicit captureかを決定する。捕獲されずに生き残った粒子は散乱の解析が行われ、散乱後の状態が決定されるが、次の条件が満足される場合には熱的取り扱いをそれ以外では非熱的取り扱いが行われる。

- 入力データで熱的取り扱いを指定した
- 衝突核が、水素、重水素、あるいは利用者指定の核である
- 中性子の入射エネルギーが熱的取り扱いの閾値以下である

熱的取り扱いでは、入射エネルギーによって二つの解析モデルを使い分ける。粒子のエネルギーが4～10eVの時には、单原子ガスと中性子との衝突モデルすなわち自由ガス・モデルを、また粒子のエネルギーがそれ以下のときは中性子が分子・結晶格子によって散乱されるS(α, β)モデルは中性子のエネルギーが2 eV以下の時に重要である。

非熱的取り扱いでは入射エネルギーに対する弾性散乱断面積および、非弾性散乱断面積から作られる移行確率テーブルと一様乱数値の比較から弾性散乱、非弾性散乱の決定を行う。散乱後の粒子の飛行方向は角度分布テーブルがある場合はテーブルから、ない場合は一様分布を仮定してサンプリングされる。また、散乱後の粒子のエネルギーは、弾性散乱及び離散的なエネルギー・レベル領域の非弾性の場合には二体問題のkinematicsによって決定し、その他の場合には、6つの散乱モデルごとに別々の方法を使って解析される。

光子の反応については次のように分類されている。

- coherent scattering
- incoherent scattering
- fluorescent emission following the photoelectric absorption
- absorption in pair production

また、光子についての断面積ライブラリはENDF/B核データ・ライブラリから作成し、取り扱うことのできるエネルギー範囲は0.001eV～100.0MeVである。



Fig. 4 Nuclear reaction types treated in MCNP code.

3.3 幾何形状記述法

MCNPの幾何形状記述法は、解析的に表現される面によって形成される三次元空間上の領域間での集合演算（和、積、補集合）をとることによってセルと呼ばれる領域を定義し、複数個のセルの組合せで任意の三次元幾何形状を記述する方式である。基本となる解析的に表現できる面は次のようなものがある。

① 平面

法線が x 軸, y 軸, z 軸, 及び任意方向である平面 (4 types)

② 球面

中心が原点, x 軸上, y 軸上, z 軸上あるいは任意の位置にある (5 types)

③ 円柱面

対象軸が x 軸, y 軸, z 軸に並行あるいは x 軸, y 軸, z 軸の上にある (6 types)

④ 円錐面

対象軸が x 軸, y 軸, z 軸に並行あるいは x 軸, y 軸, z 軸の上にある (6 types)

⑤ 楕円面, 双曲線面, 放物面

軸が座標軸に並行あるいは任意の方向 (2 types)

⑥ 楕円又は円形トーラス

軸が x 軸, y 軸, z 軸に並行である (3 types)

以上 6 種 26 type

3.4 評価法と分散低減法

M C N P コードでは中性子に対して 6 種類の、光子については 4 種類の粒子束、粒子流評価法を用意しており主なものとして次のようなものがある。

- ・指定した表面を通る粒子流の評価
- ・指定した表面を通る粒子束の評価
- ・ディテクタでの粒子束、粒子流の評価
- ・セル内の平均粒子束の評価
- ・その他

ディテクタでの評価法は、ソース粒子あるいは散乱された粒子の存在する位置からディテクタまでの間に一度も散乱されずに到達する確率を求め粒子束を評価する方法の *analog estimator* (あるいは *last flight estimator*, *next event estimator*) と *one more collided flux estimator* が使用されている。

また、セル内の平均粒子束の評価では *track length estimator* が使用されている。この評価法は、セル内での粒子のトラック長の総和をセルの体積で割ることによって平均粒子束を計算する方法である。

M C N P コードでは、分散ばかりでなく同時に計算時間も減少させるために次のような分散低減法を用意している。

- ・Importance sampling
- ・Weight cutoff
- ・Time and energy cutoff
- ・Implicit or analog capture
- ・Exponential transformation
- ・Forced collision
- ・Energy splitting and Russian roulette
- ・Correlated sampling
- ・Source biassing
- ・Weight window
- ・その他

3.5 モンテカルロ計算の制御構造

粒子の輸送計算を行うMCRUNルーチン群において、一つのソース粒子のランダム・ウォークを追跡するルーチンHISTORYの基本的制御構造をFig. 5に示す。M C N P コードでは、まずSTARTPルーチン群でユーザが入力データで指定したソースの空間分布、エネルギー分布から、新しい粒子をサンプリングする。次に、TALLYDルーチン群でソース粒子のディテクタへの寄与を *analog estimator* 法などで計算する。続いて、TRACKルーチン群で次に通過するセル表面までの距離を計算し、ACETOTルーチン群で粒子が存在するセルの構成核種の粒子のエネルギーに対応した微視的断面積を計算すると同時にセル内での核種の原子密度を使用して巨視的断面積を計算する。さらに、

計算された巨視的断面積を使用して粒子の自由行程を計算する。粒子のセル内でのトラック長を決定するためにセル表面までの距離と自由行程を比較し、短い方を粒子のセル内でのトラック長とする。セル内でのトラック長が決定すると、粒子の新しい位置の計算を行い、さらにMCNPコードの track length estimator である TALLY ルーチン群で平均粒子束の計算を行う。

粒子の新しい位置が同じセル内にあるときは核種との反応とみなし、COLIDN ルーチン群で反応解析を行う。核種に吸収されたり分散低減法によって粒子のランダム・ウォークの追跡が中止された場合、粒子は『死』粒子となり、それ以外の場合粒子は生き残る。生き残った粒子に対しては TALLYD ルーチン群で反応の位置からディテクタまでの粒子の到達確率から粒子束を計算し、粒子の追跡を続ける。

粒子の新しい位置が同じセル内にあるときは核種との反応とみなし、COLIDN ルーチン群で反応解析を行う。核種に吸収されたり分散低減法によって粒子のランダム・ウォークの追跡が中止された場合、粒子は『死』粒子となり、それ以外の場合粒子は生き残る。生き残った粒子に対しては TALLYD ルーチン群で反応の位置からディテクタまでの粒子の到達確率から粒子束を計算し、粒子の追跡を続ける。

粒子の新しい位置がセルの表面上にあるときには、SURFAC ルーチン群で次のような処理を行う。まず、表面を通る粒子流の評価を行い、次に表面を通過するかあるいは反射するかの決定を行う。表面を通過する粒子に対しては、importance sampling によってスプリッティング、追跡中止、などの事象を決定する。生き残った粒子は TRACK ルーチン群で再度処理される。ランダム・ウォークの追跡の終わった粒子については、各種の統計量を計算する。スプリッティングによって子粒子が粒子バンクに貯えられている場合には、BANKIT ルーチン群で粒子バンクから子粒子を一つずつ取り出してソース粒子と同じ処理を行う。子粒子がない場合には、TALSHF ルーチンで粒子束及び分散の計算をするためにセル別、時間別、エネルギー別にデータの編集を行う。

粒子のランダム・ウォークを追跡していく過程で、稀に粒子の位置とセルとの間の関係に矛盾が生じることがある。この場合には粒子を迷子粒子とみなし、同じ条件で粒子を再サンプリングしてより精度の高い幾何形処理系のアルゴリズムを使用してランダム・ウォークの再追跡が行われる。

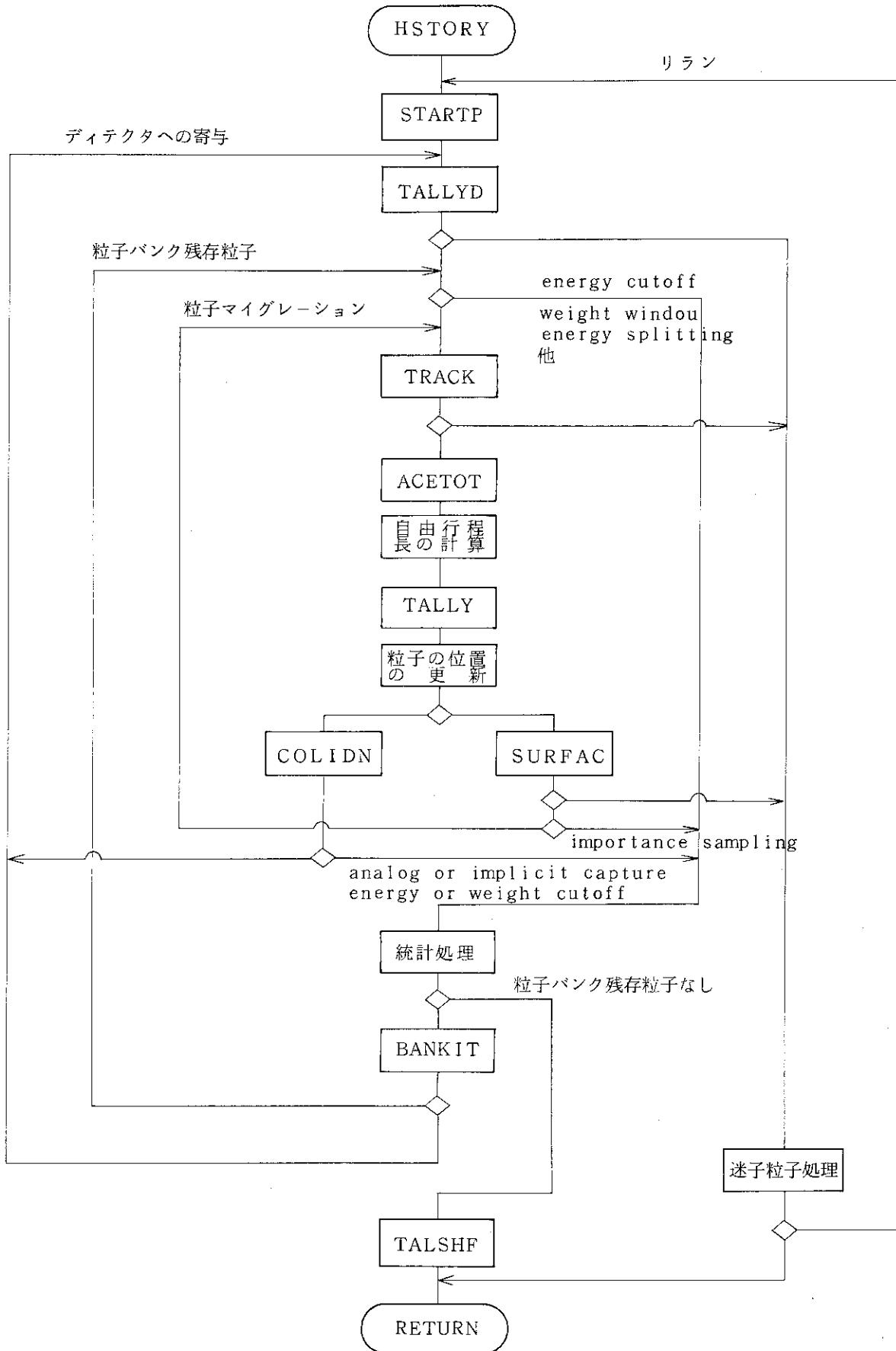


Fig. 5 The control structure of random walk control routine HSTORY.

4. ベクトル化の方法

コードの動的挙動解析ツール FORTUNE⁵⁾によってサブルーチン毎のコスト（処理時間に比例する量）比率を解析し、高いコスト比率をしめるルーチンとそれに関連するルーチンの一部をベクトル化の対象として選択する。ベクトル化対象ルーチン名及び、その機能をTable 2に示す。ベクトル化されたMCNPコードを以降ベクトル版と呼ぶ。

モンテカルロ・コードをベクトル化するときの一般的問題点として次のような事柄が挙げられる。

- ① 粒子の『死』によってベクトル長が急速に減少したり、小数の粒子が長く生き延びるために平均ベクトル長が低下し、ベクトル計算機のハードウェアを有効に利用出来ない。
- ② 主記憶の使用量が増大する。
- ③ オリジナル版とベクトル版では乱数列の違いのため粒子の振る舞いが異なる。
(ただし、MCNPコードにおいては粒子のサンプリング時に個々の粒子に対し乱数を設定し、この値を初期値としてランダム・ウォークをするため乱数列はオリジナル版、ベクトル版とも同じである。)
- ④ コードのベクトル化のための改造によるオーバーヘッドによる性能低下が大きい。
(ベクトル版スカラ・モードの性能の低下)

ベクトル化の手法は浅井らが臨界安全解析コード KENO IV²⁾のベクトル化の際に使用したイベント・バンク方式を採用した。以下では、この方法を解説する。

4.1 事象処理用粒子バンクの導入

イベント・バンク方式を使用してベクトル化を行うためには、粒子のランダム・ウォーク中に生じる事象を定義しなければならない。事象とは粒子が媒質中をランダム・ウォークしてゆく過程で必要となる断面積の計算、セルの検索、反応後の状態の決定などを指す。粒子の振る舞いは互いに独立であるため行動は個々の粒子で様々であるが、ある時刻において同一のイベントを持つ粒子をその事象に対応した事象処理用粒子バンクと呼ばれる粒子溜りに集めると、同一事象バンクの粒子に対しては同じ処理を施せばよいため、粒子に関するベクトル処理が可能となる。

粒子に関するベクトル処理の例をFig.6に示す。この例では(a)がオリジナル版の粒子の位置更新計算部分で(b)がそれに対するベクトル版での位置更新計算部分である。ここで(b)のベクトル版ソース中に見られる配列TRKBKが事象処理用粒子バンクであり粒子番号が保存されている、またNTRKは同バンク内にある粒子の数を表す。ベクトル版では、複数粒子を並列に処理するために粒子についてのDOループがかかっている。複数の粒子を区別するため粒子の属性を表す変数は配列化（この配列をベクトル変数と呼ぶ）され個々の粒子の属性に対するアクセスは粒子にあらかじめ割りつけられている粒子番号を使って行う。このためDOループ中の粒子の物理属性へのアクセスは事象処理用バンクTRKBKを介して間接的に行っている。この間接アドレッシング用の配列を特にリスト・ベクトルと呼ぶ。

Table 2 Names, functions and FORTUNE cost of vectorized routines.

ルーチン名	機能概要	コスト比 (%)
ACECOL	弹性, 非弹性散乱を解析する	0.1
ACENU	核分裂反応から生じる平均中性子数を計算する	0.1 以下
ACETOT	粒子のエネルギーに対応する媒質の断面積を計算する	7.8
ANGL	粒子の方向余弦と通過する表面の法線ベクトルとのなす角度を計算する	0.4
CALCPS	DETECTOR 方向に散乱される粒子の確率密度を計算する	1.2
CHKCEL	粒子の位置とセル番号の間に矛盾がないかどうかを調べる	1.5.2
COLIDN	衝突解析処理の制御を行う	0.5
DDDET	衝突地点から DETECTOR までの距離及び方向余弦を計算する	0.8
HISTORY	粒子のランダム・ウォーク過程を制御する	6.0
JBIN	物理量蓄積領域中の使用すべき場所のインデックスをサーチする	2.1
NEWCEL	粒子が次に入るセルを検索する	1.7
SURFAC	セルの表面を通過して次のセルに移る粒子に対して IMPORTANCE SAMPLING, SURFACE SPLITTING の処理を行う	0.5
TALLY	粒子束評価のために粒子のセル内でのトラック長の累積を行う	4.7
TALLYD	DETECTOR に対する寄与を累積する	3.5
TALSHF	個々のヒストリーの累積データを全体のヒストリーの累積データのテーブルに足し込む	6.9
TRACK	粒子のセル内での飛程長を計算する	29.8
TRANSM	散乱点から DETECTOR までの減衰因子を計算する	2.7
TRNSPT	中間結果の出力の制御を行う	0.1
合計		84.0

・他に比較的大きなコスト比率をしめるルーチンとしてLGEVAL (8.3%) があるが、これはベクトル化不可能であった。

(a)

```

D=MIN(PMF,DLS)
DT=D/VEL
TME=TME+DT
XXX=XXX+UUU*D
YYY=YYY+VVV*D
ZZZ=ZZZ+WWW*D
EGO=ERG

```

(b)

```

*VOCL LOOP,NOVREC
DO 100 JJ=1,NTRK
  II=TRKBK(JJ)
  D$(II)=MIN(PMF$(II),DLS$(II))
  DT$(II)=D$(II)/VEL$(II)
  TME$(II)=TME$(II)+DT$(II)
  XXX$(II)=XXX$(II)+UUU$(II)*D$(II)
  YYY$(II)=YYY$(II)+VVV$(II)*D$(II)
  ZZZ$(II)=ZZZ$(II)+WWW$(II)*D$(II)
  EGO$(II)=ERG$(II)
100 CONTINUE

```

Fig. 6 An example of vectorization for particle transport processing.

4.2 新しい粒子の逐次サンプリング

並列に処理可能な粒子の最大数をN個とする。始めに、N個の粒子をサンプリングして1～N番の粒子番号を付ける。ベクトル版では最初N個の粒子が同時にランダム・ウォークすることになるが、体系から漏れたり、核種との反応で吸収されたり、あるいは各種の分散低減法の操作によってランダム・ウォークの追跡が終了したりして死粒子が発生しアクティブな粒子数が減少する。粒子数の減少はベクトル処理効率の低下をまねくため『死』粒子と同数の新しい粒子をサンプリングし、つねにアクティブな粒子数がN個に近くなるように保つ。

Fig.7に新しい粒子の逐次サンプリングの流れを示した。死粒子がスプリッティングによる子粒子を中性子バンクにもっている場合には、子粒子に死んだ親粒子の粒子番号を継承させ中性子バンクから取り出し再使用する。他方、子粒子を持たない場合には、次の新しい粒子に死粒子の粒子番号を継承させてサンプリングする。

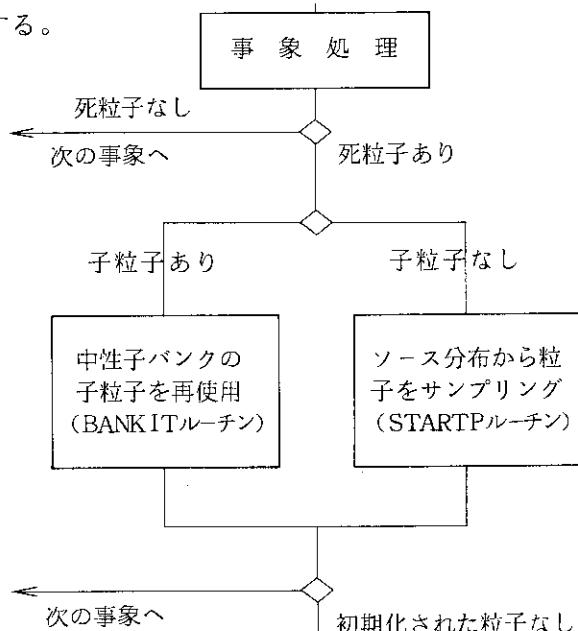


Fig. 7 The control structure of successive sampling for new source particles.

4.3 粒子数が多い事象の優先処理

ベクトル版において、事象処理バンク中の粒子の数は事象処理時の D O ループのベクトル長に関係している。このため事象処理バンク中の粒子の数が多い程ベクトル計算機の性能を引き出すことができるうことになる。ベクトル版において複数粒子のランダム・ウォークでは粒子が各事象処理用粒子バンクに散らばってしまう。散らばり方が均一でないため事象処理の順序は Fig.8 に示すように、事象 A 用のバンクと事象 B 用のバンクの粒子数を比較し、粒子数の多い事象バンクの処理を優先的に行うアルゴリズムを採用した。

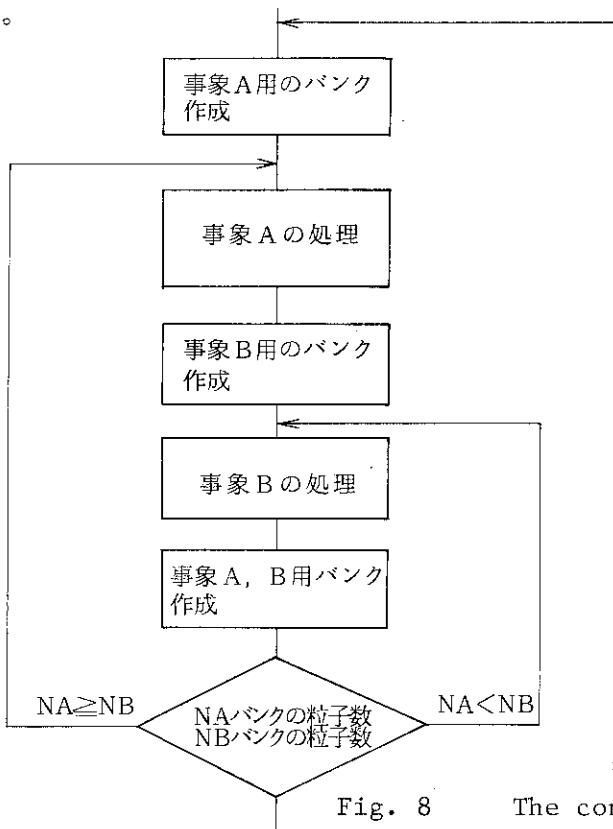


Fig. 8 The control structure of effective selection method for event bank.

4.4 例外処理部分の孤立化

モンテカルロ・コードには例外処理部分が多い。Fig.9の(a)にその一例を示した。ルーチン TRNSM 1 (ルーチン TRNSM のベクトル版での呼び名) は処理過程で迷子になった粒子であることを識別するためのフラグ K D B ¥に迷子粒子の場合は 1 そうでない場合には 0 をセットする。事象処理用粒子バンク WRKBK 内に迷子粒子がある場合には、迷子粒子用の事象処理バンク LSTBK を作成し、WRKBK の再編成を行う。しかしながら、迷子粒子は稀にしか発生しないため、LSTBK の作成及び WRKBK の再構成はほとんど必要がない。

V P システムのコンパイラ (V P コンパイラと以降呼ぶ) は、処理効率を高めるために I F 文用マスクのセット、レジスタ結合用の命令の発生及び変数のアドレス計算を D O ループの実行に先立って行う。このために、稀にしか実行されない部分を含む D O ループの場合には、逆にこのような計算が毎回無駄に実行されることになる。そこで、Fig.9の(b)に示すように D O ループ 1 3 4 で例外処理の存在をチェックし、例外処理が存在しない場合にはレジスタ結合命令の発生、変数のアドレス計算などの無駄な処理をさせないように例外処理部分を D O ループ 1 3 5 として分離した。

(a)

```

IF(NTRN.LE.0) GO TO 133
CALL TRNSM1(DD$,S$)
133 CONTINUE
NW=0
*VOCL LOOP,NOVREC
DO 135 JJ=1,NWRK
II=WRKBK(JJ)
IF(KDB$(II).NE.0)THEN
KDB$(II)=KDB$(II)+2
NLST=NLST+1
LSTBK(NLST)=II
ELSE
NW=NW+1
WRKBK(NW)=II
END IF
135 CONTINUE
NWRK=NW

```

(b)

```

IF(NTRN.LE.0) GO TO 133
CALL TRNSM1(DD$,S$)
133 CONTINUE
NR=0

DO 134 JJ=1,NWRK
II=WRKBK(JJ)
IF(KDB$(II).NE.0)NR=NR+1
134 CONTINUE

IF(NR.NE.0)THEN
NW=0
*VOCL LOOP,NOVREC
DO 135 JJ=1,NWRK
II=WRKBK(JJ)
IF(KDB$(II).NE.0)THEN
KDB$(II)=KDB$(II)+2
NLST=NLST+1
LSTBK(NLST)=II
ELSE
NW=NW+1
WRKBK(NW)=II
END IF
135 CONTINUE

NWRK=NW
END IF

```

Fig. 9 An example of exceptional events isolation from DO loops.

4.5 事象処理部分のベクトル化

ベクトル版モンテカルロ・コードにおいては事象の処理が終わるたびに処理後の粒子の状態に従って対応する事象バンクへの粒子の分類を頻繁に行っている。Fig.10は事象分類部分のベクトル化の一例を示したものである。ここでは、ある事象処理終了後に、次に通過するセル表面までの距離DLS￥と粒子の自由行程表 P M F￥を比較して次に反応処理事象に行くのか表面処理事象に行くのかを判断してCLDBK, SFCBKを作成している。このような事象処理分類を行う（リスト・ベクトルを作成する）部分も条件によっては高速にベクトル処理することが可能である。IF文のネスト構造の多重重度が低く、かつ分類すべき粒子数（DOループの繰り返し回数）が10以上のときにはベクトル処理するようにした。

```

DO 100 JJ=1,NTRK
    II = TRKBK(JJ)
    IF(PMF$(II).LT.DLS$(II)) THEN
        NCID=NCID+1
        CLDBK(NCID)=II
    ELSE
        NSFC=NSFC+1
        SFCBK(NSFC)=II
    ENDIF
100      CONTINUE

```

Fig. 10 An example of vectorization for event branch.

4.6 幾何形状分類を伴う処理のベクトル化

今回の入力データではTRACK（セル内での粒子の飛程を計算する）およびCHKCL（粒子の空間座標とセルの間の矛盾をチェックする）という二つのルーチンがC P Uタイムの約半分を消費している。これらのルーチンは粒子が存在するセルを構成するすべての面に対して、粒子の飛跡と面との交点までの距離や粒子の存在する点と面との位置関係の判断（粒子は面が作る正の領域に存在するのか、負の領域に存在するのかの判断）を行い、それらを総合してセル内での粒子の飛程を計算したり、粒子の空間座標とセルの間の矛盾をチェックしている。このため、これらのルーチンは6種類26タイプの幾何形状の要素に対する分類処理部分と各幾何形状要素に対する処理部分を持っているという点で同じ構造であると言うことができる。制御構造をFig.11の(a)に示す。

ベクトル版のアルゴリズムをFig.11の(b)に示す。並列に処理されている個々の粒子は一般的には別々のセル内に存在する。それぞれのセルを構成する面の数は一般的には等しくない。このためまず最初に、それぞれの粒子が存在するセルを取り囲む一番目の面の幾何形状のタイプ（6種類、26タイプ）ごとに粒子バンクを作成する。次に、各粒子バンクごとに処理を行う。最後に、各粒子ごとに処理すべき表面の数が異なるために、未処理の表面を持つ粒子だけを集め粒子バンクWRKBKの再構成を行う。以上のような処理をWRKBK内の粒子がなくなるまでくりかえす。

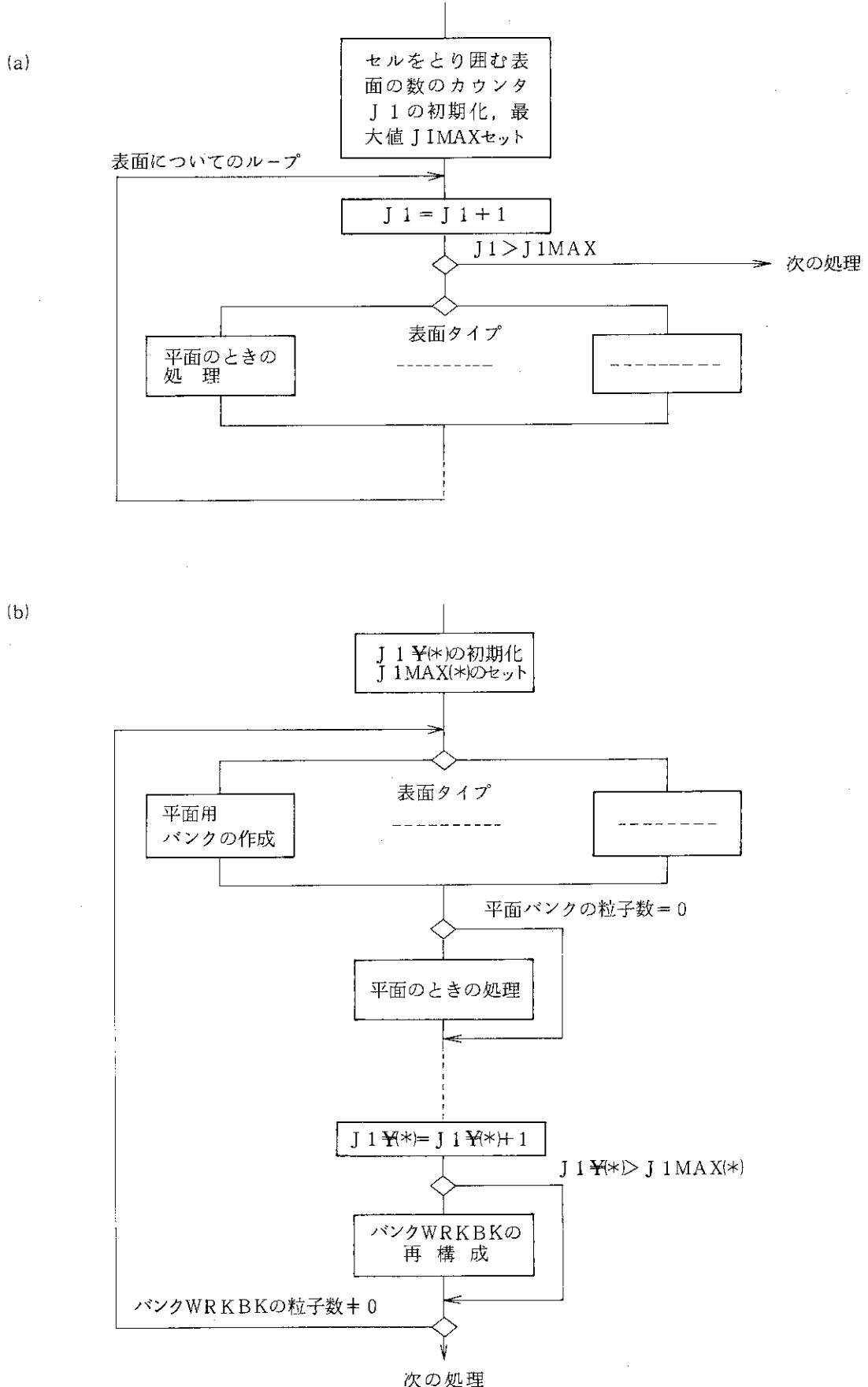


Fig. 11 The control structure in TALLY, TALLYD for the original version in (a) and vectorized version in (b).

4.7 ベクトル変数とスカラ変数間のインターフェース整合

ベクトル化は粒子の輸送計算を行うMCRUNルーチン群の一部のルーチンに対して行われた。このために、オリジナル版ルーチンとベクトル版ルーチンが混在している。従って、粒子の属性を表す変数については同じ意味を持つスカラ変数とベクトル変数が存在することになる。例えば粒子のx座標はオリジナル版のルーチン中では変数XXXであるが、ベクトル版のルーチン中では粒子に粒子番号を付けて管理しなければならないために配列XXX￥(I)である。ここでIは粒子番号である。これらのスカラ変数とベクトル変数間の橋渡しはFig.12に示すようなインターフェース整合用ルーチンで行った。作成したインターフェース整合用ルーチンの機能をTable 3に示す。

```

SUBROUTINE ITFOC SUBN , BANK , NBK )
C
COMMON/PBLCOM/ XXX,YYY,ZZZ,.....
COMMON/PBLCM$/ XXX$(250),YYY$(250),ZZZ$(250),...
DIMENSION BANK(NBK)

DO 1000 JJ=1,NBK
    II = BANK(JJ)
    / PBLCM$ /
    XXX = XXX$(II)
    YYY = YYY$(II)
    ZZZ = ZZZ$(II)
    .....
    CALL SUBN
    / PBLCM$ /
    XXX$(II) = XXX
    YYY$(II) = YYY
    ZZZ$(II) = ZZZ
    .....
1000    CONTINUE
END

```

Fig. 12 An example of interface routine in order to match vectorized subroutine and original one.

Table 3 Names, and functions of interface subroutines for common data area
in order to match the vectorized variables and the scalar variables.

ルーチン名	機能概要
ITFO	引数の無いサブルーチン用のインターフェース整合用ルーチン
ITF1I	引数の一つあるサブルーチン用のインターフェース整合用ルーチン
ITFADV	サブルーチン ADV I J K のためのインターフェース整合用ルーチン
ITFBNK	" BANKIT "
ITFEXP	" EXPIRE "
ITFOMC	" OMCFE "
ITFTRS	" TORUS "
ITFWML	" WTMULT "
ITFWWD	" WTWNDO "
ITFFUN	" FISP "
YYSTAR	" STARTP "

5. ベクトル化の結果と評価

今回は18個のルーチンをベクトル化し、あらたに11個のインターフェース・ルーチンを作成し、データの整合性を取るために13個のルーチンに若干の修正を施した。ベクトル化したルーチンのオリジナル版ベースでのステートメント数は約5,000（コメントを含む）でMCNPコード全体の約13%にあたる。ベクトル化版、オリジナル版について種々の測定を行った。ここでは、その測定結果をまとめ、今回用いたベクトル化の手法の評価を行う。

5.1 ベクトル化の結果

ベクトル化作業の妥当性を評価するために、2章で述べた入力データでオリジナル版とベクトル版をVP-100のスカラ・モードとベクトル・モードで実行し出力結果を比較した。

(1) オリジナル版とベクトル版の実行結果の比較

オリジナル版とベクトル版の結果を比較するとベクトル版は反応では核分裂が若干少なく、(n, Xn)反応が逆に若干多い。粒子の『死』の原因別では、体系外への漏れによる『死』が若干少なく、逆にweight cut offによる『死』はやや多い。この原因は次のようなものである。MCNPコードはベクトル化の改造によって各粒子の乱数列が変化しない構造になっているため、オリジナル版とベクトル版での粒子の振る舞いは原則的には同じはずである。しかし、VPコンパイラはベクトル演算器を効率的に使用できるようにDOループ内での演算順序を入れ換えるため、計算される物理量が微妙に異なる。このため、オリジナル版とベクトル版の粒子の振る舞いはごく微少ではあるがずれを生じ、ごく小数の粒子はそれが原因となってオリジナル版の粒子とは全く異なった振る舞いをする。

中性子束や中性子流などの個々の粒子の寄与を累積することによって得られる物理量では1%程度異なり、セル別、エネルギー別、タイム別の個々の累積データでは数%程度オリジナル版とは異なる。これらの累積によって計算される物理量が異なる大きな原因として総和の順序及び中間結果の出力のタイミングがベクトル版とオリジナル版で異なることがあげられる。これらの問題に関してはベクトル版MCNPコードを実用に供する前に検討が必要かと思われる。

(2) ベクトル版の実行性能および使用資源

オリジナル版とベクトル版のVP-100上での実効性能、使用資源をTable 4に示す。なお、実行時の条件は次のようなものである。

- ① 入力データはGOZILLA（付録Aに記載）
- ② ベクトル化版スカラ・モード、オリジナル版実行時のデータはFORTRAN77コンパイラをOPTIMIZE3で使用して作成したロード・モジュールによる。
- ③ ベクトル版の欄において*印なしのデータは並列に処理出来る粒子の最大数を250個、*印付きのデータは並列に処理出来る粒子の最大数を500個として実行した。
- ④ 括弧内の数値はオリジナル版の処理時間、主記憶使用量を基準とした時の対応する比率を示す。ベクトル版のオリジナル版から見た処理速度の向上比は約1.3倍、ベクトル版のスカラ・モードから見た処理速度比は約2倍（250粒子並列処理で2.1倍、500粒子並列処理で2.3倍）である。また実行時の主記憶使用量はオリジナル版と比較すると250粒子並列処理で約5倍、500粒子並

列処理で約 8.6 倍に増加した。増加分の大部分はベクトル化のために配列化されたり、配列の次元が上がったデータ・エリアによるものである。

Table 4 Resources required for executing the vectorized and original version of MCNP code.

	オリジナル版		ベクトル版
	スカラ・モード	スカラ・モード	ベクトル・モード
処理時間 ⁺ (秒)	47.9 (1.0)	78.4 (0.61) 84.6*(0.56)	37.7 (1.27) 36.4*(1.31)
主記憶使用量 ⁺⁺ (M byte)	1.4 (1.0)	7. (5.0) 13.* (8.6)	7. (5.0) 13.* (8.6)

+ 括弧内の数値はオリジナル版を基準とした処理速度比
++ " " " メモリ増加量

* 500粒子並列処理を示す

次にプログラムの動的挙動解析ツール FORTUNE⁵⁾を使用して、ベクトル化版のサブルーチンのコスト比率、平均ベクトル長、ベクトル化率などを測定した（250粒子並列処理、入力データは GOZILLA、FORTUNE は COST(2)オプションで使用）。その測定結果を Table 5 に示す。オリジナル版ベースでコード全体での CPU タイムの約 9.3% を消費するルーチンがベクトル化の対象となり、コード全体のベクトル化率は約 7.6% であった。ここで、コスト比率はルーチンに費やす CPU タイムのコード全体の CPU タイムに対する比率に相当するものである。また、ベクトル化率はベクトル命令で実行される部分 CPU タイムの全 CPU タイムに対する比率である。

Table 5 Summary of FORTUNE cost ratios, vectorization ratios and vector lengths of the vectorized routines.

ルーチン名	コスト率 (%)	ベクトル化率 (%)	主な平均
			ベクトル長
TRAC	28.2	85.6	25, 38, 63
CHKCEL	20.3	95.3	24, 30, 38
ACETOT	7.7	58.5	20~33, 63
TALLYD	6.7	87.5	52, 58, 79, 155
TALLY	5.9	51.0	72, 110
TALSHF	4.1	99.7	12
HISTORY	3.2	60.2	21, 35, 79, 81, 110
TRNSM1	1.8	98.7	38~51, 79
NEWCL1	1.8	24.3	24~54
JBIN	1.5	99.6	99~110
CALCPS	0.6	99.5	12, 47, 55
ANGL	0.6	51.5	8, 12
SURFAC	0.5	64.7	32, 65, 82
DDDET	0.2	84.6	79
TRNSPT	0.1以下	91.6	250

- ・ プログラム全体でのベクトル化率は約 7.6% である。
- ・ 並列に処理できる粒子の最大数を 250 個とした場合

5.2 結果の評価

前節でも述べたようにベクトル版はオリジナル版に比べて約1.3倍高速化された。これは、4章のベクトル化の方法で述べたソース粒子の逐次サンプリング、粒子数の多い事象の優先的処理、例外処理の孤立化及び事象分類部分のベクトル化の効果が現れたためである。以下では、これらの効果とベクトル版の問題点について解説する。

(1) ソース粒子の逐次サンプリングの効果

Fig.13及びTable 6は250粒子を一度にサンプリングし全ての粒子の処理が終了した後に次の250粒子サンプリングするアルゴリズムのベクトル化（ベクトル版1と呼ぶ、Fig.13の(a)）と新しい粒子を逐次サンプリングするアルゴリズムのベクトル版（ベクトル版2と呼ぶ、Fig.13の(b)）のアクティブな粒子の数の推移と、主なルーチンでの平均ベクトル長の増減を表している。

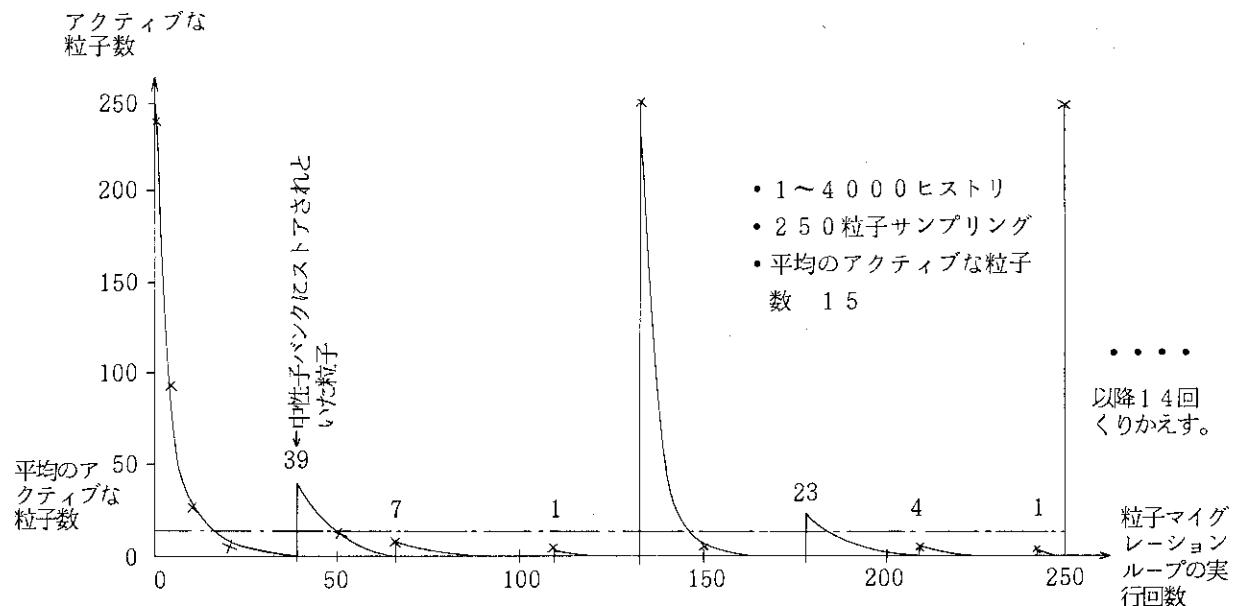
Fig.13の(a), (b)より、平均のアクティブな粒子数は、ベクトル版1が15に対してベクトル版2が141で、約9倍になっていることがわかる。また、Table 6に示すように、おもなルーチンの平均ベクトル長も4～9倍になっている。ベクトル・モードがスカラ・モードよりも高速に実行されるために必要とする最低粒子数を15とすると、Fig.13の(a), (b)よりこの個数以下で実行される回数はベクトル版1で約1,500回、ベクトル版2で約70回で約1/20以下である。今回の入力データでは、Fig.13(b)の死粒子の総数の推移で示すように実行回数あたり250個中約38個の割合で急速に粒子が死んでゆく。このような系では、死粒子が発生次第逐次新しい粒子をサンプリングし継ぎ足すベクトル版2の方法の有効性は明白である。ベクトル版1からベクトル版2へアルゴリズムを変更することによって約30%のCPUタイムを減らすことが出来た。

Table 6 Vector lengths of typical vectorized routines for algorithm 1 and algorithm 2.

	ベクトル化版1の 平均ベクトル長	ベクトル化版2の 平均ベクトル長
TRACK	9～12	34～54
CHKCEL	7～8	26～30
HSTORY	11～18, 250	35～148, 250
TALLYD	7～8	44～56
TALLY	14～16	94～148
ACETOT	6～12	24～57
TRANSM	8～11	31～56

(並列に処理できる粒子の最大数を250個としたとき)

(a)



(b)

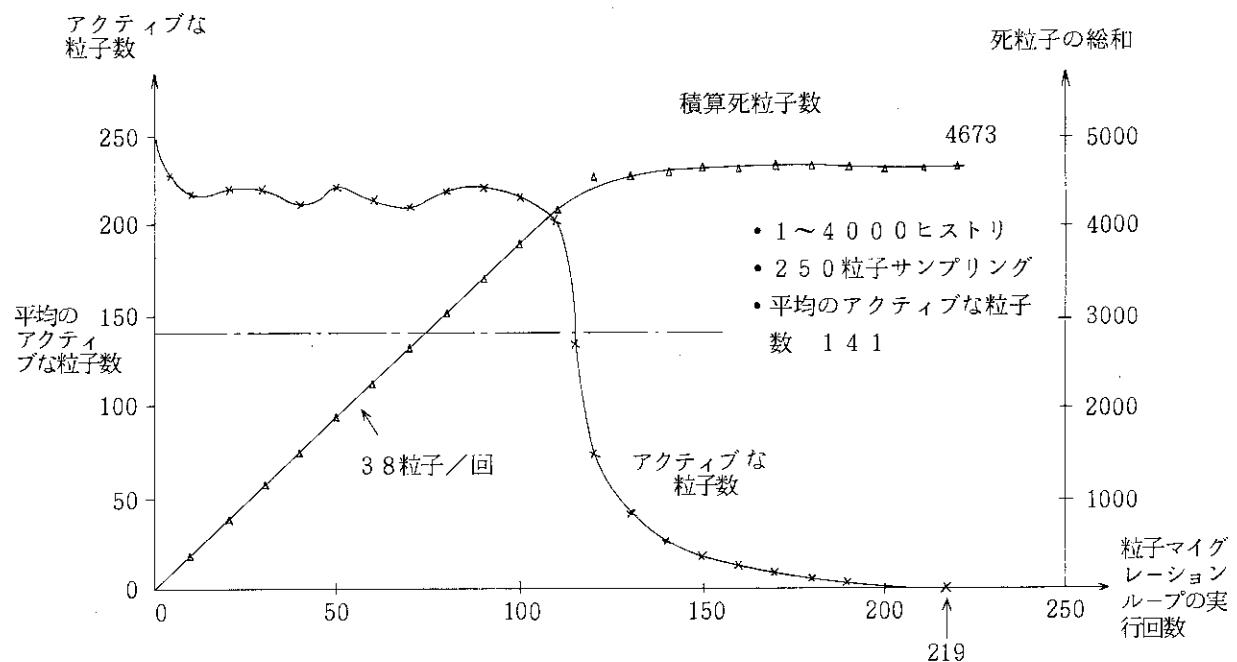


Fig. 13 Transition of active particle number during the random walk for batch source particle sampling method in (a) and successive source particle sampling method in (b).

(2) 粒子数の多い事象の優先処理の効果

粒子数の多い事象から優先的に処理する方法の採用によって、Table 6に示したベクトル版2の主要ルーチンの平均ベクトル長はさらに3~13程度長くなった。これはスカラ・モードでの効果は無いがベクトル・モードで実行する場合には効果が現れた。

(3) 例外処理事象の孤立化

コードの全体のCPUタイムの大半を消費する3~4つのルーチンは、例外処理部分、事象処理部分が多いため、例外処理部分の孤立化、事象処理部分のベクトル化も25%のCPUタイムを減らし、効果があった。

(4) ベクトル化による高速化の困難な高コスト・ルーチンLGEVALの存在

LGEVALルーチンはオリジナル版のFORTUNEでの解析結果のコスト比率で8.2%をしめる高コスト・ルーチンであるが、その構造の複雑さからベクトル化による高速化が困難なため（全実行文24行中IF文が12行）スカラ処理を行っている。

(5) ベクトル化改造のもたらすオーバーヘッドによる性能低下

4章で述べたようにベクトル化においてはコードの制御構造の大幅な改造が必要となる。モンテカルロ法においては、この改造によって多大な計算のオーバーヘッドが生じるためベクトル版のスカラ・モードでの実効性能の低下が生じる。ベクトル・モードのパイプライン演算器を使用した演算時間の短縮は、改造によって計算時間の増加したベクトル版を使用してなされるわけであるから、オーバーヘッドによる性能低下を極力避なければならない。Table 2からベクトル版のスカラ・モードでの実行時間はオリジナル版の実行時間と比較して1.6~1.8倍になっていることがわかる。このためベクトル版スカラ・モードのベクトル版ベクトル・モードの性能比が2.1~2.3であるのにオリジナル版からみると約1.3倍の性能向上しか得られないことになる。

このオーバーヘッドの原因は次のように分類される。

- ① 変数の配列化、間接アドレスの多用によるデータ・アクセス時のアドレス計算量の増加
- ② DOループの制御構造によるもの（アドレスの設定等）
- ③ オリジナル版の条件ブランチに対応するベクトル版のリスト・ベクトルの作成による演算量の増加

①、②のオーバーヘッドのユーザ側での対処法は4.2節で解説した例外処理の孤立化でDOループ内でのむだなアドレス計算等を除く方法が効果的であるが、配列化された変数に間接アドレスでアクセスする際のアドレスの計算量を本質的に減らすことはできない。③に関しては、分類処理の条件ブランチのネスト構造が単純であり分類処理のループ長が適当に大きい場合にはベクトル処理しても効率を上げることが可能であるが、オリジナル版の単純な条件ブランチから比較すると多大なオーバーヘッドと言える。このオーバーヘッドをさらに減らすための提案として浅井らのgeometric pipeline²⁾（幾何形状分類処理用）、event pipeline（事象分類処理用）という二種類のリスト・ベクトル生成用ハードウェアの提案がある。

geometric pipelineを使用するとセルごとの粒子の分類処理も高速に行なえるため、今まで回帰的なデータの定義・参照をともなう演算であるためにスカラ演算で行っていた粒子束の計算部分などのestimatorの部分が条件しだいではベクトル演算による高速化が可能となる。ベクトル版MCNPコードではTable 7に示すように幾何形状分類処理、事象分類処理および回帰的データの定義・参照のためにベクトル化をあきらめていた部分が全体の処理時間の約23.7%をしめており

geometric pipeline, event pipeline の適用はベクトル化の改造によってもたらされたオーバーヘッドの低減に対して有効であると考えられる。

Table 7 CPU time classified by the calculation type of DO loop.

V/S*	分類	CPU時間比(%)**	備考
V	一般ベクトル演算	25.1	
V	event branch	11.2	
V	search	2.8	
V	geometric branch	2.5	
S	一般スカラ演算	41.6	
S	recurrence	10.0	+
S/V	interface	0.8	++
S	初期設定、結果の 編集・出力		+++
		6.0	

* Vはベクトル演算, Sはスカラ演算を示す

** 全体のCPU時間に対する個々の項目のしめる割合

+ 短ベクトル長、ベクトル化困難な外部手続きが原因となるスカラ演算のループ

++ 回帰的なデータの定義・参照関係が原因のスカラ演算

+++ インターフェース整合用ルーチンのオーバーヘッド

6. おわりに

汎用の連続エネルギー・モンテカルロ・コードMCNPのベクトル化をベクトル計算機FACOM VP-100上で行った。ベクトル版のオリジナル版に対する実効性能は約1.3倍、そしてベクトル版ベクトル・モードでのベクトル版スカラ・モードに対する実効性能は約2.3倍である。

MCNPコードのベクトル化においては、新しい粒子の逐次サンプリング、例外処理の孤立化、事象処理部分のベクトル化が有効であった。

モンテカルロ・コードのベクトル化ではコードの改造時に持ち込まれる計算のオーバーヘッドのためにベクトル版のスカラ・モードの性能低下が生じる。このオーバーヘッドによるベクトル版のスカラ・モードでの性能低下がモンテカルロ法の高速計算をさまたげている大きな原因の一つとなっており、ベクトル処理システムのハードウェアに対する今後の課題と考えられる。

geometric pipeline, event pipeline の適用はベクトル化の改造によってもたらされたオーバーヘッドの低減に対して有効であると考えられる。

Table 7 CPU time classified by the calculation type of DO loop.

V/S*	分類	CPU時間比(%)**	備考
V	一般ベクトル演算	25.1	
V	event branch	11.2	
V	search	2.8	
V	geometric branch	2.5	
S	一般スカラ演算	41.6	
S	recurrence	10.0	+
S/V	interface	0.8	++
S	初期設定、結果の 編集・出力		+++
		6.0	

* Vはベクトル演算, Sはスカラ演算を示す

** 全体のCPU時間に対する個々の項目のしめる割合

+ 短ベクトル長、ベクトル化困難な外部手続きが原因となるスカラ演算のループ

++ 回帰的なデータの定義・参照関係が原因のスカラ演算

+++ インターフェース整合用ルーチンのオーバーヘッド

6. おわりに

汎用の連続エネルギー・モンテカルロ・コードMCNPのベクトル化をベクトル計算機FACOM VP-100上で行った。ベクトル版のオリジナル版に対する実効性能は約1.3倍、そしてベクトル版ベクトル・モードでのベクトル版スカラ・モードに対する実効性能は約2.3倍である。

MCNPコードのベクトル化においては、新しい粒子の逐次サンプリング、例外処理の孤立化、事象処理部分のベクトル化が有効であった。

モンテカルロ・コードのベクトル化ではコードの改造時に持ち込まれる計算のオーバーヘッドのためにベクトル版のスカラ・モードの性能低下が生じる。このオーバーヘッドによるベクトル版のスカラ・モードでの性能低下がモンテカルロ法の高速計算をさまたげている大きな原因の一つとなっており、ベクトル処理システムのハードウェアに対する今後の課題と考えられる。

謝 辞

M C N P コードのベクトル化に際して、ソース・コードをはじめとして各種資料を提供していただきました原子炉システム研究室 森貴正氏に感謝します。

また付録Bの粒子の軌跡図を提供していただいた財原子力データセンターの小松正也氏、さらに本作業にあたり一貫してご援助していただいた財原子力データセンターの西敬二郎氏、富士通株の奥田基氏に感謝します。

富士通株の牧野光弘氏には本報告書執筆にあたり有効なアドバイスをいただきました、感謝します。

また、図版の整理などに協力してくださった富士通株の山田圭子さんに感謝します。

参 考 文 献

- (1) 原子力コード研究会：“原子力におけるソフトウェア開発研究会報告書”，JAERI-M 85-017 (1985).
- (2) K. Asai et al : “Vectorization of the KENO-IV code”. Nuclear Science and Engineering 92, 298-307 (1986).
- (3) M.H. Kalos : “The NYU Ultracomputer”. 1983, Mathematics and Computation Topical Meeting of the american Nuclear Society.
- (4) Los Alamos Radiation Transport Group (X-6) : “Reference for MCNP and the manual”, LA-7386-M (1981).
- (5) 富士通株式会社：“FACOM OS IV/F4 MSP FORTUNE 使用手引 V10用”，78SP-5360.

謝 辞

M C N P コードのベクトル化に際して、ソース・コードをはじめとして各種資料を提供していただきました原子炉システム研究室 森貴正氏に感謝します。

また付録Bの粒子の軌跡図を提供していただいた財原子力データセンターの小松正也氏、さらに本作業にあたり一貫してご援助していただいた財原子力データセンターの西敬二郎氏、富士通株の奥田基氏に感謝します。

富士通株の牧野光弘氏には本報告書執筆にあたり有効なアドバイスをいただきました、感謝します。

また、図版の整理などに協力してくださった富士通株の山田圭子さんに感謝します。

参 考 文 献

- (1) 原子力コード研究会：“原子力におけるソフトウェア開発研究会報告書”，JAERI-M 85-017 (1985).
- (2) K. Asai et al : “Vectorization of the KENO-IV code”. Nuclear Science and Engineering 92, 298-307 (1986).
- (3) M. H. Kalos : “The NYU Ultracomputer”. 1983, Mathematics and Computation Topical Meeting of the american Nuclear Society.
- (4) Los Alamos Radiation Transport Group (X-6) : “Reference for MCNP and the manual”, LA-7386-M (1981).
- (5) 富士通株式会社：“FACOM OS IV/F4 MSP FORTUNE 使用手引 V10用”，78SP-5360.

付録A 入力データ

本文2章で説明した入力データ GOZILLA を Fig.A-1 に示す。

```

1-      MCNP TEST PROBLEM
2-      C      1 TO 12: DETECTORS
3-          1 3 1.0E-2 -1      91 -92
4-          2 3 1.0E-2 -2      91 -92
5-          3 3 1.0E-2 -3      91 -92
6-          4 3 1.0E-2 -4      91 -92
7-          5 3 1.0E-2 -5      91 -92
8-          6 3 1.0E-2 -6      91 -92
9-          7 3 1.0E-2 -7      91 -92
10-         8 3 1.0E-2 -8      91 -92
11-         9 3 1.0E-2 -9      91 -92
12-        10 3 1.0E-2 -10     91 -92
13-        11 3 1.0E-2 -11     91 -92
14-        12 3 1.0E-2 -12     91 -92
15-      C      101: SAMPLE, 102: SPACE ARROUND SAMPLE ,103: GRAPHITE REFLECTOR
16-          101 1 0.01 -20 91 -92
17-          102 0      20 -31 -32 33 34 91 -92
18-          103 2 0.01 (31:32:-33:-34)
19-                  -41 -42 43 44 91 -92
20-          104 0      (41:42:-43:-44)
21-                  -51 -52 53 54
22-                  1 2 3 4 5 6 7 8 9 10 11 12
23-                  91 -92
24-          105 0      (51:52:-53:-54:-91:92)
25-
26-      C      DETECTORS
27-          1 C/Z   100.0 -50.0 10.
28-          2 C/Z   100.0  0.0 10.
29-          3 C/Z   100.0  50.0 10.
30-          4 C/Z   50.0 100.0 10.
31-          5 C/Z   0.0 100.0 10.
32-          6 C/Z   -50.0 100.0 10.
33-          7 C/Z   -100.0 50.0 10.
34-          8 C/Z   -100.0  0.0 10.
35-          9 C/Z   -100.0 -50.0 10.
36-         10 C/Z   -50.0 -100.0 10.
37-         11 C/Z   0.0 -100.0 10.
38-         12 C/Z   50.0 -100.0 10.
39-      C      SAMPLE
40-          20 C/Z   0.0  0.0 40.
41-      C      INNER WALL OF GRAPHITE REFLECTOR.
42-          31 PX    50.0
43-          32 PY    50.0
44-          33 PX   -50.0
45-          34 PY   -50.0
46-      C      OUTER WALL OF GRAPHITE REFLECTOR.
47-          41 PX    75.0
48-          42 PY    75.0
49-          43 PX   -75.0
50-          44 PY   -75.0
51-      C      OUTER WALL OF THE SYSTEM.
52-          51 PX    150.0
53-          52 PY    150.0
54-          53 PX   -150.0
55-          54 PY   -150.0
56-      C      BOUNDARY FOR Z-DIRECTION.
57-          91 PZ   -150.0
58-          92 PZ    150.0

```

```

59-
60-      IN      1 15R 0
61-      C
62-      C      << SOURCE ( D-T NEUTRON SPECTRUM ) >>
63-      C
64-      C      SRC5 ~150.1 0.0 0.0 104 1. 300. 1.0 0.0 0.0 105 53 J
65-      C      VOID
66-      SRC1  45. 45. 0. 102 1.0
67-      SI    10.616 11.663 12.479  13.007  13.480  13.969
68-      SP    14.477 15.003 15.548
69-      SP    D .0000 .0039431 .0039366 .0047753 .0078840 .0141736
70-      SP    .1062637 .5221151 .3369086
71-      C
72-      C      << MATERIAL >>
73-      C
74-      M1    92238  1.0 92235  0.01  94239  1.0 8016  20.0
75-      M2    6012   1.0
76-      M3    2003   1.0
77-      C
78-      C      << TALLY >>
79-      C
80-      F4    1 2 3 (1 2 3) (4 5 6) (7 8 9) (10 11 12) 101 102 103 104
81-      F4    (1 2 3 4 5 6 7 8 9 10 11 12)
82-      F5    100. 0.0 0.0 0.0
83-      C
84-      C      << ENERGY >>
85-      C
86-      ERGN  0.0 16.0
87-      EO    0.4E-6  1.0 5.0 10.0 14.0 16.0
88-      C
89-      C      << TIME >>
90-      C
91-      T4    1.0 10.0 100.0 1000.0 1000000.0
92-      C
93-      C      << CUT OFF >>
94-      C
95-      CUTN  1.0E6 1.0E-7 3J
96-      NPS   12000
97-      CTME 10.0
98-      PRDMP 4000 4000
99-

```

Fig. A-1 A sample input data (GOZILLA).

付録B 粒子の軌跡

ベクトル化版において、入力データとしてGOZILLA(付録A参照のこと)を使ったときの粒子の振る舞いを図形表示した。MCNPコードの場合ベクトル化版／オリジナル版双方とも乱数の進行は同一であるので粒子は同じ動きをする。1～50番目の粒子の軌跡をFig.B-1に示す。粒子は点状の粒子源からほぼ等方的に発生していることがわかる。核分裂性物質内では散乱を起こし粒子の飛行方向が不規則に変化しており、反射体では一部の粒子が反射されている。

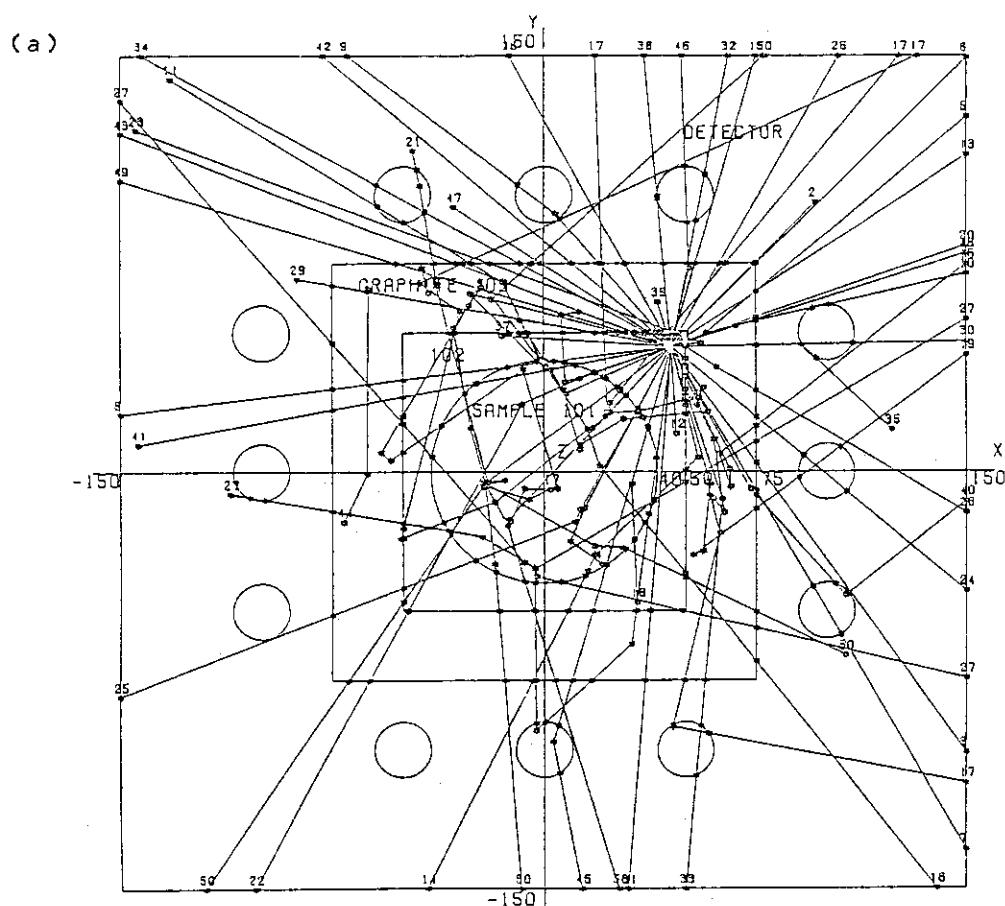


Fig. B-1 Trajectories of particles in the geometrical structure of top view in (a)

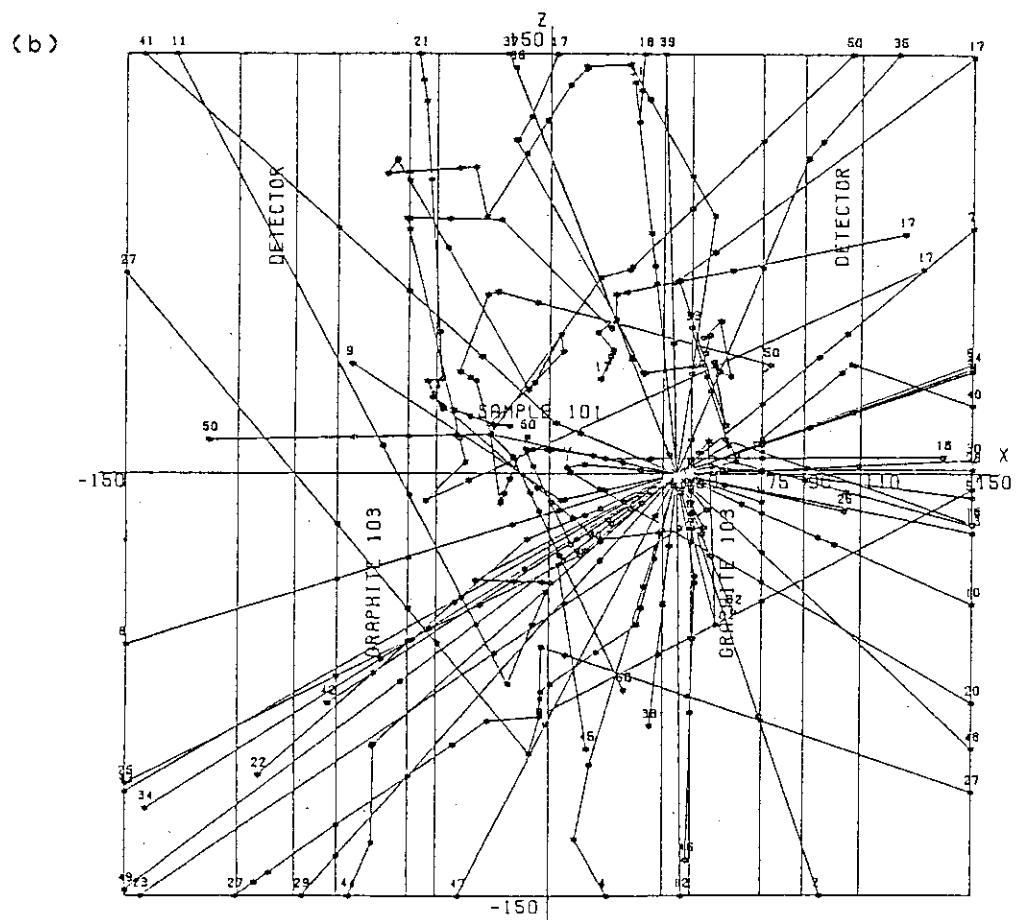


Fig. B-1 Trajectories of particles in the geometrical structure of top view in (b).

また、核分裂性物質内で分裂を起こした粒子の軌跡の一例をFig. B-2 に示す。

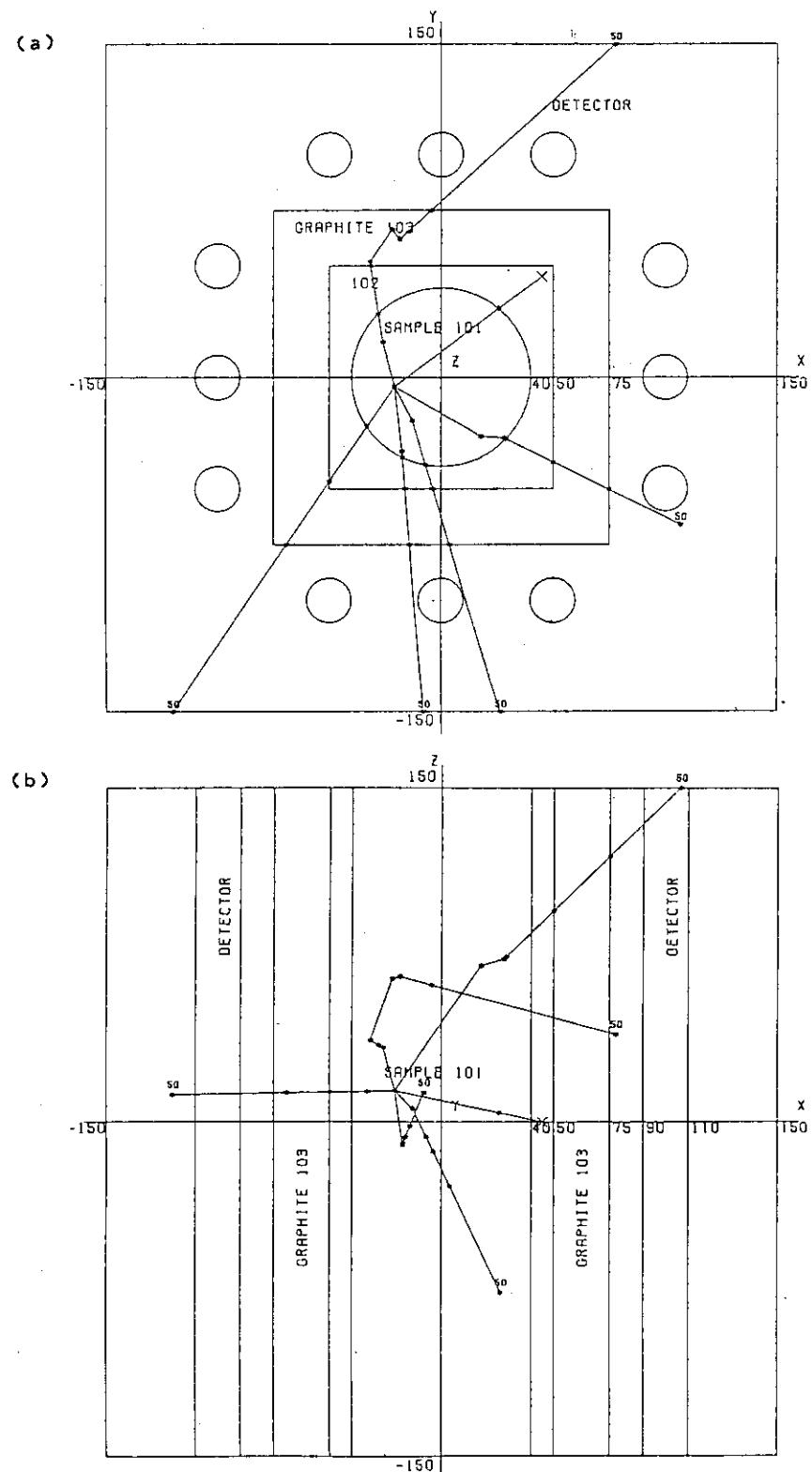


Fig. B-2 Trajectories of particles which causes fission reaction in the geometric structure of top view in (a) and side view in (b).

付録C コードの木構造

M C N P コードの木構造を以下に示す。粒子の輸送計算を行なうMCRUNルーチン配下で、1つ1つの粒子の一生を追跡するHISTORYルーチン群については全構造を示した。この木構造において、二重下線=を引いたルーチンはベクトル化したルーチンである。

```
MAIN  ---+--IMCN
      I
      +--PLOT
      I
      +--XACT
      I
      +--MCRUN
```

Fig. C-1 Vectorized routines (underlined) in the tree structure at the top level of MCNP code.

```
MCRUN -----SECOVL---+--PDPROB
      I
      +--TRFMAT
      I
      +--ITALLY
      I
      +--VOLUME
      I
      +--TRNSPT-----HISTORY
      I
      +--OUTPUT
```

Fig. C-2 Vectorized routines (underlined) in the tree structure at MCRUN routine level.

```

HSTORY--+-STARTP
    I
    +-+TALLYD
    I
    +-+DXTRAN
    I
    +-+TRACK
    I
    +-+ACETOT
    I
    +-+PHOTCT
    I
    +-+TALLY
    I
    +-+SURFAC
    I
    +-+COLIDN
    I
    +-+COLIDP

```

Fig. C-3 Vectorized routines (underlined) in the tree structure at HSTORY routine level.

```

STARTP-----%SETRAN
    +-+ISOS -----RANG
    +-+SOURCE-----EXPIRE
    +-+CHKSRC-----NAMCHG
    I           +-+EXPIRE
    I           +-+CHKCEL-----ANGL
    I           +-+LGEVAL
    +-+SOURCA----SRC SMP-----RANG
    I           +-+CBIAS -----RANG
    I           I           +-+ROTA S -----RANG
    I           +-+RANG
    I           +-+ROTA S -----RANG
    I           +-+NEWCEL-----EXPIRE
    I           I           +-+CHKCEL-----ANGL
    I           I           +-+LGEVAL
    I           +-+ISOS -----RANG
    I           +-+CHKCEL-----ANGL
    I           I           +-+LGEVAL
    I           +-+EXPIRE
    +-+SOURCK
    +-+EXPIRE
    +-+EVENTP

```

Fig. C-4 Vectorized routines (underlined) in the tree structure at STARTP routine level.

TRACK ----- TORUS ----- QUART
I +--RPZERO-----EXPIRE
I +--CPZERO----#FC
I +-#FR
I +-#FI
I +-#EVAL ----#FC
I +-#FR
I +-#FI
I +-#ZA
+--LGEVAL

Fig. C-5 Vectorized routines (underlined) in the tree structure at TRACK routine level.

ACETOT ----- ACENU ----- ACEFCN

Fig. C-6 Vectorized routines (underlined) in the tree structure at ACETOT routine level.

```

TALLY -----DOSEF
    +-+JBIN -----BEYOND
    +-+ANGL
    +-+CHKCEL -----ANGL
    I           +-+LGEVAL
    +-+TRACK --
    I
    I
    I
    +-+WTMULT-----ACETOT-----ACENU -----ACEFCN
    I           +-+GETXS -----ACEFCN
    I           I           +-+ACENU -----ACEFCN
    I           +-+PHOTOT
    +-+TALLYX
    +-+BEYOND
    +-+EXPIRE
    +-+REVERT

```

Fig.C-5と同じ

Fig. C-7 Vectorized routines (underlined) in the tree structure at TALLY routine level.

```

SURFAC-----TALLY --
    I
    I
    I
    +-+ANGL
    +-+CHKCEL -----ANGL
    I           +-+LGEVAL
    +-+NEWCEL -----EXPIRE
    I           +-+CHKCEL -----ANGL
    I           I           +-+LGEVAL
    +-+WTWNDO-----BANKIT-----EXPIRE
    I           I           +-+FASTIO
    I           +-+RANG
    +-+RANG
    +-+BANKIT-----EXPIRE
           +-+FASTIO

```

Fig.C-7と同じ

Fig. C-8 Vectorized routines (underlined) in the tree structure at SURFAC routine level.

```

DXTRAN----RANG
    +-+ROTAS -----RANG
    +-+CALCPS-----CHKCEL-----ANGL
    I           I           +-+LGEVAL
    I           +-+SRCDX
    +-+TRANSM-----CHKCEL-----ANGL
    I           I           +-+LGEVAL
    I           +-+RANG
    I           +-+TRACK -+
    I           I           Fig.C-5 と同じ
    I           I
    I           I
    I           +-+ACETOT-----ACENU -----ACEFCN
    I           +-+PHOTOT
    I           +-+NEWCEL-----EXPIRE
    I           +-+CHKCEL-----ANGL
    I           +-+LGEVAL
    +-+BANKIT-----EXPIRE
    +-+FASTIO

```

Fig. C-9 Vectorized routines (underlined) in the tree structure at DXTRAN routine level.

```

PHOTC -----RANG
    +-+ACEFCN
    +-+ACECAS-----RANG
    I           +-+ACEFCN
    I           +-+ACETBL
    I           +-+ACENU -----ACEFCN
    I           +-+IFISP
    I           +-+FISP -----RANG
    I           +-+ACECOS-----RANG
    I           +-+EXPIRE
    +-+ISOS -----RANG
    +-+ACECOS-----RANG
    +-+ROTAS -----RANG

```

Fig. C-10 Vectorized routines (underlined) in the tree structure at PHOTC routine level.

```

COLIDN-----RANG
    +-+ACEGAM
    +-+BANKIT-----EXPIRE
    I           +-+FASTIO
    +-+COLIDK-----RANG
    I           +-+ACECOL-----RANG
    I           I           +-+ISOS      -----RANG
    I           I           +-+ACECOS-----RANG
    I           I           +-+ROTA S -----RANG
    I           I           +-+ACENU      -----ACEFCN
    I           I           +-+ACECAS-----RANG
    I           I           I           +-+ACEFCN
    I           I           I           +-+ACETBL
    I           I           I           +-+ACENU      -----ACEFCN
    I           I           I           +-+IFISP
    I           I           I           +-+FISP       -----RANG
    I           I           I           +-+ACECOS-----RANG
    I           I           I           +-+EXPIRE
    I           I           +-+EXPIRE
    +-+SABCOL-----RANG
    I           +-+EXPIRE
    I           +-+ROTA S -----RANG
    +-+ROTA S -----RANG
    +-+ISOS      -----RANG
    +-+ACECOL-----RANG
    I           I           +-+ISOS      -----RANG
    I           I           +-+ACECOS-----RANG
    I           I           +-+ROTA S -----RANG
    I           I           +-+ACENU      -----ACEFCN
    I           I           +-+ACECAS-----RANG
    I           I           I           +-+ACEFCN
    I           I           I           +-+ACETBL
    I           I           I           +-+ACENU      -----ACEFCN
    I           I           I           +-+IFISP
    I           I           I           +-+FISP       -----RANG
    I           I           I           +-+ACECOS-----RANG
    I           I           I           +-+EXPIRE
    I           I           +-+EXPIRE
    +-+EVENTP

```

Fig. C-11 Vectorized routines (underlined) in the tree structure at COLIDN routine level.

```

COLIDP-----RANG
    +-+KLEIN      -----RANG
    +-+ROTA S -----RANG
    +-+ISOS      -----RANG
    +-+BANKIT-----EXPIRE
    I           +-+FASTIO
    +-+EVENTP

```

Fig. C-12 Vectorized routines (underlined) in the tree structure at COLIDP routine level.

```

TALLYD-----RANG
    +-+OMCFE -----ACETOT-----ACENU -----ACEFCN
    I           +-+PHOTOT
    I           +-+CALCPS-----CHKCEL-----ANGL
    I           I           I           +-+LGEVAL
    I           I           +-+SRCDX
    I           +-+RANG
    I           +-+ROMAS -----RANG
    I           +-+TRANSM-----CHKCEL-----ANGL
    I           I           I           +-+LGEVAL
    I           I           +-+RANG
    I           I           +-+TRACK --+
    I           I           I           Fig.C-5と同じ
    I           I           I
    I           I           +-+ACETOT-----ACENU -----ACEFCN
    I           I           +-+PHOTOT
    I           I           +-+NEWCEL-----EXPIRE
    I           I           +-+CHKCEL-----ANGL
    I           I           +-+LGEVAL
    I           +-+COLIDN-- Fig.C-11と同じ
    I           I           (ただし、ルーチンは全てオリジナル版)
    I           I
    I           +-+COLIDP-- Fig.C-12と同じ
    I
    I
    +-+DDDET -----RANG
    +-+CALCPS-----CHKCEL-----ANGL
    I           I           +-+LGEVAL
    I           +-+SRCDX
    +-+BEYOND
    +-+TRANSM-----CHKCEL-----ANGL
    I           I           +-+LGEVAL
    I           +-+RANG
    I           +-+TRACK --+
    I           I           Fig.C-5と同じ
    I           I
    I           +-+ACETOT-----ACENU -----ACEFCN
    I           +-+PHOTOT
    I           +-+NEWCEL-----EXPIRE
    I           +-+CHKCEL-----ANGL
    I           +-+LGEVAL
    +-+JBIN   -----BEYOND
    +-+DOSEF
    +-+WTMULT-----ACETOT-----ACENU -----ACEFCN
    I           +-+GETXS -----ACEFCN
    I           I           +-+ACENU -----ACEFCN
    I           +-+PHOTOT
    +-+REVERT

```

Fig. C-13 Vectorized routines (underlined) in the tree structure at TALLYD routine level.