



JAEA-Data/Code

2016-018

DOI:10.11484/jaea-data-code-2016-018

MVP/GMVP Version 3 : General Purpose Monte Carlo Codes for Neutron and Photon Transport Calculations Based on Continuous Energy and Multigroup Methods

Yasunobu NAGAYA, Keisuke OKUMURA, Takeshi SAKURAI and Takamasa MORI

Nuclear Science and Engineering Center
Sector of Nuclear Science Research

March 2017

Japan Atomic Energy Agency

日本原子力研究開発機構

JAEA-Data/Code

本レポートは国立研究開発法人日本原子力研究開発機構が不定期に発行する成果報告書です。
本レポートの入手並びに著作権利用に関するお問い合わせは、下記あてにお問い合わせ下さい。
なお、本レポートの全文は日本原子力研究開発機構ホームページ (<http://www.jaea.go.jp>)
より発信されています。

国立研究開発法人日本原子力研究開発機構 研究連携成果展開部 研究成果管理課
〒319-1195 茨城県那珂郡東海村大字白方 2 番地4
電話 029-282-6387, Fax 029-282-5920, E-mail:ird-support@jaea.go.jp

This report is issued irregularly by Japan Atomic Energy Agency.
Inquiries about availability and/or copyright of this report should be addressed to
Institutional Repository Section,
Intellectual Resources Management and R&D Collaboration Department,
Japan Atomic Energy Agency.
2-4 Shirakata, Tokai-mura, Naka-gun, Ibaraki-ken 319-1195 Japan
Tel +81-29-282-6387, Fax +81-29-282-5920, E-mail:ird-support@jaea.go.jp

© Japan Atomic Energy Agency, 2017

MVP/GMVP Version 3 : General Purpose Monte Carlo Codes for Neutron and Photon Transport Calculations Based on Continuous Energy and Multigroup Methods

Yasunobu NAGAYA, Keisuke OKUMURA, Takeshi SAKURAI and Takamasa MORI

Nuclear Science and Engineering Center
Sector of Nuclear Science Research
Japan Atomic Energy Agency
Tokai-mura, Naka-gun, Ibaraki-ken

(Received December 19, 2016)

In order to realize fast and accurate Monte Carlo simulation of neutron and photon transport problems, two Monte Carlo codes MVP (continuous-energy method) and GMVP (multigroup method) have been developed at Japan Atomic Energy Agency. The codes have adopted a vectorized algorithm and have been developed for vector-type supercomputers. They also support parallel processing with a standard parallelization library MPI and thus a speed-up of Monte Carlo calculations can be achieved on general computing platforms.

The first and second versions of the codes were released in 1994 and 2005, respectively. They have been extensively improved and new capabilities have been implemented. The major improvements and new capabilities are as follows: (1) perturbation calculation for effective multiplication factor, (2) exact resonant elastic scattering model, (3) calculation of reactor kinetics parameters, (4) photo-nuclear model, (5) simulation of delayed neutrons, (6) generation of group constants.

This report describes the physical model, geometry description method used in the codes, new capabilities and input instructions.

Keywords: Monte Carlo Method, Transport Calculation, Neutron, Photon, Continuous Energy, Multigroup, Nuclear Data Library, Vectorization, Parallel Computation, User's Manual

MVP/GMVP 第3版：連続エネルギー法及び多群法に基づく 汎用中性子・光子輸送計算モンテカルロコード

日本原子力研究開発機構 原子力科学研究部門 原子力基礎工学研究センター
長家 康展・奥村 啓介・櫻井 健・森 貴正

(2016年12月19日受理)

高速かつ高精度な中性子・光子輸送モンテカルロ計算を実現するため、日本原子力研究開発機構において、2つのモンテカルロコード MVP(連続エネルギー法) と GMVP(多群法) が開発されてきた。これらのコードはベクトル型アルゴリズムを採用し、ベクトル計算機用に開発されてきたが、標準並列化ライブラリー MPI を用いた並列計算にも対応しており、一般の計算機環境でもモンテカルロ計算の高速化が可能である。両コードは正確な物理モデル、詳細な幾何形状表現法、分散低減法等、実用コードとして十分な機能を有している。

これらコードの第1版は1994年、第2版は2005年に公開され、それ以降も様々な改良と機能拡張が行われてきた。第2版公開以降の主な改良点と新機能は(1)実効増倍率に対する摂動計算手法、(2)厳密共鳴弾性散乱モデル、(3)動特性パラメータ計算機能、(4)光核反応モデル、(5)遅発中性子のシミュレーション、(6)多群定数生成機能である。

本報告書では2つのコードで用いられている物理モデル、幾何形状表現法、新たな機能及びそれらの使用方法が記載されている。

Contents

1	Introduction	1
2	Basics of Monte Carlo Methods	4
2.1	Particle Transport Calculation	4
2.2	Particle Transport with Monte Carlo Methods	7
2.3	Spatial Description	8
2.3.1	Spatial Description by Combinatorial Geometry	8
2.3.2	Repeated Geometry (LATTICE Geometry)	8
2.4	Boundary Conditions	10
2.5	Treatment of Nuclear Reactions	11
2.6	Estimation of Physical Quantities	13
2.6.1	Collision Estimator	13
2.6.2	Track Length Estimator	13
2.6.3	Point Detector Estimator	14
2.6.4	Surface Crossing Estimator	15
2.7	Statistical Uncertainty	15
2.8	History and Batch	17
2.9	Variance Reduction Techniques	17
2.9.1	Russian Roulette	19
2.9.2	Splitting	20
2.9.3	Path Stretching (Exponential Transform)	20
2.9.4	Options for Variance Reduction	22
2.10	Fixed-source and Eigenvalue Problems	22
3	Outline of Codes	24
3.1	Functions	24
3.2	Vector Calculation	25
3.3	Parallel Calculation	26
4	Physics Model	29
4.1	Representation of Cross Sections	29
4.1.1	Neutron Cross Sections	29
4.1.2	Photoatomic Cross Sections	30
4.1.3	Photonuclear Cross Sections	31
4.1.4	Preparation of Neutron Cross Sections at Arbitrary Temperatures	31
4.2	Analysis of Neutron Collision	32
4.2.1	Selection of Collision Nuclide	32
4.2.2	Generation of Fission Neutrons	34
4.2.3	Absorption	34
4.2.4	Scattering	35
4.2.5	Thermal Neutron Scattering	36
4.2.6	Resonance Elastic Scattering	37
4.2.7	Speed of Neutrons	39
4.3	Estimation of Eigenvalue	39

4.4	Estimation of Real Variance in Eigenvalue Problems	40
4.5	Reactor Noise Analysis	42
4.6	Photon Reaction	43
4.6.1	Detailed and Simple Models	45
4.6.2	Photoelectric Effect	45
4.6.3	Electron Pair Production	45
4.6.4	Incoherent (Compton) Scattering	45
4.6.5	Coherent (Thomson) Scattering	46
4.6.6	Bremsstrahlung Photon	47
4.6.7	Analysis of Photonuclear Reactions	47
4.6.8	Speed of Photons	48
5	MVP/GMVP Geometry	49
5.1	Combinatorial Geometry	49
5.2	BODIEs	49
5.3	Definition of ZONES	50
5.3.1	Operation of BODY Combination	50
5.3.2	Unsupported BODY Operation	51
5.3.3	OR at the Head of BODY Operations	51
5.3.4	INPUT-ZONE and ZONE	52
5.3.5	Multiply Defined ZONE	53
5.3.6	Identification of ZONES and Undefined Regions	54
5.3.7	Finiteness of ZONES	54
5.4	Specification of Materials and Boundary Conditions	54
5.5	Repeated Geometry	55
5.5.1	Concept of LATTICE Geometry	55
5.5.2	Hierarchical Structure of Nested LATTICE Geometry	56
5.5.3	Definition of CELLS	59
5.5.4	Definition of LATTICEs and Coordinate Transformation	59
5.5.5	Specification of CELLS Placed at SUBFRAMEs	63
5.5.6	Allocation of LATTICE at FRAME ZONE	66
5.5.7	Shape of FRAME ZONE	66
5.6	Statistical Geometry Model	68
5.6.1	How to Input the Statistical Geometry Model	69
5.7	Hierarchical Expression of REGION Names	70
5.8	TALLY REGION	71
5.9	REGION Name and REGION Number	72
6	Input Instructions	74
6.1	Features	74
6.2	Structure of Input Data	74
6.3	Effective Columns and Comments in a Line	75
6.4	Option Block	76
6.5	Format of Input Data Description	76
6.5.1	Labeled Free Format Input	77
6.5.2	Data Block	77
6.5.3	Labeled Value and Arithmetic Operation	78
6.5.4	Repeated Values and Equally Spaced Values	80

6.5.5	Operations and Functions Applicable to Data Series	81
6.5.6	Reference of Variables Defined in the Codes as Symbolic Parameters	82
6.6	Data Input by REGION	82
6.6.1	REGION-dependent Data	83
6.6.2	Input Format - 1	84
6.6.3	Input Format - 2	84
6.6.4	Specification of REGIONs by Wild Card Characters * and ?	85
7	Title and Options	86
7.1	Title for Input Data	86
7.2	Rule for Option Block	86
7.3	List of Available Options	86
7.4	Conflict between Options	91
7.5	Detailed Information for Each Option	93
8	Calculation Control	115
8.1	Control Data for Both MVP and GMVP	115
8.2	Control Data Only for MVP	123
8.3	Input Data for PICTURE Option	125
9	Cross Section and Material Data	127
9.1	Input Data for MVP	127
9.1.1	CROSS SECTION or XSEC Data Block	127
9.1.2	Specification of Compositions (IDMAT and Number Density)	127
9.1.3	Specification of Nuclide ID in Photon Problems	129
9.1.4	Use of Photonuclear Reaction Cross Sections	129
9.1.5	Use of Arbitrary-temperature Neutron Cross Sections	130
9.2	Options for MVP Cross Section Data	131
9.3	Input Data for GMVP	134
9.3.1	Cross Section Form	134
9.3.2	CROSS SECTION Data Block	135
9.3.3	Input Parameters of Multigroup Cross Sections	135
9.3.4	Specification of Material Compositions	144
10	Geometry Data	146
10.1	Structure of Geometry Data	146
10.2	Definition of BODies	146
10.2.1	BODY RPP (Right Parallelepiped)	149
10.2.2	BODY CYL (Cylinder in <i>z</i> -direction)	149
10.2.3	BODY RCC (Right Circular Cylinder)	150
10.2.4	BODY SPH (Sphere)	151
10.2.5	BODY BOX (Parallelepiped)	152
10.2.6	BODY WED or RAW (Wedge)	152
10.2.7	BODY ARB (Arbitrary Polyhedron with 4,5 or 6 Faces)	153
10.2.8	BODY HAF (Half Space with a Planer Surface)	154
10.2.9	BODY RHP (Right Hexagonal Prism)	155
10.2.10	BODY HEX (Hexagonal Prism)	155
10.2.11	BODY RCL (Right Cylinder with Specified Radial Direction)	156
10.2.12	BODY TRC (Truncated Right Cone)	157

10.2.13 BODY TEC (Truncated Elliptic Cone)	158
10.2.14 BODY ELL (Spheroid)	159
10.2.15 BODY GEL (General Ellipsoid)	160
10.2.16 BODY ELT (Elliptical Torus)	160
10.2.17 BODY GQS (General Quadratic Curved Surface)	161
10.2.18 BODY BBC (BODY-by-BODY Combination)	162
10.3 Input Data for ZONES	162
10.3.1 ZONE Name	163
10.3.2 REGION Name	163
10.3.3 Material ID	163
10.3.4 Combinatorial Geometry	164
10.3.5 'INPUT-ZONE' and 'ZONE'	164
10.3.6 ZONE Specification and Calculation Time	164
10.3.7 Periodic Boundary Condition	165
10.4 Input Data for LATTICE Geometry	166
10.4.1 Specification of LATTICE Structure	166
10.4.2 Examples of Input Data for LATTICE Geometry	171
10.4.3 Arrangement of LATTICES	174
10.5 Definition of CELLS	176
10.5.1 Definition of ZONES in a CELL	177
10.6 Examples for LATTICE Geometry	177
10.7 Tallying in the Same CELLS	184
10.7.1 Labeling FRAME ZONES	184
10.7.2 Labeling SUBFRAMES	184
10.8 Input Data for Statistical Geometry Model	185
10.8.1 Definition of the STGM Region	186
10.8.2 Allocation of STGM Regions	188
10.8.3 Definition of STG CELLS	188
10.8.4 Dummy CELL for MATRIX	189
10.8.5 Example for Multiple Kinds of STG Particles	189
10.9 Definition of TALLY REGIONS	191
11 Source Data	194
11.1 General Rules for Source Input Data	194
11.1.1 SOURCE Data Block	194
11.1.2 Source Information Set	194
11.2 Representation of Sampling Information	196
11.3 Values and Mathematical Expression	196
11.4 Sampling with Functions	197
11.5 Sampling Functions	199
11.6 Data-transform Functions	211
11.7 Selection of Sampled Values	214
11.7.1 ACCEPT and END-ACCEPT	214
11.7.2 Logical and Relational Functions	215
11.7.3 Acceptance/Rejection by Specified Space	216
11.8 Biased Sampling	216
11.9 Examples for Using Sampling Functions	217
11.9.1 Comparison of Interpolation Methods for #TABLE Function	217

11.9.2	Sampling Functions with One Variable (#TABLE, #COSINE, #GAUSS) and Selection of Sampled Values	218
11.9.3	Sampling Function with Two Variables: #TABLEXY	220
11.9.4	#TABLEXY, Work Variable and Selection of Sampled Values	220
12	Tally Data	223
12.1	TALLY Data Block	223
12.2	Common Specification to All Tallies	223
12.3	Input Data for Each Tally	230
12.3.1	Specification of Tally ID	230
12.3.2	Specification of Tally Label	230
12.3.3	Specification of Estimator	230
12.3.4	Specification of Tally Particle	231
12.3.5	Specification of Tally Dimensions	231
12.3.6	Specification of Tally Bins	233
12.3.7	Specification of Reaction Rate Calculation	243
12.3.8	Specification for Reactor Noise Analysis	246
12.4	Example of the TALLY Data Block	247
12.5	Volume of Spatial Regions	249
13	Variance Reduction Data	252
13.1	Available Variance Reduction Techniques	252
13.2	Variance Reduction Parameters	252
14	Secondary Particle Data	258
14.1	Generation of Secondary Particles	258
14.2	Data for Fission Neutrons and Secondary Photons	258
15	Perturbation Data	264
15.1	General Rules of Input Data for Perturbation Calculations	264
15.1.1	PERTURBATION Data Block	264
15.2	Input Data for Each Perturbation	264
15.2.1	ID	264
15.2.2	LABEL	265
15.2.3	Estimator	265
15.2.4	Tally Particle	265
15.2.5	Tally Parameter	265
15.2.6	Perturbation Parameter	266
15.2.7	Perturbation Calculation Method	266
15.2.8	Number of Generations for Perturbed Source Effect	267
15.2.9	Perturbed Quantity	267
15.2.10	Perturbed Region	267
15.2.11	Perturbed Nuclide	267
15.2.12	Example of PERTURBATION Data Block	268
16	I/O Files	271
16.1	FORTTRAN I/O Unit	271
16.2	Specification of I/O Files in a Command Line	271
16.3	Index File for MVP Cross Sections	273

16.4	Cross Section Files for GMVP	274
16.5	Printing Output	276
16.5.1	Output on Input Data Processing	276
16.5.2	Output on Random Walk	277
16.5.3	Calculation Results	279
16.5.4	Event Monitor	283
16.6	Binary Output of Calculation Results	284
16.7	Restart File	301
16.8	Fission Source File	301
17	How to Execute MVP/GMVP	303
17.1	Setting User's Environment	303
17.1.1	Setting Environment Variables	304
17.1.2	Command Search Path	304
17.1.3	Example of Shell Startup Files	304
17.2	How to Execute MVP	305
17.2.1	Files Related to MVP	305
17.2.2	Execution of MVP with the runmvp Command	305
17.3	How to Execute GMVP	309
17.3.1	Files Related to GMVP	309
17.3.2	Execution of GMVP with the rungmvp Command	309
17.4	Specification of Memory Size	311
17.4.1	Dynamic Memory Allocation	311
17.4.2	Static Memory Allocation	312
	Acknowledgments	312
	References	313
	Appendixes	
A	Installation of MVP/GMVP	317
A.1	How to Install the MVP/GMVP Code on UNIX or Linux Platforms	317
A.1.1	Extracting Files	318
A.1.2	Setting Up the Environment to Execute Programs	319
A.1.3	Compilation and Linkage	321
A.2	Preparation of Cross Section Data	324
A.2.1	Processing of MVP Cross Section Data	324
A.2.2	Processing of GMVP Cross Section Data	325
A.3	Test Calculation with Samples	326
A.3.1	Test Calculation for MVP	326
A.3.2	Test Calculation for GMVP	327
A.4	Program Source Files	329
A.5	Platform Dependence	330
A.5.1	Linux	330
A.5.2	FreeBSD	330
A.5.3	CYGWIN	330
A.5.4	MS Windows	330

B	CGVIEW	331
B.1	Output of CGVIEW Code	331
B.2	Input Data for CGVIEW Code	332
B.3	CGVIEW Commands	336
B.4	Script to execute CGVIEW	352
B.5	Problems in Using CGVIEW and Solutions	353
C	MVP-ART	355
C.1	Input Data for MVP-ART	355
C.2	Sample Input Data	356
D	Sample Programs to Read Binary Output File	359
D.1	Sample 1	359
D.2	Sample 2	361
E	Codes to Convert the Form of Cross Section Libraries	368
E.1	Conversion of MVP Libraries from Text to Binary Form	368
E.2	Conversion of MVP Libraries from Binary to Text Form	369
E.3	Conversion of File Form for GMVP Libraries	371
F	FORTTRAN-source Assimilation Tool	372
F.1	Function of MVPFAT	372
F.2	Input and Output of MVPFAT	375
F.3	Example of the Process with MVPFAT	375
F.4	Execution Command for MVPFAT	377
F.5	Parameters of SET and UNSET Commands	377
G	MVP Cross Section Libraries	379
G.1	Neutron Libraries	379
G.1.1	JENDL-4.0	379
G.1.2	JENDL-4.0u1	386
G.1.3	JENDL-4.0TZ	387
G.2	Photoatomic Data Library	387
G.2.1	MCPLIB	387
G.2.2	ENDF/B-VI.8	389
G.3	Photonuclear Data Library	391
G.4	Electron Library	395
G.5	Dosimetry Library	397
H	Nuclear Reactions in MVP libraries	400
H.1	Neutron Reactions	400
H.2	Photoatomic Reactions	401
H.3	Photonuclear Reactions	401
	Index	403
	Source Code of MVP/GMVP and Auxiliary Programs	417

目次

1	はじめに	1
2	モンテカルロ法の基礎	4
2.1	粒子輸送計算の手法	4
2.2	モンテカルロ法による粒子輸送計算の手順	7
2.3	空間の表現	8
2.3.1	組み合わせ幾何形状による空間の表現	8
2.3.2	繰り返しのある形状の表現 (格子形状)	8
2.4	境界条件	10
2.5	核反応の取り扱い	11
2.6	物理量の評価法	13
2.6.1	衝突による評価法	13
2.6.2	粒子の飛行距離による評価法	13
2.6.3	点検出器評価法	14
2.6.4	面検出器評価法	15
2.7	統計誤差	15
2.8	ヒストリーとバッチ	17
2.9	分散低減法	17
2.9.1	ロシアン・ルーレット	19
2.9.2	スプリッティング	20
2.9.3	パス・ストレッチング	20
2.9.4	分散低減オプション	22
2.10	固定源問題と固有値問題	22
3	コードの概要	24
3.1	機能	24
3.2	ベクトル計算	25
3.3	並列計算	26
4	物理モデル	29
4.1	断面積の表現	29
4.1.1	中性子断面積	29
4.1.2	光原子反応断面積	30
4.1.3	光核反応断面積	31
4.1.4	任意温度における中性子断面積の準備	31
4.2	中性子衝突反応の解析	32
4.2.1	衝突核種の決定	32
4.2.2	核分裂中性子の発生	34
4.2.3	吸収反応	34
4.2.4	散乱反応	35
4.2.5	熱中性子散乱	36
4.2.6	共鳴弾性散乱	37
4.2.7	粒子の速さ	39
4.3	固有値の評価法	39

4.4	固有値問題における真の分散の評価	40
4.5	炉雑音解析	42
4.6	光子の反応	43
4.6.1	詳細モデルと簡易モデル	45
4.6.2	光電効果	45
4.6.3	電子対生成	45
4.6.4	非干渉性(コンプトン)散乱	45
4.6.5	干渉性(トムソン)散乱	46
4.6.6	制動輻射光子	47
4.6.7	光核反応解析	47
4.6.8	光子の速さ	48
5	幾何形状の表現	49
5.1	組み合わせ形状	49
5.2	ボディ	49
5.3	ゾーンの定義	50
5.3.1	ボディ組み合わせの演算	50
5.3.2	サポートされないボディ演算	51
5.3.3	ボディ演算の先頭に置かれる OR	51
5.3.4	入力ゾーンとゾーン	52
5.3.5	空間で重なって定義されたゾーン	53
5.3.6	ゾーンの同定と未定義空間	54
5.3.7	ゾーンの有限性	54
5.4	物質の指定と境界条件	54
5.5	繰り返し形状表現	55
5.5.1	格子形状表現の概念	55
5.5.2	多重格子形状の階層構造	56
5.5.3	セルの定義	59
5.5.4	格子の定義と座標変換	59
5.5.5	サブフレームに配置されるセルの指定	63
5.5.6	格子のフレームゾーンへの配置	66
5.5.7	フレームゾーンの形状	66
5.6	確率論的幾何形状モデル	68
5.6.1	確率論的幾何形状モデルの入力方法	69
5.7	領域 (REGION) 名の階層的表現	70
5.8	タリー領域 (TALLY REGION)	71
5.9	領域 (REGION) 名と領域 (REGION) 番号	72
6	入力データの概要	74
6.1	特徴	74
6.2	入力データの構成	74
6.3	有効な文字数とコメント	75
6.4	オプション部	76
6.5	入力データの記述形式	76
6.5.1	データ名付き自由形式入力	77
6.5.2	データブロック	77
6.5.3	名前付き数値データと演算機能	78
6.5.4	データ列の繰り返しと等間隔データの作成	80

6.5.5	データ列に適用できる演算と関数	81
6.5.6	コード内変数値のシンボリックパラメータとしての参照	82
6.6	領域 (REGION) 毎に入力するデータ	82
6.6.1	領域 (REGION) に依存したデータ	83
6.6.2	入力形式 - 1	84
6.6.3	入力形式 - 2	84
6.6.4	ワイルドカード文字*と?を用いた領域 (REGION) の指定	85
7	タイトルとオプション	86
7.1	入力データのタイトル	86
7.2	オプション入力の規則	86
7.3	指定可能なオプションの一覧	86
7.4	オプションの相互関係	91
7.5	各オプションの詳細	93
8	計算のコントロール	115
8.1	MVP と GMVP に共通のコントロールデータ	115
8.2	MVP のみのコントロールデータ	123
8.3	PICTURE オプションに対する入力データ	125
9	断面積と物質組成データ	127
9.1	MVP に対する入力データ	127
9.1.1	CROSS SECTION または XSEC データブロック	127
9.1.2	物質組成の指定 (IDMAT 及び個数密度)	127
9.1.3	光子問題での核種 ID の指定	129
9.1.4	光核反応断面積の利用	129
9.1.5	任意温度中性子断面積の利用	130
9.2	MVP 断面積データに対するオプション	131
9.3	GMVP に対する入力データ	134
9.3.1	断面積形式	134
9.3.2	CROSS SECTION データブロック	135
9.3.3	多群断面積の入力パラメータ	135
9.3.4	物質組成の指定	144
10	幾何形状データ	146
10.1	幾何形状データの構造	146
10.2	ボディの定義	146
10.2.1	ボディ RPP (直方体)	149
10.2.2	ボディ CYL (z 軸に平行な中心軸を持つ直円柱)	149
10.2.3	ボディ RCC (直円柱)	150
10.2.4	ボディ SPH (球)	151
10.2.5	ボディ BOX (平行六面体)	152
10.2.6	ボディ WED または RAW (三角柱)	152
10.2.7	ボディ ARB (任意の四、五または六面体)	153
10.2.8	ボディ HAF (平面によって区切られた半空間)	154
10.2.9	ボディ RHP (中心軸が Z 軸に平行で 1 つの側面が X 軸に垂直な正六角柱)	155
10.2.10	ボディ HEX (任意の方向を向いた正六角柱)	155
10.2.11	ボディ RCL (底面に方向性を持った直円柱)	156
10.2.12	ボディ TRC (円錐台)	157

10.2.13	ボディ TEC (楕円錐台)	158
10.2.14	ボディ ELL (回転楕円体)	159
10.2.15	ボディ GEL (三軸不等楕円体)	160
10.2.16	ボディ ELT (断面が楕円のトーラス)	160
10.2.17	ボディ GQS (任意の二次曲面)	161
10.2.18	ボディ BBC (ボディの組み合わせ)	162
10.3	ゾーンの入力データ	162
10.3.1	ゾーン名	163
10.3.2	領域名	163
10.3.3	物質 ID	163
10.3.4	ボディの組み合わせ	164
10.3.5	‘入力ゾーン’ と ‘ゾーン’	164
10.3.6	ゾーンの指定方法と計算時間	164
10.3.7	周期境界条件	165
10.4	格子形状の入力データ	166
10.4.1	格子構造の指定	166
10.4.2	格子形状の入力データ例	171
10.4.3	格子の空間内への配置	174
10.5	セルの定義	176
10.5.1	セルに含まれるゾーンの定義	177
10.6	格子形状の入力例	177
10.7	同じ種類のセル内でのタリー	184
10.7.1	フレーム・ゾーンの名前付け	184
10.7.2	サブフレームの名前付け	184
10.8	確率論的幾何形状モデルの入力データ	185
10.8.1	STGM 領域の定義	186
10.8.2	STGM の空間内への配置	188
10.8.3	STG 粒子セルの定義	188
10.8.4	媒質 (マトリックス) のためのダミーセル	189
10.8.5	複数の STG 粒子の入力例	189
10.9	タリー領域の定義	191
11	線源データ	194
11.1	粒子源入力データの一般規則	194
11.1.1	SOURCE データブロック	194
11.1.2	線源セット	194
11.2	サンプリング情報の表現	196
11.3	値と式	196
11.4	関数を用いたサンプリング	197
11.5	サンプリング関数	199
11.6	データ加工指示関数	211
11.7	サンプリングされた値の選別	214
11.7.1	ACCEPT と END-ACCEPT	214
11.7.2	論理関数と関係関数	215
11.7.3	特定の空間部分による受け入れ/棄却	216
11.8	重み調節によるバイアスサンプリング	216
11.9	サンプリング関数の使用例	217
11.9.1	#TABLE 関数における補間法の比較	217

11.9.2	1 変数のサンプリング関数 (#TABLE, #COSINE, #GAUSS) と結果の選別	218
11.9.3	2 変数のサンプリング関数: #TABLEXY	220
11.9.4	#TABLEXY、作業変数と結果の選別	220
12	タリーデータ	223
12.1	TALLY データブロック	223
12.2	すべてのタリーに共通な指定	223
12.3	個別タリーの入力データ	230
12.3.1	タリー ID の指定	230
12.3.2	ラベルの指定	230
12.3.3	エスティメータの指定	230
12.3.4	タリー粒子の指定	231
12.3.5	タリー次元の指定	231
12.3.6	タリービンの指定	233
12.3.7	反応率計算の指定	243
12.3.8	炉雑音解析に対する指定	246
12.4	TALLY データブロックの入力例	247
12.5	空間領域の体積	249
13	分散低減法データ	252
13.1	利用できる分散低減法	252
13.2	分散低減パラメータ	252
14	二次粒子に関するデータ	258
14.1	二次粒子の発生	258
14.2	核分裂中性子と二次光子に関するデータ	258
15	摂動計算の入力データ	264
15.1	摂動計算入力データの一般的規則	264
15.1.1	PERTURBATION データブロック	264
15.2	各摂動パラメータに対する入力	264
15.2.1	ID	264
15.2.2	LABEL	265
15.2.3	評価法の指定	265
15.2.4	タリー粒子の指定	265
15.2.5	タリーパラメータの指定	265
15.2.6	摂動パラメータの指定	266
15.2.7	摂動計算手法の指定	266
15.2.8	核分裂源分布変化を考慮するための世代数	267
15.2.9	摂動量の指定	267
15.2.10	摂動領域の指定	267
15.2.11	摂動核種の指定	267
15.2.12	摂動計算データ入力例	268
16	入出力ファイル	271
16.1	FORTTRAN I/O Unit	271
16.2	コマンド行での入出力ファイルの指定	271
16.3	MVP 断面積のインデックスファイル	273
16.4	GMVP の断面積ファイル	274

16.5	プリント出力	276
16.5.1	入力データ処理に関する出力	276
16.5.2	ランダムウォークに関する出力	277
16.5.3	計算結果	279
16.5.4	事象モニター	283
16.6	計算結果のバイナリ出力	284
16.7	リスタートファイル	301
16.8	核分裂源ファイル	301
17	MVP/GMVP の実行方法	303
17.1	利用環境の設定	303
17.1.1	環境変数の設定	304
17.1.2	コマンドサーチパス	304
17.1.3	シェル起動ファイルの例	304
17.2	MVP の実行方法	305
17.2.1	MVP に関連するファイル	305
17.2.2	runmvp コマンドによる MVP の実行	305
17.3	GMVP の実行方法	309
17.3.1	GMVP に関連するファイル	309
17.3.2	rungmvp コマンドによる GMVP の実行	309
17.4	メモリサイズの指定	311
17.4.1	動的メモリ確保	311
17.4.2	静的メモリ確保	312
	謝辞	312
	参考文献	313
	付録	
A	MVP/GMVP のインストール	317
A.1	UNIX または Linux 上での MVP/GMVP コードのインストールの方法	317
A.1.1	ファイルの展開	318
A.1.2	実行環境の設定	319
A.1.3	コンパイルとリンク	321
A.2	断面積データの準備	324
A.2.1	MVP 断面積データの処理	324
A.2.2	GMVP 断面積データの処理	325
A.3	サンプル入力データによるテスト計算	326
A.3.1	MVP のテスト計算	326
A.3.2	GMVP のテスト計算	327
A.4	プログラムソースファイルについて	329
A.5	各機種/OS に固有の情報	330
A.5.1	Linux	330
A.5.2	FreeBSD	330
A.5.3	CYGWIN	330
A.5.4	MS Windows	330

B	CGVIEW	331
B.1	CGVIEW の出力画面	331
B.2	CGVIEW コードの入力データ	332
B.3	CGVIEW コマンド	336
B.4	CGVIEW を実行するためのスクリプト	352
B.5	CGVIEW を使用する上での問題点と対処方法	353
C	MVP-ART	355
C.1	MVP-ART の入力データ	355
C.2	サンプル入力データ	356
D	バイナリ出力ファイルを読むためのサンプルプログラム	359
D.1	例 1	359
D.2	例 2	361
E	断面積のファイル形式を変換するためのコード	368
E.1	MVP ライブラリのテキスト形式からバイナリ形式への変換	368
E.2	MVP ライブラリのバイナリ形式からテキスト形式への変換	369
E.3	GMVP ライブラリのファイル形式変換	371
F	FORTRAN ソースを使用環境に適合させるツール	372
F.1	MVPFAT の機能	372
F.2	MVPFAT の入出力	375
F.3	MVPFAT による処理例	375
F.4	MVPFAT の実行コマンド	377
F.5	SET と UNSET コマンドのパラメータ	377
G	MVP 断面積ライブラリ	379
G.1	中性子ライブラリ	379
G.1.1	JENDL-4.0	379
G.1.2	JENDL-4.0u1	386
G.1.3	JENDL-4.0TZ	387
G.2	光原子反応ライブラリ	387
G.2.1	MCPLIB	387
G.2.2	ENDF/B-VI.8	389
G.3	光核反応ライブラリ	391
G.4	電子ライブラリ	395
G.5	ドジメトリ・ライブラリ	397
H	MVP ライブラリで定義されている核反応	400
H.1	中性子反応	400
H.2	光原子反応	401
H.3	光核反応	401
	索引	403
	MVP/GMVP と付属プログラムのソースコード	417

List of Figures

Fig. 2.1	Random walk of a particle	7
Fig. 2.2	Definition of ZONES by CG	9
Fig. 2.3	Description of a repeated geometry with a LATTICE geometry	10
Fig. 2.4	Hexagonal prism LATTICE without the FREE-LATTICE-FRAME option	11
Fig. 2.5	“History” as the smallest unit of independent samples	18
Fig. 2.6	“Batch” as the unit of independent samples	18
Fig. 2.7	Russian roulette	19
Fig. 2.8	Path stretching	21
Fig. 3.1	Comparison of history- and event-based algorithms	26
Fig. 3.2	Flow of tasks in MVP/GMVP	27
Fig. 3.3	Calculation flow of parallel processing	28
Fig. 3.4	Parallel processing for eigenvalue problems	28
Fig. 4.1	Probability table of the U-235 capture cross section around 500 eV	32
Fig. 4.2	Neutron reactions in collision treated in MVP	33
Fig. 4.3	Calculation flow of the collision analysis in MVP	33
Fig. 4.4	Time spectra for different gate widths	44
Fig. 5.1	ZONE definition by combination of BODIEs	51
Fig. 5.2	Unsupported parentheses and alternative expression	52
Fig. 5.3	Equivalent OR usage	52
Fig. 5.4	Multiply defined ZONE	53
Fig. 5.5	Geometry model defined with LATTICE capability	57
Fig. 5.6	Concept of nested LATTICE geometry	58
Fig. 5.7	CELL-coordinates (X_c, Y_c, Z_c)	59
Fig. 5.8	Definition of CELL	60
Fig. 5.9	CELL array and CELL-array coordinates (X_a, Y_a, Z_a)	61
Fig. 5.10	FRAME of a right hexagonal prism LATTICE and LATTICE coordinates (X_ℓ, Y_ℓ, Z_ℓ)	64
Fig. 5.11	LATTICE coordinates (X_ℓ, Y_ℓ, Z_ℓ) of a right hexagonal prism LATTICE and CELL-array coordinates (X_a, Y_a, Z_a)	65
Fig. 5.12	Two LATTICEs with different pitches	65
Fig. 5.13	Free and conventional LATTICE FRAMEs	67
Fig. 5.14	Nested LATTICE geometry where the LATTICE FRAME crosses the CELLs	68
Fig. 5.15	Schematic view of the statistical geometry model in MVP/GMVP (STGM region and STG particle)	70
Fig. 10.1	BODY RPP	149
Fig. 10.2	BODY CYL	150
Fig. 10.3	BODY RCC	151
Fig. 10.4	BODY SPH	151
Fig. 10.5	BODY BOX	152
Fig. 10.6	BODY WED (or RAW)	153
Fig. 10.7	BODY ARB	154

Fig. 10.8	BODY HAF	155
Fig. 10.9	BODY RHP	156
Fig. 10.10	BODY HEX	157
Fig. 10.11	BODY RCL	157
Fig. 10.12	BODY TRC	158
Fig. 10.13	BODY TEC	159
Fig. 10.14	BODY ELL	159
Fig. 10.15	BODY GEL	160
Fig. 10.16	BODY ELT	161
Fig. 10.17	Splitting calculation geometry	165
Fig. 10.18	Specification of periodic boundary condition and tracking of a particle	166
Fig. 10.19	RCELL to specify the reference SUBFRAME in a hexagonal LATTICE	168
Fig. 10.20	Arrangement of CELLS for LATTICE ID 10	172
Fig. 10.21	Example of a hexagonal LATTICE with a hexagonal frame	175
Fig. 10.22	A right-parallelepiped LATTICE geometry which includes another right-parallelepiped LATTICE with a different pitch	179
Fig. 10.23	A right-parallelepiped LATTICE with the FREE-LATTICE-FRAME option	181
Fig. 10.24	A hexagonal-prism LATTICE geometry which includes another hexagonal-prism LATTICE with a different pitch	183
Fig. 10.25	An example for multiple kinds of STG particles	191
Fig. 11.1	Comparison of interpolation methods for #TABLE function (1)	218
Fig. 11.2	Comparison of interpolation methods for #TABLE function (2) : LINEAR and STEP	219
Fig. 11.3	Sampling function with a variable (selection of sampled values with #TABLE, #COSINE, #GAUSS)	220
Fig. 11.4	Result for the #TABLEXY function	221
Fig. 11.5	Distribution of X	221
Fig. 11.6	Distribution of Y for each X	221
Fig. 11.7	Result of selected sampled values	222
Fig. 11.8	Distribution of X : selected sampled values	222
Fig. 11.9	Distribution of Y for each X : selected sampled values	222
Fig. 16.1	Output of special tallies in the file of calculated results	297
Fig. B.1	Output of the CGVIEW code	332
Fig. B.2	Check a geometry with randomly generated lines - FLOOD	338
Fig. B.3	Specification of the LATTICE level to be drawn - LEVEL	340
Fig. B.4	Specification of the drawing area for CGVIEW - PAPER, SIZE	343
Fig. B.5	Rotation of the drawing area around the rotation axis perpendicular to the screen - ROT	346
Fig. B.6	Rotation of the drawing area around the rotation axis parallel to the screen - ROTX, ROTY	347
Fig. B.7	Specification of drawing lines for CGVIEW - SCAN(sx sy)	348
Fig. B.8	Zooming in/out a drawing area - ZOOM	351

List of Tables

Table 5.1	Material IDs	55
Table 5.2	Direction of the axes in the LATTICE coordinates	66
Table 6.1	Basic functions and logical/relational operators available in input data	79
Table 6.2	Variables which can be referred to as symbolic parameters	83
Table 7.1	Default values and incompatibility for options	91
Table 9.1	Data structure of the ANISN/DTF-type cross sections for each energy group	139
Table 10.1	Available BODies in MVP/GMVP	147
Table 10.2	Material ID's and materials	163
Table 10.3	Parameters in SZLAT to define CELLS	171
Table 10.4	Parameters in SZHEX to define the frame size of a hexagonal LATTICE and the relation between the CELL-array coordinates and the LATTICE coordinates	171
Table 11.1	Source parameters	196
Table 11.2	Default values for source parameters	196
Table 11.3	Sampling function	199
Table 11.4	Data-transform function	199
Table 11.5	Logical and relational functions for conditional sampling	215
Table 13.1	Variance reduction parameters that the user must specify	252
Table 15.1	List of input parameters for perturbation calculations	268
Table 16.1	I/O files common to MVP/GMVP	271
Table 16.2	I/O files of MVP	272
Table 16.3	I/O files of GMVP	272
Table 16.4	MVP cross section index files and I/O units	274
Table A.1	Content of directories related to MVP/GMVP code	319
Table A.2	Environmental variable MVPHOST for each platform	321
Table A.3	Shell scripts to prepare MVP libraries	324
Table A.4	Sample input data files for MVP	326
Table A.5	Sample input data files for GMVP	328
Table A.6	Directories for program source files	329
Table B.1	CGVIEW commands	334
Table G.1	MVP library based on JENDL-4.0 with the free gas model	379
Table G.2	MVP library based on JENDL-4.0 with ENDF/B-VI.8 thermal scattering data	386
Table G.3	MVP library based on JENDL-4.0u1 with the free gas model	387
Table G.4	MVP library based on JENDL-4.0TZ with the free gas model	387
Table G.5	MVP photoatomic data library	388
Table G.6	MVP photoatomic data library based on ENDF/B-VI.8	390
Table G.7	MVP photonuclear data library	392

Table G.8	MVP electron library	396
Table G.9	MVP dosimetry library	398
Table H.1	Neutron reactions in MVP neutron libraries	400
Table H.2	Photoatomic reactions in MVP photoatomic data libraries	401
Table H.3	Photonuclear reactions in MVP photonuclear data libraries	402

1. Introduction

The Monte Carlo method has been often used for particle transport calculations of neutrons and photons, etc. in the three-dimensional complicated systems. This method has a unique feature for constructing the detailed geometry model by using the flexible geometry description capability. In addition, the continuous-energy Monte Carlo method has characteristics that all nuclear data are given at every energy point without using the multigroup approximation, and the interaction of particles with materials is treated as exactly as possible. Such advantages motivate us to use the Monte Carlo method for obtaining highly accurate reference solutions and for performing experimental analysis and detailed design of fission reactor cores, radiation shielding, criticality safety, fusion reactor neutronics and so on.

On the other hand, large computation time is often required to obtain reliable solutions because the Monte Carlo method is the stochastic method. This has been one of the drawbacks for the Monte Carlo method. However, the rapid progress in computing power is mitigating it and drastically expands the application field of the Monte Carlo method. Japan Atomic Energy Agency (JAEA) has studied the Monte Carlo algorithms to enhance the performance of the vector and parallel supercomputers since over twenty years ago, and has also developed high-speed and highly precise Monte Carlo codes MVP (the continuous-energy method) and GMVP (the multigroup method). They are referred to as the MVP/GMVP code in the following chapters. In recent years, the scalar-type supercomputers are dominant and thus the MVP/GMVP code cannot take advantage of the event-based algorithm¹⁾ suitable for vector-type supercomputers on general computing platforms. However, the MVP/GMVP code can perform parallel calculations with a standard parallelization library MPI²⁾ and thus the speedup of the Monte Carlo calculations can be realized on the general platforms.

The first version of the MVP/GMVP code³⁾ was domestically released in 1994 and it has been used in many research institutes, universities and industries on nuclear energy. The code was developed mainly for nuclear reactor core analysis. It implemented the sufficient capabilities from the viewpoint of the analysis; it had flexible geometrical modeling and supported evaluated nuclear data such as JENDL-3.1⁴⁾ and JENDL-3.2⁵⁾ that were the latest ones at that time. However, the capabilities for shielding calculations were limited; the input specification of particle source is limited and only a few tallying functions were available. The extension of the code had been thus desired. In addition, evaluated nuclear data such as ENDF/B-VI⁶⁾ adopted the double differential cross sections (File 6) to improve the accuracy of energy-angle correlation in the scattering model.

In order to meet these requirements, the following improvement and addition of new functions have been carried out for both MVP and GMVP codes:

- capability to treat the scattering model expressed with File 6 (MF=6) of the ENDF-6 format⁷⁾
- time-dependent tallies
- reaction rate calculation with the pointwise response function
- flexible source specification
- availability of pointwise cross section libraries at an arbitrary temperature (prepared by the MVP-ART code)
- estimation of real variances in eigenvalue problems
- point detector and surface crossing estimators

- statistical geometry model
- function of reactor noise analysis used for simulation of the Feynman- α experiment
- arbitrary shaped lattice boundary
- periodic boundary condition
- parallelization with standard libraries
- supporting many platforms.

These extended capabilities enable us to perform shielding calculations and criticality calculations for commercial reactors. The capabilities were implemented into the MVP/GMVP code Version 2⁸⁾ and the new version was released in 2005.

In order to expand the applicability of the Monte Carlo method, new methodologies have been developed after the release of version 2. In particular, reactor physicists and engineers needed the reference solutions with a Monte Carlo code for reactor physics parameters except for the effective multiplication factor, neutron flux and reaction rates. To satisfy their needs, the following Monte Carlo methodologies have been developed and implemented into the MVP code:

- perturbation calculation for effective multiplication factor
- exact resonant elastic scattering model
- calculation of reactor kinetics parameters
- photonuclear reaction model
- simulation of delayed neutrons
- generation of group constants.

The perturbation capability is a function to calculate the small reactivity worth. The reactivity worth can be calculated with the difference between the inverses of effective multiplication factors. If the difference is small, the relative statistical uncertainty becomes large and thus much computation time is required. To circumvent the difficulty, the Monte Carlo perturbation techniques of the correlated sampling and the differential operator sampling have been implemented into the MVP code.

The exact resonant elastic scattering model has been developed to improve the Doppler reactivity worth for the thermal systems.⁹⁾ In the conventional scattering model used in Monte Carlo codes, the up-scattering of neutrons due to the thermal motion of a target nucleus is considered up to 4 eV and neutrons above 4 eV suffer slowing down only (asymptotic model). The exact model considers the thermal motion over the entire energy region and the energy dependence on the resonant elastic scattering probability for it.¹⁰⁾ Namely, the exact model does not include any approximations.

The calculation capability of reactor kinetics parameters provides the functions to calculate the effective delayed neutron fraction and the neutron generation time. These parameters are reaction rates weighted with the adjoint flux and it is difficult to calculate them with the continuous-energy Monte Carlo methods. Recently, a new methodology has been developed to calculate the adjoint-weighted reaction rates by interpreting the adjoint flux as the iterated fission probability. An alternative methodology with the Monte Carlo perturbation techniques has been also developed. The MVP code adopts the latter method to calculate the reactor kinetics parameters.

The photonuclear reaction model and the simulation capability of delayed neutrons are the functions for active non-destructive assay by photon interrogation for uranium-bearing waste. For the details of the method, one should consult Reference 11.

The capability of the multigroup constant generation provides an alternative method to calculate the effective multigroup cross sections without the cell calculation codes such as the PIJ module in the MOSRA-SRAC code¹²⁾ or the SLAROM-UF code.¹³⁾ The flexible geometry description, which is the advantage of the Monte Carlo method, enables the user to calculate multigroup constants accurately for the complicated geometries designed for new-types of reactor assemblies. Currently, this capability cannot generate all the multigroup constants for reactor calculations. It can generate the multigroup cross sections of the vector form, the multigroup cross sections of the matrix form, the diffusion coefficient calculated by the average of scattering cosine, and the higher-order Legendre expanded scattering cross sections.

In addition to the needs above, the need is growing for large-scale Monte Carlo calculations¹⁴⁾ and some benchmark problems have been proposed.^{15, 16)} A very large number of tally regions are required to calculate the power distribution at pin-by-pin level with a detailed full-core reactor core model. Namely, a very large amount of memory is required for such calculations. The previous versions of the MVP/GMVP code could not address the calculations because of the limitation of memory up to 2 Gbytes. The current version of MVP/GMVP code can use large memory beyond 2 Gbytes. A large-scale Monte Carlo calculation for detailed reactor analysis is possible with the current version (64 bit version) of MVP.¹⁷⁾

This report is the user's manual of the MVP/GMVP code Version 3 which includes the above new functions. The input instructions, calculation output and physics models are described in detail. The capabilities added to date are also explained. In addition, the usage of the CGVIEW code which draws the cross-sectional view of the calculation geometry is also described in Appendix B.

We will continue to maintain the MVP/GMVP code to enhance the usability and the quality with user feedback.

2. Basics of Monte Carlo Methods

The particle-transport Monte Carlo method is a procedure to calculate statistically the behavior of radiation such as neutrons or photons precisely. It has a distinct characteristic that one can, in principle, perform more precise calculations than the deterministic diffusion or transport methods. Therefore, several Monte Carlo codes have been developed so far. The famous codes are MORSE,¹⁸⁾ KENO,¹⁹⁾ MCNP²⁰⁾ and VIM.²¹⁾ In this chapter, the basics of the Monte Carlo methods and the actual treatment in MVP/GMVP are described.

2.1 Particle Transport Calculation

In general, the particle behavior is described by the density $n_k(\mathbf{r}, \mathbf{v}, t)$ of particles at the spatial position $\mathbf{r} = (x, y, z)$, the velocity $\mathbf{v} = (v_x, v_y, v_z)$ and time t . k indicates the kind of particles such as neutrons or photons but it is omitted in the following description except for the case in confusion.

Often used is the angular density $n_k(\mathbf{r}, E, \mathbf{\Omega}, t)$ that has independent variables the particle energy E and the unit vector of the flight direction $\mathbf{\Omega}$ instead of the particle velocity. The degree of freedom for $\mathbf{\Omega}$ is 2.

The particle behavior is described by the following differential-form transport equation (Boltzmann equation):²²⁾

$$\frac{\partial n(\mathbf{r}, E, \mathbf{\Omega}, t)}{\partial t} + \mathbf{v} \cdot \nabla n(\mathbf{r}, E, \mathbf{\Omega}, t) + \Sigma_t(\mathbf{r}, E)vn(\mathbf{r}, E, \mathbf{\Omega}, t) = \int d\mathbf{v}' \Sigma_t(\mathbf{r}, E')f(\mathbf{r}; E', \mathbf{\Omega}' \rightarrow E, \mathbf{\Omega})vn(\mathbf{r}, E', \mathbf{\Omega}', t) + S(\mathbf{r}, E, \mathbf{\Omega}, t), \quad (2.1)$$

where Σ_t is the total cross section, $f(\mathbf{r}; E', \mathbf{\Omega}' \rightarrow E, \mathbf{\Omega})$ the transition probability for the total cross section, $v(=|\mathbf{v}|)$ the particle speed and $S(\mathbf{r}, E, \mathbf{\Omega}, t)$ the particle source. The transition probability is expressed by the probability f_x for each reaction type x as follows:

$$\Sigma_t(\mathbf{r}, E')f(\mathbf{r}; E', \mathbf{\Omega}' \rightarrow E, \mathbf{\Omega}) = \sum_x \Sigma_x(\mathbf{r}, E')f_x(\mathbf{r}; E', \mathbf{\Omega}' \rightarrow E, \mathbf{\Omega}). \quad (2.2)$$

Defining the angular flux

$$\Phi(\mathbf{r}, E, \mathbf{\Omega}, t) = vn(\mathbf{r}, E, \mathbf{\Omega}, t), \quad (2.3)$$

the actual calculation is based on the following equation:

$$\frac{1}{v} \frac{\partial \Phi(\mathbf{r}, E, \mathbf{\Omega}, t)}{\partial t} + \mathbf{\Omega} \cdot \nabla \Phi(\mathbf{r}, E, \mathbf{\Omega}, t) + \Sigma_t(\mathbf{r}, E)\Phi(\mathbf{r}, E, \mathbf{\Omega}, t) = \int dE' \int d\mathbf{\Omega}' \Sigma_t(\mathbf{r}, E')f(\mathbf{r}; E', \mathbf{\Omega}' \rightarrow E, \mathbf{\Omega})\Phi(\mathbf{r}, E', \mathbf{\Omega}', t) + S(\mathbf{r}, E, \mathbf{\Omega}, t). \quad (2.4)$$

Here only the reaction of particles with nucleus is taken into account and the interaction between the particles is ignored. This assumption is adequate for electrically neutral particles such as neutrons and photons.

In general, Eq. (2.4) is often solved by discretizing independent variables and quantities such as cross sections for space, energy, angle and time in the deterministic method. On the other hand, no discretization for space and time is required in the Monte Carlo method. Furthermore, no discretization for angle and energy is required in the continuous-energy Monte Carlo method. It is because the transport equation is solved stochastically by simulating the particle behavior in the Monte Carlo method. In order to describe the method mathematically, we derive the collision density equation. At first, we obtain the integral-form transport equation from Eq. (2.4).

$$\Phi(\mathbf{r}, E, \boldsymbol{\Omega}, t) = \int_0^\infty ds e^{-\eta(s)} Q(\mathbf{r} - s\boldsymbol{\Omega}, E, \boldsymbol{\Omega}, t - s/v) \quad (2.5)$$

$$\eta(s) = \int_0^s ds' \Sigma_t(\mathbf{r} - s'\boldsymbol{\Omega}, E) \quad (2.6)$$

$$Q(\mathbf{r}, E, \boldsymbol{\Omega}, t) = \int dE' \int d\boldsymbol{\Omega}' \Sigma_t(\mathbf{r}, E) f(\mathbf{r}; E', \boldsymbol{\Omega}' \rightarrow E, \boldsymbol{\Omega}) \Phi(\mathbf{r}, E', \boldsymbol{\Omega}', t) + S(\mathbf{r}, E, \boldsymbol{\Omega}, t). \quad (2.7)$$

$\eta(s)$ is the integral of a reaction cross section along the direction of the particle flight and is called the optical path length or the optical thickness. $Q(\mathbf{r}, E, \boldsymbol{\Omega}, t)$ is the particle source including the scattering source.

By using Eq. (2.5), the collision density $\Psi(\mathbf{r}, E, \boldsymbol{\Omega}, t)$ can be obtained as follows:

$$\begin{aligned} \Psi(\mathbf{r}, E, \boldsymbol{\Omega}, t) &= \Sigma_t(\mathbf{r}, E) \Phi(\mathbf{r}, E, \boldsymbol{\Omega}, t) \\ &= \Sigma_t(\mathbf{r}, E) \int_0^\infty ds e^{-\eta(s)} Q(\mathbf{r} - s\boldsymbol{\Omega}, E, \boldsymbol{\Omega}, t - s/v). \end{aligned} \quad (2.8)$$

We transform Eq. (2.8) into the 3-dimensional form to describe the collision density in the kernel form:

$$\begin{aligned} \Psi(\mathbf{r}, E, \boldsymbol{\Omega}, t) &= \int d\mathbf{r}' \Sigma_t(\mathbf{r}, E) \exp \left[- \int_0^{|\mathbf{r}-\mathbf{r}'|} \Sigma_t(\mathbf{r} - s'' \frac{\mathbf{r} - \mathbf{r}'}{|\mathbf{r} - \mathbf{r}'|}) ds'' \right] \\ &\quad \times \frac{\delta(\boldsymbol{\Omega} \cdot \frac{\mathbf{r} - \mathbf{r}'}{|\mathbf{r} - \mathbf{r}'|} - 1)}{|\mathbf{r} - \mathbf{r}'|^2} Q(\mathbf{r}', E, \boldsymbol{\Omega}, t') \\ &= \int d\mathbf{r}' \Sigma_t(\mathbf{r}, E) e^{-\eta(|\mathbf{r}-\mathbf{r}'|)} \frac{\delta(\boldsymbol{\Omega} \cdot \frac{\mathbf{r} - \mathbf{r}'}{|\mathbf{r} - \mathbf{r}'|} - 1)}{|\mathbf{r} - \mathbf{r}'|^2} Q(\mathbf{r}', E, \boldsymbol{\Omega}, t'), \end{aligned} \quad (2.9)$$

where

$$\mathbf{r}' = \mathbf{r} - s\boldsymbol{\Omega} \quad (2.10)$$

$$t' = t - \frac{s}{v}, \quad (2.11)$$

and $\delta(x)$ is the Dirac delta function.

We define the transport kernel T and the collision kernel C as follows:

$$T(\mathbf{r}, E, \boldsymbol{\Omega}; \mathbf{r}') = \Sigma_t(\mathbf{r}, E) e^{-\eta(|\mathbf{r}-\mathbf{r}'|)} \frac{\delta(\boldsymbol{\Omega} \cdot \frac{\mathbf{r} - \mathbf{r}'}{|\mathbf{r} - \mathbf{r}'|} - 1)}{|\mathbf{r} - \mathbf{r}'|^2} \quad (2.12)$$

$$C(\mathbf{r}, E, \boldsymbol{\Omega}; E', \boldsymbol{\Omega}') = \sum_x C_x(\mathbf{r}, E, \boldsymbol{\Omega}; E', \boldsymbol{\Omega}') \quad (2.13)$$

$$C_x(\mathbf{r}, E, \boldsymbol{\Omega}; E', \boldsymbol{\Omega}') = \frac{\Sigma_x(\mathbf{r}, E') f_x(\mathbf{r}; E', \boldsymbol{\Omega}' \rightarrow E, \boldsymbol{\Omega})}{\Sigma_t(\mathbf{r}, E')}. \quad (2.14)$$

Furthermore, we define the transition kernel:

$$K(\mathbf{r}, E, \boldsymbol{\Omega}; \mathbf{r}', E', \boldsymbol{\Omega}') = T(\mathbf{r}, E, \boldsymbol{\Omega}; \mathbf{r}') C(\mathbf{r}', E, \boldsymbol{\Omega}; E', \boldsymbol{\Omega}'). \quad (2.15)$$

Then Eq. (2.8) can be expressed as follows:

$$\begin{aligned} \Psi(\mathbf{r}, E, \boldsymbol{\Omega}, t) &= \hat{S}(\mathbf{r}, E, \boldsymbol{\Omega}, t) \\ &+ \int d\mathbf{r}' \int d\boldsymbol{\Omega}' \int_0^\infty dE' K(\mathbf{r}, E, \boldsymbol{\Omega}; \mathbf{r}', E', \boldsymbol{\Omega}') \Psi(\mathbf{r}', E', \boldsymbol{\Omega}', t'), \end{aligned} \quad (2.16)$$

where $\hat{S}(\mathbf{r}, E, \boldsymbol{\Omega}, t)$ is the first-collision source (or the first-flight collision density) at the phase space $(\mathbf{r}, E, \boldsymbol{\Omega}, t)$. It is defined in the following equation:

$$\hat{S}(\mathbf{r}, E, \boldsymbol{\Omega}, t) = \int d\mathbf{r}' T(\mathbf{r}, E, \boldsymbol{\Omega}; \mathbf{r}') S(\mathbf{r}', E, \boldsymbol{\Omega}, t). \quad (2.17)$$

For simplicity, we define the phase space with $\Gamma = (\mathbf{r}, E, \boldsymbol{\Omega}, t)$. Then Eq. (2.16) turns out to be

$$\Psi(\Gamma) = \hat{S}(\Gamma) + \int d\Gamma' K(\Gamma; \Gamma') \Psi(\Gamma'). \quad (2.18)$$

Equation (2.18) describes the collision density and the solution is expressed by the Neumann series:

$$\Psi(\Gamma) = \sum_{m=0}^{\infty} \int d\Gamma' K_m(\Gamma; \Gamma') \hat{S}(\Gamma'), \quad (2.19)$$

where

$$\begin{aligned} K_0(\Gamma; \Gamma') &= \delta(\Gamma - \Gamma'), \\ K_1(\Gamma; \Gamma') &= K(\Gamma; \Gamma'), \\ K_2(\Gamma; \Gamma') &= \int d\Gamma_1 K(\Gamma; \Gamma_1) K(\Gamma_1; \Gamma'), \\ &\vdots \\ K_i(\Gamma; \Gamma') &= \int d\Gamma_1 \cdots \int d\Gamma_{i-1} K(\Gamma; \Gamma_{i-1}) K(\Gamma_{i-1}; \Gamma_{i-2}) \cdots K(\Gamma_1; \Gamma'). \end{aligned} \quad (2.20)$$

From Eqs. (2.18) and (2.20), we find that the collision density is the sum of the contribution from particles colliding at Γ for the first time and after one or more collisions. The Monte Carlo method is based on Eq. (2.18). Normalizing the source to unity, Eq. (2.18) is the probability density for the number of collisions at Γ .

2.2 Particle Transport with Monte Carlo Methods

In the previous section, we stated the transport equation to describe the particle behavior. To be precise, the Monte Carlo method solves the equation but it is easy to understand the method from the viewpoint of the simulation of particles in space.

As shown in Fig. 2.1, the “pseudo”-particles walk around in the three-dimensional virtual space of a computer so that they obey the physical laws between particles and nuclei. We can calculate physical quantities by scoring events of the particles. The particle behavior is determined with random numbers* so that it satisfies physical laws statistically. This is called random walk. The events in the random walk are as follows.

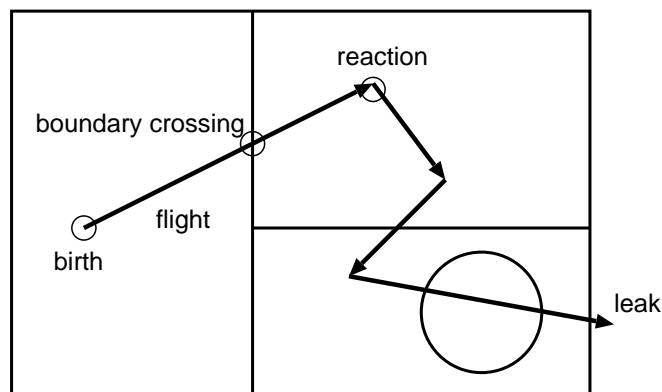


Fig. 2.1 Random walk of a particle

● **Generation of particles** Particles are generated with the probability density for the specified space, energy, time and the kinds of particles and then start random walk.

● **Flight of particles** The energy of particles does not change in the flight between collisions with a nucleus in space. The distance from a collision to the next collision is sampled statistically with reaction cross sections of the material where the particles pass. The probability density function of the distance is the transport kernel expressed by Eq. (2.12).

When the total cross section is constant in a flight pass, the probability density function $p(x)$ for the distance to the collision x is

$$p(x)dx = \Sigma_t e^{-\Sigma_t x} dx. \quad (2.21)$$

In the MVP/GMVP code, the distance to the collision is sampled from Eq. (2.21) because the code treats only the system where the cross sections of each zone are constant. Since the system is, in general, divided into zones whose materials are different from each other, the particles may pass through multiple zones until the next collision. In this case, the distance is sampled more than once.

Usually the calculation geometry is finite. If a particle is out of the geometry, the random walk of the particle is terminated.

Since the distance from a particle position to a zone boundary is calculated many times in the code, the calculation of the distance often dominates a large part of the total calculation time.

*Uniform random numbers are always used in particle-transport Monte Carlo calculations. Uniform random numbers are simply called random numbers in this document.

● **Zone search** In general, a particle passes through multiple zones. In this case, it is required to determine the zone the particle enters next when the particle is out of a zone. The next zone can be determined simply from the arrangement in space in some cases but there are many cases where it cannot be. The procedure to improve the efficiency of the zone search is used. For example, the zone the particle may enter from a zone is recorded (learned) in the program. Therefore, the processing time for the zone search becomes larger with the increasing number of zones the particle may enter. Users should consider decreasing the number of zones adjacent to a zone.

● **Reaction with nuclei** When a particle collides with a nucleus, the energy and the flight direction of the particle change with the probability distribution for some channels of the reaction cross sections. In the absorption reaction, the random walk of the particle may be terminated.

2.3 Spatial Description

In the conventional numerical method such as the finite difference, finite element and boundary element methods and so on, it is common that the geometrical space is divided into a lot of meshes and the curved shape is approximated with polygons. On the other hand, it is not necessary that the space is divided into meshes, and the spatial geometry can be described by the combination of curved surfaces such as cylinder, sphere etc. in the Monte Carlo method. This section describes the concept of the combinatorial geometry (CG) employed in MVP/GMVP. The details are given in Chapter 5.

2.3.1 Spatial Description by Combinatorial Geometry

In order to describe the geometry the MVP/GMVP code has the basic shapes such as cylinder, rectangular parallelepiped, ellipsoid, torus etc. In MVP/GMVP, the shapes are called “BODies”.

A spatial region (“ZONE” that is a uniform region filled with a single kind of material) is defined by the combination of BODies as shown in Fig. 2.2. Since the definition of ZONES provides the information for particles to move around, it is unnecessary that the ZONES should be divided into meshes. Of course, there are some cases where the ZONES are intentionally divided into sub-ZONES to calculate the distribution of physical quantities in a spatial region. Since such a division is meaningless for particles, there are no influences on the calculated values (for example, total reaction rate, neutron multiplication factor etc.) except for statistical fluctuation. The users should divide a uniform ZONE finely only for a special purpose. This is a large difference from other methods where the division of meshes essentially affects the calculation accuracy.

2.3.2 Repeated Geometry (LATTICE Geometry)

In particle transport problems, there are many cases where the same geometrical shape is arranged repeatedly like fuel assemblies or pins in a reactor. If all the ZONES for the repeated geometry were defined with the combination of BODies, the amount of input data would be huge. Therefore the method to model such a repeated geometry easily has been developed. In the method, a geometrical shape is defined for the unit of the repeated geometry and then the geometrical shapes are arranged in certain regions repeatedly as shown in Fig. 2.3.

In MVP/GMVP, the unit of a repeated geometry is called a “LATTICE CELL” or simply “CELL”. A “LATTICE geometry” is comprised of CELLS and a LATTICE FRAME which defines the outer limit of the CELL arrangement. The LATTICE geometry can be placed at an arbitrary position along an arbitrary direction and a part of a spatial region.

MVP/GMVP has the following LATTICE geometries.

- Rectangular parallelepiped LATTICE

The LATTICE CELLS of a rectangular parallelepiped shape are arranged without the gap between the CELLS.

- Hexagonal prism LATTICE

The LATTICE CELLS of a hexagonal prism shape are arranged without the gap between the CELLS.

- Statistical geometry model

The LATTICE CELLS are not regularly arranged in a LATTICE for this model. This is the special LATTICE geometry where specified spherical CELLS are distributed statistically in the LATTICE with a constant packing fraction. In MVP/GMVP, the model must be input as a type of the LATTICE geometries.

Any shape of the LATTICE FRAME is acceptable when the FREE-LATTICE-FRAME option is

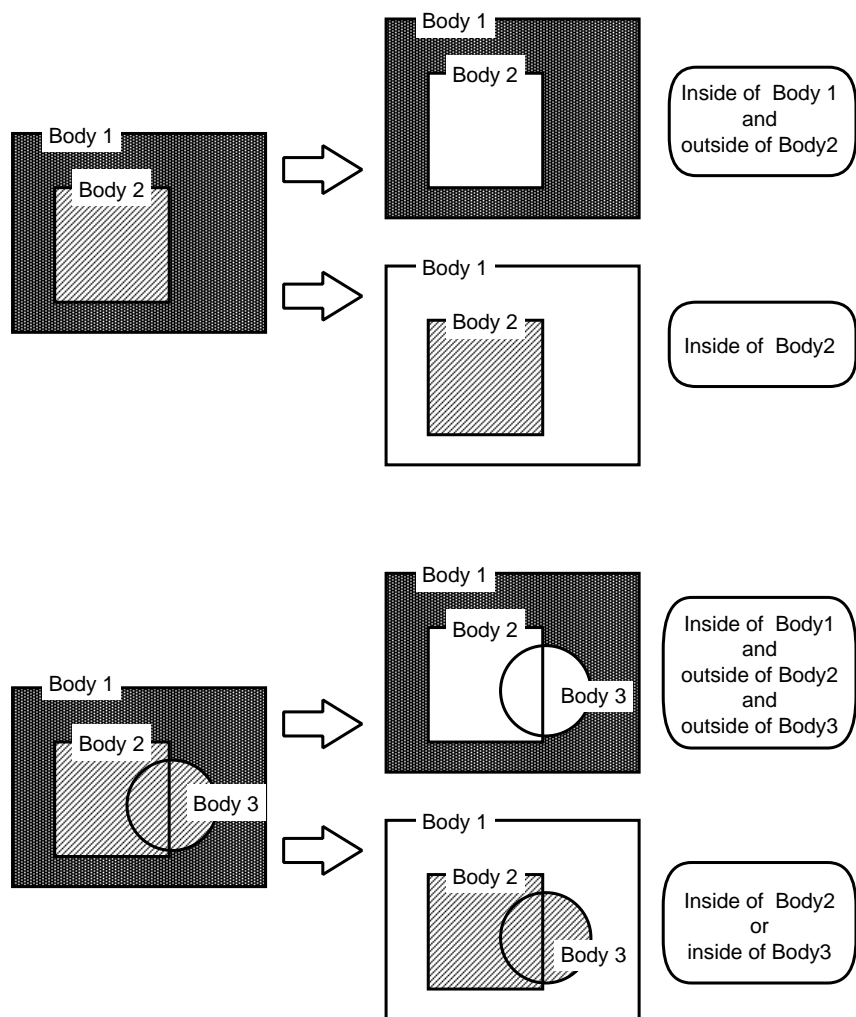


Fig. 2.2 Definition of ZONES by CG

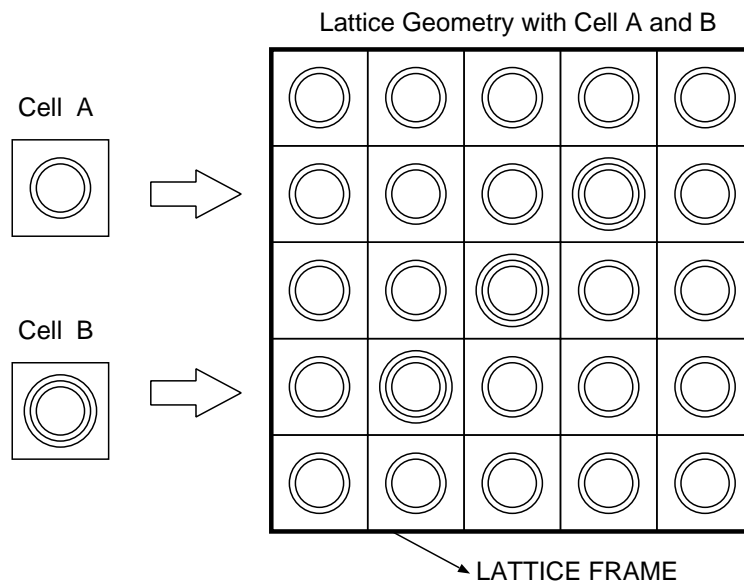


Fig. 2.3 Description of a repeated geometry with a LATTICE geometry

specified in the input data. This option is automatically effective for the statistical geometry model. Without the option, the following restrictions are imposed for the LATTICE FRAME.

- Rectangular parallelepiped LATTICE

The LATTICE FRAME must be the outer boundary of CELL arrangement. No CELLS must be cut off by the LATTICE FRAME.

- Hexagonal prism LATTICE

The LATTICE FRAME cuts off the CELL arrangement in the cross-sectional view and only the inside of the FRAME is defined as the LATTICE geometry. There are some CELLS a part of which is cut off by the LATTICE FRAME but MVP/GMVP can treat them properly without the FREE-LATTICE-FRAME option. But the CELLS must have the same height and fit to the LATTICE FRAME in the vertical direction.

Following 3 shapes of the LATTICE FRAME can be used for the hexagonal prism LATTICE. (See Fig. 2.4.)

- (1) Hexagonal prism FRAME
- (2) Cylindrical FRAME
- (3) Rectangular parallelepiped FRAME

2.4 Boundary Conditions

In general, boundary conditions are described with “fictitious materials” in the particle-transport Monte Carlo codes using CG. The following materials are commonly used.

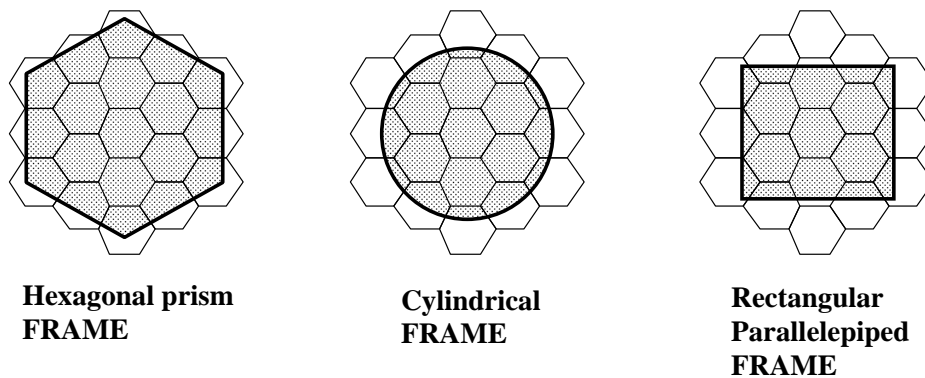


Fig. 2.4 Hexagonal prism LATTICE without the FREE-LATTICE-FRAME option

(1) Perfect reflective material (mirror reflective material)

A particle which enters the perfect reflective material is reflected on the surface of the material as light is reflected with a mirror. This corresponds to the perfect reflective boundary condition.

(2) Isotropic reflective material (white reflective material)

A particle which enters the isotropic reflective material is isotropically reflected on the surface of the material. The reflection probability is the same for all the reflected directions. This corresponds to the isotropic reflective (white) boundary condition.

(3) Material for vacuum boundaries

The random walk of a particle which enters the material for vacuum boundaries is terminated. This corresponds to the vacuum boundary condition.

(4) Material for periodic boundaries

A fictitious material is used to represent the periodic boundary condition. Currently, the periodic boundary condition can be applied only for BODies that have a pair of parallel planes. The user cannot specify the 60-degree rotational boundary condition etc.

All the materials can be used in MVP/GMVP. In principle, these materials can be placed at any spatial region except for the material for periodic boundaries. However, the user must consider whether the calculation model with a fictitious material is adequate or not.

2.5 Treatment of Nuclear Reactions

In the treatment of electrically neutral particles such as neutrons and photons, the interaction between particles is ignored and only the reactions with nuclei or atoms are considered. The particle behavior in the reaction with nuclei or atoms is determined on the basis of the information of evaluated nuclear data files. There are two methods for the treatment of energy:

- Continuous-energy method
- Multigroup method

In the case of the continuous-energy method, the Monte Carlo codes use the continuous-energy form of cross sections processed from the evaluated nuclear data files. On the other hand, the multigroup cross sections averaged in some energy ranges (energy groups) are used in the multigroup method. The continuous-energy method has the feature of high accuracy but usually requires more calculation time than the multigroup method.

- Codes with the continuous-energy method: MVP, MCNP, VIM etc.
- Codes with the multigroup method: GMVP, MORSE, KENO etc.

In the continuous-energy method, the reaction of a particle with a nucleus is determined as follows. The total cross section of the nucleus for neutrons is expressed as the sum of the partial cross sections for possible reactions:

$$\sigma_t = \sigma_a + \sigma_{el} + \sigma_{inel} + \dots, \quad (2.22)$$

$$\sigma_a = \sigma_\gamma + \sigma_f + \dots, \quad (2.23)$$

where σ_a , σ_{el} and σ_{inel} are the absorption (capture σ_γ + fission σ_f + ...), elastic scattering and inelastic scattering cross sections, respectively. The total cross section for photons is expressed as the sum of the partial cross sections:

$$\sigma_t = \sigma_a + \sigma_{coh} + \sigma_{inc} + \sigma_{pp}, \quad (2.24)$$

$$\sigma_a = \sigma_{pe}, \quad (2.25)$$

where σ_a , σ_{coh} , σ_{inc} and σ_{pp} are the absorption (photo-electric effect σ_{pe}), coherent scattering, incoherent scattering and electron pair production cross sections, respectively. Then, the probability p_x for reaction x is

$$p_x = \frac{\sigma_x}{\sigma_t}. \quad (2.26)$$

The reaction is determined from the probability for each reaction with random numbers and then the particle is processed according to the selected reaction.

The processing of scattering reactions usually determines the energy and scattering direction after the reaction. For the absorption reaction, following two treatments are available.

- **Analog absorption**

When the absorption reaction is selected, the random walk of the particle is terminated.

- **Non-analog absorption**

A “part” of the particle is assumed to be absorbed and the reaction except for absorption is selected. In the non-analog simulation like the non-analog absorption, the “weight” of the particle is introduced. This is the parameter to give the particle the substantial number of particles. When a particle is generated from a source, the weight is usually set to unity. After a “part” of the particle is absorbed, the weight is multiplied by $(\sigma_t - \sigma_a)/\sigma_t$. The probability of reaction x except for absorption is $p_x = \sigma_x/(\sigma_t - \sigma_a)$.

The MVP code can treat the analog and the non-analog absorption depending on the energy of the particle. The GMVP code can use only the non-analog absorption.

2.6 Estimation of Physical Quantities

Since no information can be obtained from only the random walk of particles, a procedure to obtain physical quantities is required. The procedure is performed with estimators in the Monte Carlo method. In this section, the estimators available in MVP/GMVP are described. In MVP/GMVP, the statistical estimation of physical quantities from the random walk is called “tallying”.

2.6.1 Collision Estimator

In general, the physical quantities estimated with the Monte Carlo method can be expressed with the collision density Ψ as follows:

$$I = \int_{\Gamma \in V_T} d\Gamma h(\Gamma) \Psi(\Gamma), \quad (2.27)$$

where $\Gamma = (\mathbf{r}, E, \boldsymbol{\Omega}, t)$ is the set of coordinates (position, energy, angle, time) in the phase space and V_T is the region of interest in the phase space. $h(\Gamma)$ is the function to represent the contribution from the collision at Γ to the reaction of interest. For example, the macroscopic capture rate can be obtained by using the following function:

$$h(\Gamma) = \frac{\Sigma_c(\Gamma)}{\Sigma_t(\Gamma)}. \quad (2.28)$$

The particle flux can be also obtained by

$$h(\Gamma) = \frac{1}{\Sigma_t(\Gamma)}. \quad (2.29)$$

In general, a microscopic reaction rate or a response can be obtained by

$$h(\Gamma) = \frac{\sigma(\Gamma)}{\Sigma_t(\Gamma)}, \quad (2.30)$$

where $\sigma(\Gamma)$ is a reaction cross section or a response function,

A particle usually repeats collisions with nuclei in the process of the random walk. The physical quantity I can be estimated by scoring the contribution in the collisions. This scoring method is called the collision estimator.

In general, the physical quantity can be estimated with the collision estimator as follows:

$$C_i = h(\Gamma_i) W_i \delta_{V_T}(\Gamma_i), \quad (2.31)$$

where C_i is the contribution of the i -th event scored, Γ_i the i -th collision point of a history, W_i the weight of the particle at the collision and $\delta_{V_T}(\Gamma_i)$ is defined as follows:

$$\begin{aligned} \delta_{V_T}(\Gamma_i) &= 1 \text{ for } \Gamma_i \in V_T \\ &= 0 \text{ for } \Gamma_i \notin V_T. \end{aligned}$$

2.6.2 Track Length Estimator

In a vacuum region, physical quantities cannot be estimated with the collision estimator. In addition, the estimation with the collision estimator is less accurate in the region where the number of collisions is small. The track length estimator overcomes the drawbacks for the collision estimator. The physical

quantities can be estimated from the track of particles in a region. The contribution of the i -th event scored is estimated as follows:

$$C_i = \sigma(L_i)W_iL_i\delta_{V_T}(\Gamma_i), \quad (2.32)$$

where L_i is the track (path) length of the particle passing through the region V_T and W_i is the weight of particle in flight. The expression $\sigma(L_i)$ means that the reaction cross sections are assumed to be constant in space and time along the path L_i . (“ZONES” in MVP/GMVP obey this assumption.) The track length estimator is often used because the physical quantities can be estimated in vacuum regions.

This estimator can be derived from the collision estimator by introducing the infinite fictitious collision cross sections.²³⁾

2.6.3 Point Detector Estimator

The collision and track length estimators are available for the estimation of the integral or averaged value in a finite region. However, they cannot be applied for the estimation of physical quantities at a point because the possibility that particles reach the point in the random walk is extremely low.

The physical quantities at a point are estimated from the contribution assuming that the particles fly to the point from the collision points and sources. Letting the detector point \mathbf{r}_d , the collision density at \mathbf{r}_d is derived from Eq. (2.9):

$$\Psi(\mathbf{r}_d, E, \boldsymbol{\Omega}, t) = \int d\mathbf{r}' \Sigma_t(\mathbf{r}_d, E) e^{-\eta(|\mathbf{r}_d - \mathbf{r}'|)} \frac{\delta(\boldsymbol{\Omega} \cdot \frac{\mathbf{r}_d - \mathbf{r}'}{|\mathbf{r}_d - \mathbf{r}'|} - 1)}{|\mathbf{r}_d - \mathbf{r}'|^2} Q(\mathbf{r}', E, \boldsymbol{\Omega}, t'). \quad (2.33)$$

The angular fluxes are estimated on the basis of this equation but the point detector estimator often estimates the angle-integrated fluxes. Integrating the equation over the angle $\boldsymbol{\Omega}$,

$$\begin{aligned} \psi(\mathbf{r}_d, E, t) &= \frac{1}{\Sigma_t(\mathbf{r}_d, E)} \int d\boldsymbol{\Omega} \Psi(\mathbf{r}_d, E, \boldsymbol{\Omega}, t) \\ &= \int d\mathbf{r}' \frac{e^{-\eta(|\mathbf{r}_d - \mathbf{r}'|)}}{|\mathbf{r}_d - \mathbf{r}'|^2} Q(\mathbf{r}', E, \frac{\mathbf{r}_d - \mathbf{r}'}{|\mathbf{r}_d - \mathbf{r}'|}, t'). \end{aligned} \quad (2.34)$$

Since Q includes the external and scattering sources as shown in Eq. (2.7), the flux at \mathbf{r}_d can be obtained from the contribution of the particles which reach directly from the external source and are scattered with nuclei.

From Eq. (2.34), the physical quantity is actually estimated as follows:

$$\begin{aligned} \psi(\mathbf{r}_d, E, t) &= \sum_i W_i f(\mathbf{r}_i; E, \boldsymbol{\Omega}_i) \frac{e^{-\eta(|\mathbf{r}_d - \mathbf{r}_i|)}}{|\mathbf{r}_d - \mathbf{r}_i|^2} \\ &\quad + \sum_j W_j f(\mathbf{r}_j; E', \boldsymbol{\Omega}' \rightarrow E, \boldsymbol{\Omega}_j) \frac{\Sigma_s(\mathbf{r}_j, E')}{\Sigma_t(\mathbf{r}_j, E')} \frac{e^{-\eta(|\mathbf{r}_d - \mathbf{r}_j|)}}{|\mathbf{r}_d - \mathbf{r}_j|^2}, \end{aligned} \quad (2.35)$$

$$\boldsymbol{\Omega}_i = \frac{\mathbf{r}_d - \mathbf{r}_i}{|\mathbf{r}_d - \mathbf{r}_i|}, \quad (2.36)$$

where W_i is the weight of the source particle, W_j the particle weight before the j -th collision, $f(\mathbf{r}_i; E, \boldsymbol{\Omega}_i)$ the probability that the source particle at \mathbf{r}_i with energy E flies to $\boldsymbol{\Omega}_i$, and $f(\mathbf{r}_j; E', \boldsymbol{\Omega}' \rightarrow E, \boldsymbol{\Omega}_j)$ the probability that the particle at \mathbf{r}_j with energy E' and the flight direction $\boldsymbol{\Omega}'$ is scattered into energy E and the flight direction $\boldsymbol{\Omega}_j$.

The point detector estimator requires the distance between the collision or source point and the detector point and the line integral $\eta(r)$ along the path. They are obtained by tracking a pseudo-particle

from a source or collision point to a detector point. Therefore, calculation time increases rapidly with the number of detectors because the pseudo-particle must be tracked for every collision.

Furthermore, special treatment must be performed for contributions from a source or collision point in the vicinity of a detector because the singularity of $1/|\mathbf{r}_d - \mathbf{r}_i|^2$ makes the variance of the point detector estimator unbounded. In addition, the probability density function of the sample mean for the estimator does not obey the central limit theorem and thus the convergence rate is slower than $1/\sqrt{N}$, where N is the total number of histories.²⁴⁾ To overcome the difficulty, many techniques have been proposed. MVP/GMVP adopts the bounded sphere approach.²⁵⁾ In this approach, the factor $\exp(-\Sigma_t r)/4\pi c^2$ is scored for each source or collision point in the bounded sphere instead of $\exp(-\Sigma_t r)/4\pi r^2$. The constant c is determined so that the contribution to the detector from source or collisions in the bounded sphere is conserved by assuming the constant flux distribution and isotropic scattering in the sphere.

The estimators that use pseudo-particles like the point detector estimator are generally called the next event estimators.

2.6.4 Surface Crossing Estimator

The surface crossing estimator estimates physical quantities when particles pass through a surface in the space. The estimator is based on the particle current rather than the particle flux.

The particle flux is defined by the number of particles that pass through a surface perpendicular to the particle flight direction per unit area. On the other hand, the particle current is defined by the number of particles that pass through a surface per unit area. The difference is in the treatment of the flight direction for each particle.

The relation between the particle current $J(\mathbf{r}, E, \boldsymbol{\Omega})$ and flux $\Phi(\mathbf{r}, E, \boldsymbol{\Omega})$ at a position \mathbf{r} along a direction $\boldsymbol{\Omega}$ on a surface is

$$J(\mathbf{r}, E, \boldsymbol{\Omega}) = |\cos \theta| \Phi(\mathbf{r}, E, \boldsymbol{\Omega}), \quad (2.37)$$

where θ is the angle between the flight direction of the particle and the normal direction of the surface.

With the surface crossing estimator, the current can be estimated by just scoring the weight of particles that pass through a detector surface. Then the flux can be calculated from the above equation as follows:

$$\Phi(\mathbf{r}, E, \boldsymbol{\Omega}) = \frac{J(\mathbf{r}, E, \boldsymbol{\Omega})}{|\cos \theta|}. \quad (2.38)$$

As shown in this equation, $\cos \theta$ of the denominator in the right hand side may be a very small value when a particle passes through the surface with the incident angle of nearly zero degree ($\theta \approx 90$ degree). It causes a large statistical uncertainty. To circumvent this problem, the absolute value of $\cos \theta$ is replaced with 0.005 when the value is less than 0.01 in MVP/GMVP.

2.7 Statistical Uncertainty

The physical quantities estimated from the random walk always involve the statistical uncertainties or fluctuation. Thus the reliability of the parameters must be estimated simultaneously. To be precise, the calculated values with the Monte Carlo method are the estimates of the mean (expected) values for random variables. When we define the emergent probability for a random variable X as $p(X)$, the averaged value of the variable $\mu(X)$ is given by

$$\mu(X) = \int X p(X) dX. \quad (2.39)$$

The averaged value is obtained from the samples

$$X_1, X_2, \dots, X_N,$$

generated independently with the probability $p(X)$. For these samples, the following value is the estimate of $\mu(X)$:

$$\mu_N = \frac{1}{N} \sum_{i=1, N} X_i. \quad (2.40)$$

This estimate converges to the real expected value as $N \rightarrow \infty$. The estimate with this feature is called the “unbiased estimator”.

The uncertainty in the Monte Carlo calculation must be the value that indicates the deviation of the estimate from the real expected value. In this sense, the variance of the random variable $V(X)$ is used for the estimation of the uncertainty:

$$V(X) = \int (X - \mu(X))^2 p(X) dX. \quad (2.41)$$

The estimate of $V(X)$ is

$$V_N = \frac{1}{N-1} \sum_{i=1, N} (X_i - \mu_N)^2. \quad (2.42)$$

The denominator is $N-1$, not N because the mean value of the estimate must be the real variance. Namely the estimate is unbiased.

The estimate of the variance is that of the population variance for X and cannot be used for the uncertainty of μ_N . The estimate can be connected with the uncertainty of μ_N by the following “central limit theorem”.

For a set of random variables that are sampled independently from the random variable X with the mean μ , the variance σ^2 (probability density function $f(X)$):

$$X_1, X_2, \dots, X_N,$$

the distribution for

$$\mu_N = \frac{1}{N} \sum_{i=1, N} X_i$$

approaches the normal distribution with the mean μ , the variance σ^2/N as $N \rightarrow \infty$.

In this sense, the square root of the variance (standard deviation) for calculated μ_N can be interpreted as the uncertainty. Since the value of σ^2 cannot be known a priori, the value is also estimated from the sampled value as V_N .

Since the variance of μ_N approaches σ^2/N , it is considered that the uncertainty of the mean value (standard deviation) for the sampled random variables decreases in inverse proportion to the square root of the number of samples. We can get more accurate estimate of the mean value as the number of samples N becomes larger. But four times the number of samples is required to obtain results with twice the statistical accuracy. The dependent samples may improve the accuracy but it is very difficult to prove that the estimate is statistically correct. Therefore, it is common that the samples are regarded as independent.

2.8 History and Batch

Let us consider what is a “sample” or “the number of samples” in the statistical process of the Monte Carlo codes like MVP/GMVP. The estimation of physical quantities in the Monte Carlo codes is performed when the specified event occurs within a certain range of space, time, energy etc. during the random walk. It seems that a value in each event – for example, the weight at collisions for the collision estimator – is regarded as a sample and the number of events as the number of samples. But this is wrong. If there are more than one event where a particle contributes to a physical quantity in the random walk, the contributions are not independent statistically. The sample with statistical independence is the sum of the contributions to the physical quantity during the random walk of a source particle (history). As illustrated in Fig. 2.5, the mean value and variance are therefore calculated by the following equations:

Mean value:

$$\bar{I} = \frac{1}{N_h} \sum_{k=1}^{N_h} \sum_{i \in E_k} C_i,$$

Variance:

$$\bar{V} = \frac{1}{N_h - 1} \sum_{k=1}^{N_h} \left(\sum_{i \in E_k} C_i - \bar{I} \right)^2,$$

where N_h is the total number of histories (path of the random walk from a source), E_k the set of events during the k -th history and C_i the contribution of the i -th event scored.

The concept of the number of “histories” is introduced, not of the number of “particles” because the path of a particle may branch during the random walk. Since there is a statistical correlation between particles branching from a particle, the sum of the contributions from the particles must be treated as one sample.

The “history” is usually the unit of samples in the Monte Carlo codes but a set of the histories (“batch”) is also the unit of samples in some cases (See Fig. 2.6). The batch is used for the statistical estimation of the eigenvalue and corresponds to the generation of fission neutrons in eigenvalue problems. In the case of MVP/GMVP, the concept of the batch is also necessary for vector processing. Simultaneous processing of multiple particles (strictly speaking, histories) enables MVP/GMVP to perform the vector processing. Therefore the statistical data for each history must be stored separately and the huge amount of memory storage is required for the estimation of physical quantities if each history is a sample. Thus the data for each batch is stored to save the memory storage.

2.9 Variance Reduction Techniques

The variance reduction technique is the one to reduce the variance, that is, improve the statistical accuracy of Monte Carlo calculations. The meaning of the variance reduction here is to improve the accuracy for results within a restricted computation time and is not always the same as to increase the number of particles for random walk within the time.

There are usually the following problems that worsen the statistical accuracy[†]:

- Particles collide and fly frequently in less important regions.

[†] There are some cases where the computation time is large because of an inefficient geometry model but the efficiency cannot be improved with the variance reduction technique.

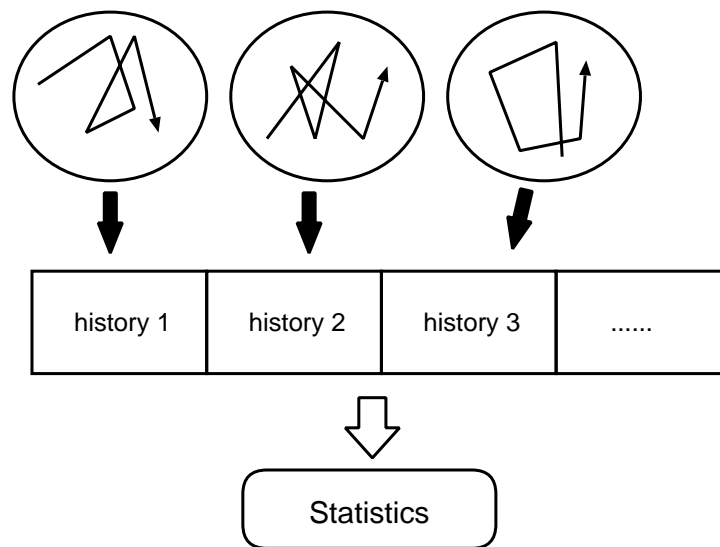


Fig. 2.5 “History” as the smallest unit of independent samples

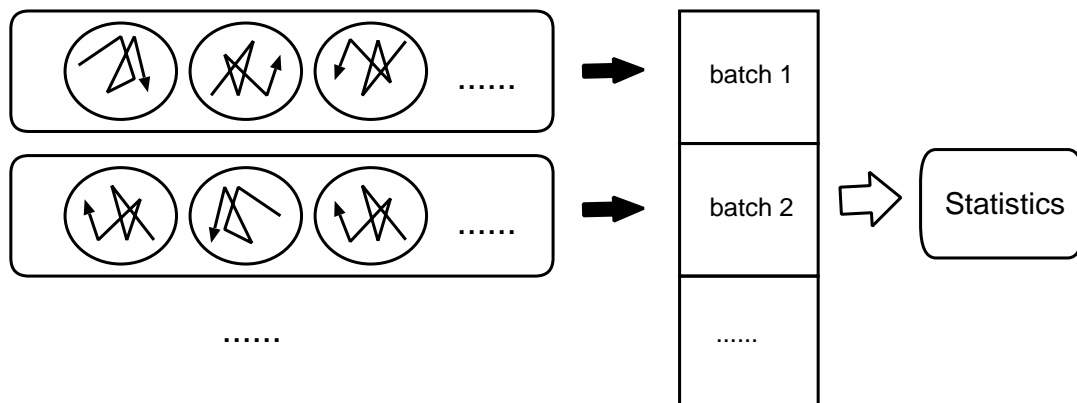


Fig. 2.6 “Batch” as the unit of independent samples

- Particles seldom reach the regions where accurate results are required.

The situation often occurs where particles seldom reach the region of interest but frequently walk around in the unimportant regions for the calculation. For example, neutrons do not reach the fuel or detector regions very much because they collide in the moderator region many times in the core calculation. In the other case for a fixed-source problem, source particles often fly to the region of less interest.

Therefore, to improve the calculation efficiency, the following situation must be made:

- The number of flights and collisions is suppressed in less important regions.
- The number of events increases in the region of interest.

However, we cannot simply terminate the random walk of particles or increase the number of particles because the obtained results may be statistically biased. Therefore, the weight of particles is

introduced as described in Section 2.5. The weight of particles is usually set to unity at the beginning of the random walk and used to express the number of particles as a real number. For example, the weight is reduced by the absorption probability at a collision and doubled for the (n,2n) reaction. In the statistical process, the contribution is halved and doubled for a particle with the weight of 0.5 and 2, respectively.

The variance reduction technique utilizes the weight to obtain the statistically correct random walk of particles.

- (1) When the number of particles is increased, the weight of each particle is decreased so that the total weight of the particles is conserved.
- (2) When the number of particles is decreased, the weight of terminated particles is added to that of survival particles.

The former is realized by splitting a particle entering a specified spatial/energy region into more than one so that the sum of the weight of split particles is conserved and resuming the random walk of each particle separately. This variance reduction technique is called “splitting”.

On the other hand, the latter cannot be easily implemented into the codes because particles move around independently in principle and thus the terminated and survival particles do not always exist in the region of interest at the same time. (MVP/GMVP processes multiple particles simultaneously but the situation is the same.) In the Monte Carlo codes, it is common that the particles entering unimportant regions are killed with a certain probability and the weight of the survived particles is increased. This variance reduction technique is called “Russian roulette”.

2.9.1 Russian Roulette

The expected value of the particle weight after and before the application of Russian roulette is conserved. Figure 2.7 illustrates the Russian roulette technique. Suppose that the weight before the application is W_0 , the following operation is performed:

- (A) The particle is “survived” with the probability p . Conversely, the particle is “killed” with $1 - p$.
- (B) The weight of the survived particle is set to W_0/p . The random walk of the killed particle is terminated.

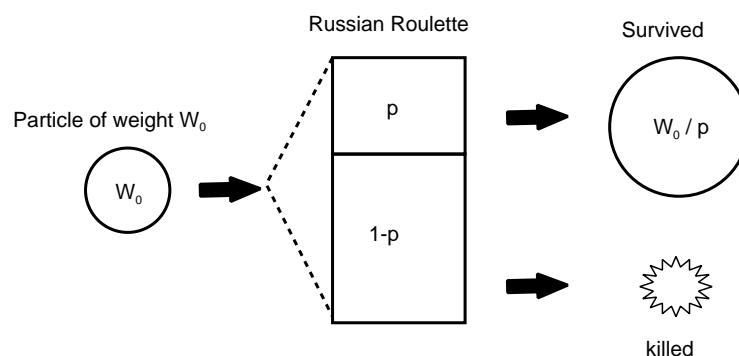


Fig. 2.7 Russian roulette

Since the weight of killed particles is regarded as 0, the expected value after Russian roulette is calculated as

$$p \cdot \frac{W_0}{p} + (1 - p) \cdot 0 = W_0.$$

Therefore the expected value of the weight is the same as that before Russian roulette.

In MVP/GMVP, there are two ways to determine the probability p .

- $p = W_0/W_s$ for a specified value W_s larger than W_0 .
- When a particle enters a spatial/energy region of importance I_1 from the other region of importance I_0 , Russian roulette is performed with the probability $p = I_1/I_0$ if $I_0 > I_1$.

The former method is called Russian roulette with weight and the latter Russian roulette with importance. The input data WSRV and XIMP for MVP/GMVP correspond to W_s and I_0 & I_1 , respectively. The data depend on the region and energy. Russian roulette with weight is actually performed for the particles whose weight is not more than W_k . W_k is given by the input parameter WKIL and depends on the region and energy.

Russian roulette is an effective variance reduction technique but it should not be performed “extremely”. Suppose that the survival probability for less important regions is set to a very small value. Then the random walk of particles might be statistically meaningless because the particles seldom pass through the region.

2.9.2 Splitting

There are also two ways for splitting in MVP/GMVP.

- If the weight W_0 of a particle is more than $2W_s$, the particle is split into W_0/W_s particles with the weight W_s . W_s is given by the input parameter WSRV.
- When a particle enters a spatial/energy region of importance I_1 from the other region of importance I_0 , the particle is split into I_1/I_0 particles if $I_0 < I_1$. The weight of the particles is $W_0 \cdot (I_0/I_1)$.

The number of the split particles W_0/W_s or I_1/I_0 is not generally an integer but splitting is performed so that the expected value for the number is W_0/W_s or I_1/I_0 .

2.9.3 Path Stretching (Exponential Transform)

Path stretching is a variance reduction technique where the flight distance between collisions is stretched in the desired direction and is shrunk in the opposite direction. This technique is often called exponential transform and used in deep-penetration problems.

When the flight direction of a particle is determined after a collision, the distance to the next collision point x is sampled from the following exponential probability distribution:

$$p(x)dx = \Sigma_t e^{-\Sigma_t x} dx,$$

where Σ_t is the total cross section. Then a sampled distance s is given by

$$s = -\frac{\ln \xi}{\Sigma_t},$$

where ξ is a random number between 0 and 1. The sampled flight distance is smaller when the cross section is larger and conversely it is larger when the cross section is smaller.

Now we consider the sampling from the biased exponential probability distribution:

$$\begin{aligned} \tilde{p}(x)dx &= \tilde{\Sigma}_t e^{-\tilde{\Sigma}_t x} dx, \\ \tilde{\Sigma}_t &= (1 - \beta)\Sigma_t. \end{aligned}$$

Then a sampled distance \tilde{s} is given by

$$\tilde{s} = -\frac{\ln \xi}{\tilde{\Sigma}_t} = \frac{1}{1 - \beta} s.$$

If β is between 0 and 1, the total cross section becomes smaller and thus the sampled flight distance becomes larger. If β is, on the other hand, a negative value, the total cross section becomes larger and thus the sampled flight distance becomes smaller. Figure 2.8 illustrates the path stretching. The dotted circle represents the isotropic distribution of the flight distance s and the solid ellipse the biased distribution of the distance \tilde{s} . For a positive β value between 0 and 1, the distance is stretched. On the other hand, the distance is shrunk for a negative β value,

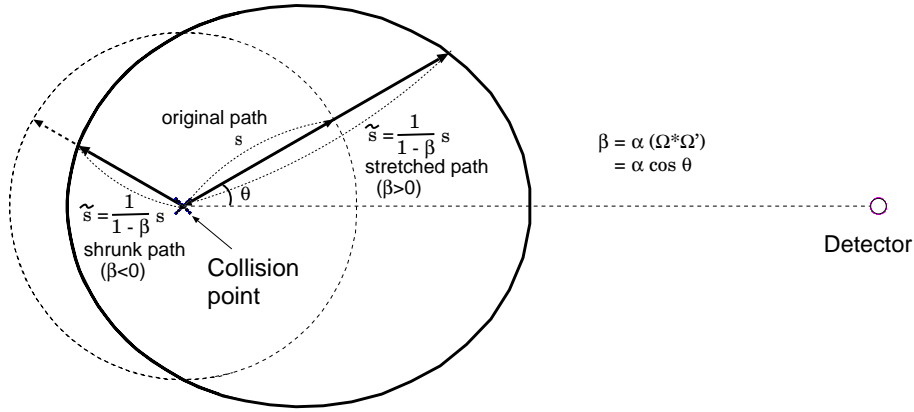


Fig. 2.8 Path stretching

In MVP/GMVP, β is determined by using the cosine of the angle between the direction from the particle position \mathbf{r} to a specified point \mathbf{c} (input data PSXYZ) and the flight direction Ω after scattering,

$$\begin{aligned} \beta &= \alpha \Omega \cdot \Omega', \\ \Omega' &= \frac{\mathbf{c} - \mathbf{r}}{|\mathbf{c} - \mathbf{r}|}, \end{aligned}$$

where α is the relaxation parameter (input data PSALP) which is specified in each spatial region (REGION).

To obtain unbiased results, the weight of the particle must be corrected as follows:

$$W' = \frac{p(x)}{\tilde{p}(x)} W = \frac{e^{-\beta \Sigma_t x}}{1 - \beta} W.$$

Now we consider a case where a particle moves into a different zone since the flight distance is sampled by each zone (ZONE) in MVP/GMVP. When the sampled flight distance x is larger than the distance to the zone boundary D , the particle enters another zone. When the biased flight distance is larger than D and the particle moves into another zone, the corrected weight is determined by using the probability $\int_D^\infty p(x) dx$ that the sampled flight distance is larger than D :

$$W' = \left(\int_D^\infty p(x) dx / \int_D^\infty \tilde{p}(x) dx \right) W = \frac{e^{-\Sigma_t D}}{e^{-(1-\beta)\Sigma_t D}} W = e^{-\beta \Sigma_t D} W \quad (\text{if } x > D).$$

2.9.4 Options for Variance Reduction

The following options can be specified for variance reduction in MVP/GMVP.

- RUSSIAN-ROULETTE: Only Russian roulette with weight is performed.
- IMPORTANCE: Russian roulette and splitting with importance are both performed.
- WEIGHT-WINDOW: Russian roulette and splitting with weight are both performed.
- PATH-STRETCH: Path stretching (exponential transform) is performed.

The third option is called “weight window” because only the particles with the weight of $W_k < W < 2W_s$ (weight window) can pass through a specified region unconditionally. Russian roulette or splitting is performed for other particles.

The basic usage of variance reduction techniques in MVP/GMVP is as follows.

- Importance is set relatively large for regions of interest in the case of “IMPORTANCE”.
- The weight for “WEIGHT-WINDOW” is set relatively “low” and “high” for important and less important regions, respectively.

2.10 Fixed-source and Eigenvalue Problems

There are two problems treated in MVP/GMVP: fixed-source problems and eigenvalue problems.

- Fixed-source problem

The information of external sources for random walk is given by the user. (the distribution functions for the position, energy etc. of source particles) The problems are solved in shielding calculations etc.

- Eigenvalue problem

The multiplication factor for a system including fissile materials is calculated as the eigenvalue problem. The problems are solved in reactor core calculations, criticality calculations etc.

The process for the fixed-source problems is relatively simple. It is repeated that particles are generated by sampling with the given source distribution and then walk around in a system. However, for eigenvalue problems the distribution of fission neutrons as the source is not known at the beginning of the calculation. Thus the following treatment is performed.

- (1) Given the guessed fission source distribution as the initial distribution, particles are generated from the distribution in the first batch (generation).
- (2) When collisions occur during the random walk, the number of fission neutrons is estimated and then the fission sites are stored for the specified number of the neutrons. In MVP, the fissioning nuclides and the energy of the colliding neutrons are also stored. The weight of the fission neutrons are given by the input data WGTF for each spatial region (See p. 258). The fission neutrons are not tracked in this batch. That is, the fission events are treated as absorption in eigenvalue problems.
- (3) The multiplication factor in a batch is calculated by the ratio of the number of fission neutrons to source neutrons[‡].

[‡]There are some methods to estimate the multiplication factor in MVP/GMVP. The details are described below.

- (4) The source neutrons in the second batch or later are unbiasedly selected from the fission source stored in the previous batch.

The above procedure is repeated for each batch. The eigenvalue is calculated from the averaged value of the multiplication factor in each batch. The repeated procedure is similar to the “outer iteration” in the deterministic method.

There are some problems in the above procedure.

- Strictly speaking, the calculated multiplication factors in different batches are not independent because the neutron source is obtained from the fission neutrons in the previous batch.
- The distribution of fission neutrons may be quite different from the correct one in the early batches and thus the multiplication factors in these batches are unreliable. But there are no good methods to know how many batches are required to obtain the correct distribution.

The user must take the following things into account.

- (a) The number of histories per batch (input data “NHIST”) should not be too small[§]. Empirically, the number should be more than a few thousand in general cases and more in large and complicated systems.
- (b) No results in the early batches are taken into account in the statistical analysis. These are called the “skipped batches”. In MVP/GMVP, the user can specify the number of the skipped batches (input data “NSKIP”). The number of the skipped batches should be relatively large for large and complicated systems.
- (c) There are some cases where sufficient fission neutrons are not stored for the next batch when the weight for secondary fission neutrons WGTF is larger than or nearly the same as the eigenvalue. In this case, more than one neutron are generated from the same fission site. However, this situation is not desirable because the statistical correlation is larger between batches. WGTF should be smaller than the predicted eigenvalue.
- (d) The fission distribution obtained for the other similar system should be used as the initial source if possible. This option is available in MVP/GMVP. See p. 207 to use the “#FISSIONFILE” option for the initial source.

MVP/GMVP has the function to estimate the variance of the eigenvalue with high reliability by taking into account the statistical correlation between batches. This is called the “estimation of the real variance”. The details are described in Section 4.4.

[§]The large number of histories per batch makes the calculation fast on vector and/or parallel computers.

3. Outline of Codes

This chapter describes the outline of functions and vector/parallel processing in the continuous-energy Monte Carlo code MVP and the multigroup Monte Carlo code GMVP.

3.1 Functions

The outline of functions in MVP and GMVP is described in the following. The details of the functions are described in the next chapter or the chapters about input/output.

(1) Problems to be solved

MVP/GMVP can solve eigenvalue problems and fixed-source problems. The multigroup code GMVP can solve forward and adjoint problems for neutron, photon and neutron-photon coupled transport. The continuous-energy code MVP can solve only the forward problems. Both codes can also perform time-dependent calculations.

(2) Geometry description

MVP/GMVP employs combinatorial geometry to describe the calculation geometry. It describes spatial regions by the combination of the 3-dimensional objects (BODies). Currently, the following objects (BODies) can be used.

- BODies with linear surfaces : half space, parallelepiped, right parallelepiped, wedge, right hexagonal prism
- BODies with quadratic surface and linear surfaces : cylinder, sphere, truncated right cone, truncated elliptic cone, ellipsoid by rotation, general ellipsoid
- Arbitrary quadratic surface and torus

The rectangular and hexagonal lattice geometry can be used to describe the repeated geometry. Furthermore, the statistical geometry model^{26,27)} is available.

(3) Particle sources

The various forms of energy-, angle-, space- and time-dependent distribution functions can be specified.

(4) Cross sections

The ANISN-type P_L cross sections or the double-differential cross sections²⁸⁾ can be used in the multigroup code GMVP. On the other hand, the special cross section libraries are used in the continuous-energy code MVP. The libraries are generated from the evaluated nuclear data (JENDL, ENDF/B, etc.) by using the LICEM code.²⁹⁾ The neutron cross sections in the unresolved resonance region are described by the probability table method. The neutron cross sections at arbitrary temperatures are available for MVP by just specifying the temperatures in the input data.

(5) Boundary conditions

Vacuum, perfect reflective, isotropic reflective (white), periodic boundary conditions can be specified.

(6) Variance reduction techniques

The basic variance reduction techniques Russian roulette kill and splitting are implemented. In addition, importance and weight window based on them are available. Path stretching and source biasing can be also used.

(7) Estimator

The track length, collision, point and surface crossing estimators are available. The eigenvalue is estimated by the track length, collision and analog estimators for neutron production and neutron balance methods. In the final estimation, the most probable value and its variance are calculated by the maximum likelihood method with the combination of the estimators.

(8) Tallies

GMVP calculates the eigenvalue, the particle flux and reaction rates by the spatial region, the energy group and the time bin, and their variances as the basic statistical parameters. In addition to these parameters, MVP calculates the effective microscopic and macroscopic cross sections and the corresponding reaction rates in the specified regions. Without the specification, these parameters are tallied for each spatial region but can be tallied for the arbitrary combination of the regions. When the LATTICE geometry is used, the option enables us to tally the parameters for any LATTICE, CELL, REGION and arbitrary combination of them. Furthermore, the results of the parameters can be used for the input of a drawing program or a burnup calculation code MVP-BURN after they are output to files.

(9) Drawing geometry

MVP/GMVP has a function to output the cross-sectional views of the calculation geometry as a text form. The CGVIEW code also draws the cross-sectional view on an arbitrary plane and output it on a display or in the postscript or encapsulated postscript form. These functions are effective for checking the calculation geometry.

(10) Burn-up calculation

The auxiliary code MVP-BURN implemented in the MVP/GMVP suite is available for burnup calculations.³⁰⁾

3.2 Vector Calculation

The conventional Monte Carlo codes for scalar computers employ the history-based algorithm. In this algorithm, particles are tracked one by one and the next particle is not tracked until the history of the previous particle is terminated. (See Fig 3.1.) On the other hand, the vectorized Monte Carlo codes have completely different algorithms called event-based algorithms, which are suitable for fast processing on vector supercomputers. There are some event-based algorithms and the MVP/GMVP code employs the stack-driven zone-selection algorithm.³¹⁾

In this algorithm, more than one particle (all particles in a batch) are tracked simultaneously. In order to increase the vectorization ratio, the particle data of which vector length is the longest is processed first. Thus the tracking of each particle is independent of the elapsed time of the particle. The unit of processing by the event is called “task” in MVP/GMVP and the flow of main tasks is shown in Fig 3.2. In all calculations, the first step is the source calculation (Source task) and the second step is the flight calculation (Flight task). After that, it cannot be predicted which task is selected because it depends on the number of particles queued for the tasks. In the program, there is an array which stores the particle

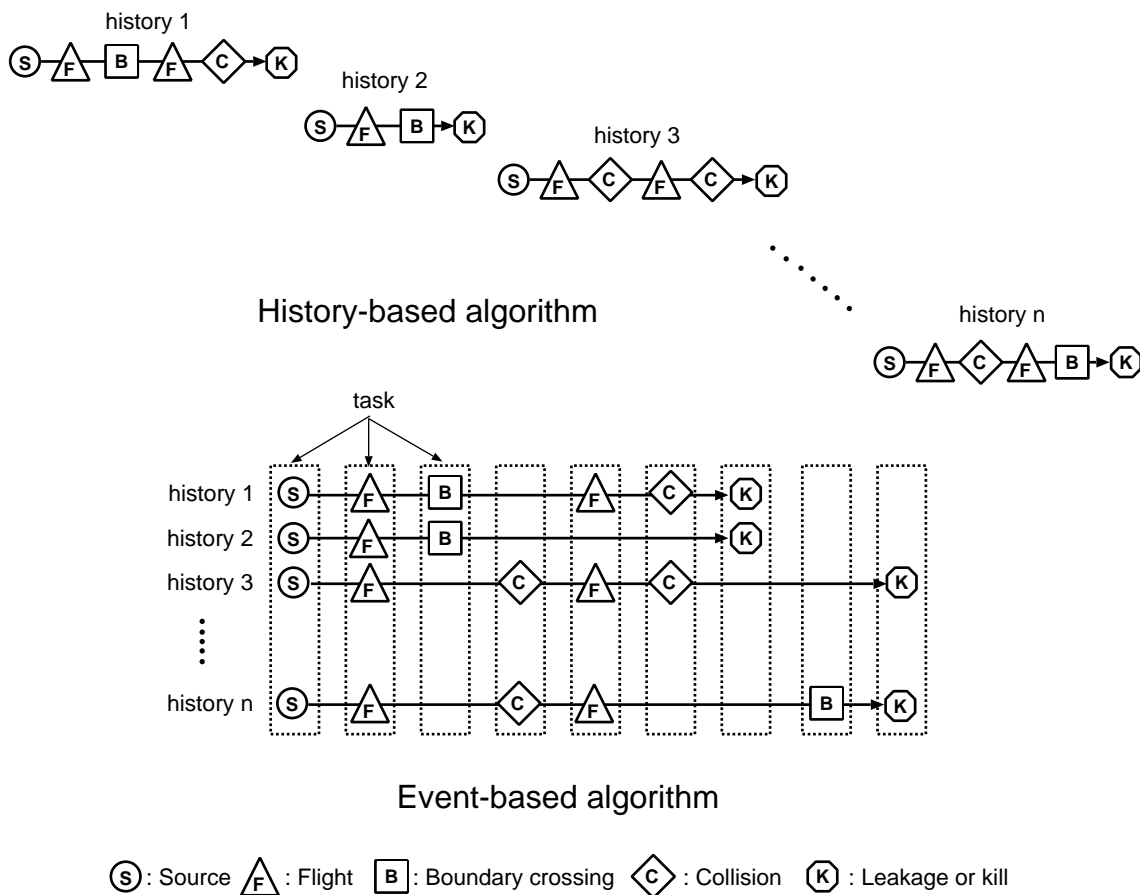


Fig. 3.1 Comparison of history- and event-based algorithms

number and other information of the particles waiting for the process of each task and it is called the “event stack”. There are also arrays which store the attributes (positions, energy, etc.) of all particles processed in a batch and they are called “particle banks”.

MVP/GMVP is as vectorized as possible and the vectorization ratio ((total CPU time in a scalar mode calculation)–(CPU time for scalar operations in a vector mode calculation)/(total CPU time in a scalar mode calculation)) is more than 95% in many cases. The speedup factor is greatly dependent on the vector length in the vector calculations. In the case of MVP/GMVP, since the vector length is approximately proportional to the number of particles in a batch (batch size), the fast calculation is possible with the large batch size. But since the required memory storage increases with the batch size, it should be determined by taking into account the available resources and the number of batches necessary for statistical processing. It is desirable that the batch size should be 10,000 or more if possible.

3.3 Parallel Calculation

MVP/GMVP can perform parallel calculations for multiple CPUs. Currently, MVP/GMVP can work on the shared memory parallel vector computers, the distributed memory parallel computers and the cluster computers connected with the network.

The method of parallel processing is somewhat different on each computer or each environment

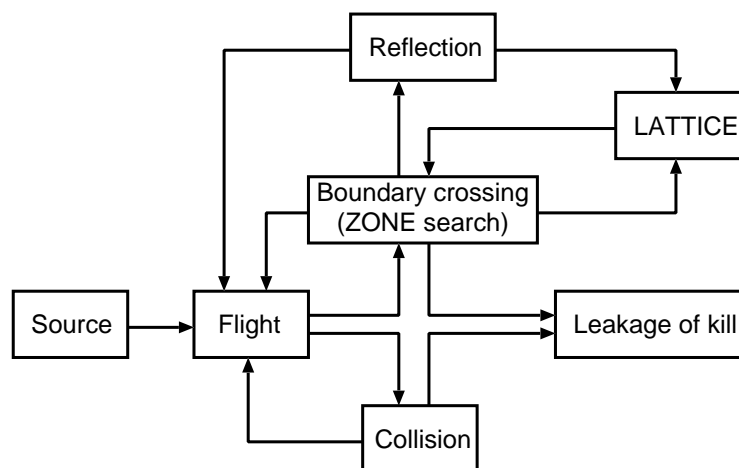


Fig. 3.2 Flow of tasks in MVP/GMVP

because the suitable method is strongly dependent on the environment. The basic procedure is however shown in Fig 3.3. First the number of processing elements (PEs) is set and input data is read. The number of batches per PE is determined for fixed-source problems or the batch size (the number of histories per batch) per PE is determined for eigenvalue problems. After a seed of an initial random number is generated for each PE and the data is sent to the PE, random walk is initiated. When all histories are terminated, the main process receives the results and performs statistical analysis.

In the parallel processing for fixed-source problems, each PE can process each batch independently since all batches are independent. Thus each PE processes N_b/n batches (N_b is the number of batches, n is the number of PEs). It is, however, impossible to process each batch independently in eigenvalue problems because of the correlation between the batches (neutron generations). Thus MVP/GMVP divides the batch size by the number of PEs, solve the divided eigenvalue problems independently and finally perform the statistical processing of the results as shown in Fig 3.4. In this case, the user should pay attention to the small batch size. It may cause the biased estimation of the eigenvalue and its variance in the eigenvalue problems or inefficient calculation on the vector computers.

The parallelization library MPI (Message Passing Interface)²⁾ is used for the computer cluster environment or the distributed memory parallel computers.

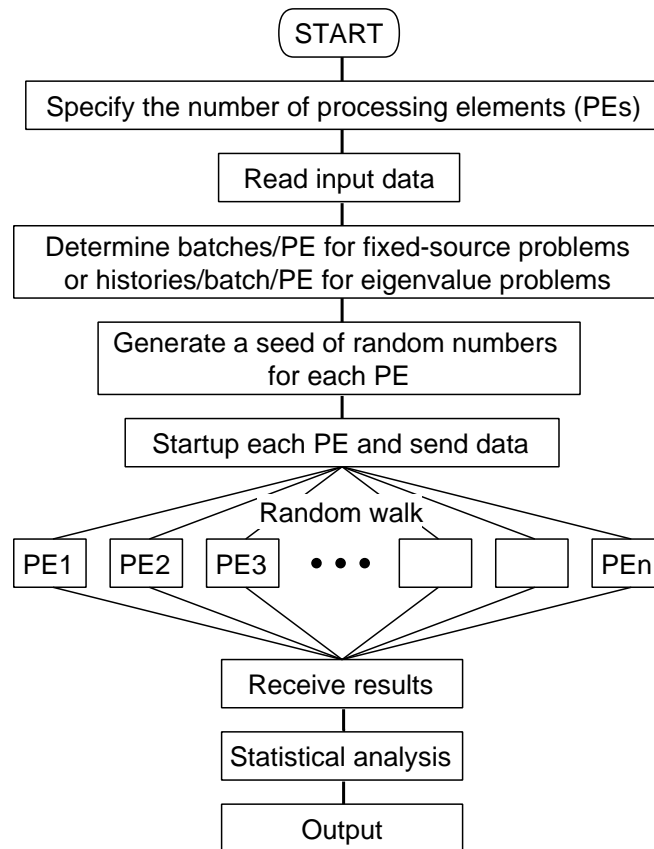


Fig. 3.3 Calculation flow of parallel processing

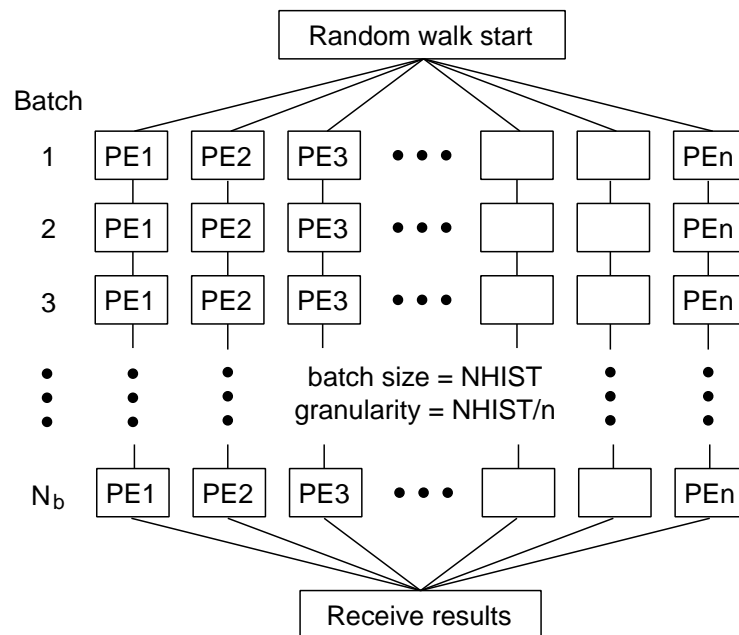


Fig. 3.4 Parallel processing for eigenvalue problems

4. Physics Model

4.1 Representation of Cross Sections

The MVP code can treat the neutron and photon reactions. The reaction cross sections and the type of the reactions are determined at any energy point for both particles.

4.1.1 Neutron Cross Sections

The energy range for nuclear reactions and transport of neutrons is from 10^{-5} eV to 20 MeV. All reactions given in the evaluated nuclear data files are treated explicitly in this energy range. The MVP cross section data files are generated for each nuclide from the evaluated nuclear data by using the LICEM code.^{29,32)} The data for the interpolation between temperatures are also included in the data files and thus cross sections at arbitrary temperatures are generated from the data.

The microscopic cross sections are provided at fine energy mesh points in the generated libraries. These energy mesh points are selected so that the requested accuracy of linearized cross sections is satisfied (normally 0.1%). The cross sections in the resolved resonance region are generated by the resonance formula given in the nuclear data file with the resonance parameters. The cross sections between the given energy points are determined by the linear-linear interpolation. The cross sections in the unresolved resonance region are determined by the probability table method. In this method, the sets of capture, elastic scattering and fission cross sections (σ_γ , σ_{el} , σ_f) and the probabilities p for the sets are given and the cross sections are determined by random numbers.

The energy and angular distributions of secondary neutrons after collision are given at each energy point with distribution functions or probability tables or based on the evaluated nuclear data file. The angular distribution is represented by the equally probable cosine bins, and is given in the center-of-mass system for the reactions for which the kinematics is available (elastic scattering and discrete level inelastic scattering) and in the laboratory system for the other reactions. If the energy distribution is given in the form of a distribution function, each parameter of the function is given for incident energies so that the function can be linearly interpolated as a function of energy.

In the table form of the energy distribution, the energy points after scattering are given in the library so that the distribution is described by the linear-linear interpolation. The probability table for the incident neutron of a given energy E is selected probabilistically from the tables at two adjacent energy points E_{i+1} , E_i that satisfy $E_{i+1} \leq E \leq E_i$. Namely, according to the specification of the interpolation method in the evaluated nuclear data, the probability table at E_i is selected with the probability

$$p_i = \frac{E - E_{i+1}}{E_i - E_{i+1}} \quad (4.1)$$

or

$$p_i = \frac{\ln(E/E_{i+1})}{\ln(E_i/E_{i+1})} \quad (4.2)$$

and the one at E_{i+1} with the probability $1 - p_i$. The former is the linear-linear interpolation. The latter is the log-linear interpolation. This scheme corresponds to the sampling from the probability distribution

$$P(E'; E) = p_i P(E'; E_i) + (1 - p_i) P(E'; E_{i+1}), \quad (4.3)$$

where E' is the secondary energy, $P(E'; E_i)$ and $P(E'; E_{i+1})$ are the probability tables for incident energies E_i, E_{i+1} , respectively.

In recent evaluated nuclear data files, new interpolation schemes, the method of corresponding points and the unit-base interpolation, are used for energy distributions of secondary particles. For both schemes, a secondary energy is sampled with the following approximate method. At first, a probability table is selected as mentioned above, and the secondary energy is sampled from the table. By assuming that both the tables are defined for the following ranges, respectively:

$$\begin{aligned} P(E'; E_i) & \text{ for } E'_{i,\min} \leq E' \leq E'_{i,\max}, \\ P(E'; E_{i+1}) & \text{ for } E'_{i+1,\min} \leq E' \leq E'_{i+1,\max}, \end{aligned}$$

the maximum and minimum energies of the distribution for the incident energy E are calculated as

$$E'_{\max} = p_i E'_{i,\max} + (1 - p_i) E'_{i+1,\max}, \quad (4.4)$$

$$E'_{\min} = p_i E'_{i,\min} + (1 - p_i) E'_{i+1,\min}. \quad (4.5)$$

Then, the secondary energy E' is determined as follows:

$$E' = E'_{\min} + \frac{E'_i - E'_{i,\min}}{E'_{i,\max} - E'_{i,\min}} (E'_{\max} - E'_{\min}), \quad (4.6)$$

when the table at E_i is selected and the sampled energy is E'_i .

$$E' = E'_{\min} + \frac{E'_{i+1} - E'_{i+1,\min}}{E'_{i+1,\max} - E'_{i+1,\min}} (E'_{\max} - E'_{\min}), \quad (4.7)$$

when the table at E_{i+1} is selected and the sampled energy is E'_{i+1} .

In the case where energy-angle distributions are given with the form of double-differential cross sections in File 6 of the ENDF-6 format,⁶⁾ MVP can treat following 3 distributions:

- (1) continuum energy-angle distribution based on the Kalbach systematics (File 6 of the ENDF-6 format, LAW=1 LANG=2),
- (2) N-body phase space distribution (File 6 of the ENDF-6 format, LAW=6),
- (3) tabular angle-energy distribution in the laboratory system (File 6 of the ENDF-6 format, LAW=7).

4.1.2 Photoatomic Cross Sections

The photoatomic data library for MVP includes the File 23 data (smooth cross sections) and the File 27 data (atomic form factors and scattering functions) given in the ENDF-6 format of the evaluated nuclear data files. MVP can treat all the reactions in the library explicitly. The photon transport calculation is performed in the energy range from 1 keV to 100 MeV. The cross sections at an arbitrary energy point are obtained by the log-log interpolation of the point-wise cross sections. Thus, the sum of the partial cross sections is used as the total cross section at an arbitrary energy point in practical calculations though the total cross section is given in the library.

Currently, two MVP photoatomic cross section libraries are available. One is converted from the MCNP photon cross section data file MCPLIB.^{33,34)} The other is generated from ENDF/B-VI.8. Since the ENDF/B-VI file does not contain the fluorescence X-ray data, the data in Reference 35 is used as for the converted library. However, fluorescence X-ray data for the elements whose atomic number is larger

than 94 are not provided in the reference. As an ad-hoc treatment, fluorescence X-ray data for plutonium ($Z=94$) are used for the heavier elements ($Z>94$).

The data for the generation of Bremsstrahlung photons by Compton recoil electrons, electrons and positrons from pair production based on the thick target approximation, are included in the MVP electron library. The library is generated from the MCNP4 electron data file EL1.³⁴⁾

4.1.3 Photonuclear Cross Sections

The photonuclear cross sections (photon interaction with nuclei) for MVP are generated from IAEA photonuclear data library.³⁶⁾ This library consists of photonuclear cross section data evaluated by several institutes such as JENDL(Japan), KAERI(Korea), CNDC(China), LANL(USA) and BOFOD(Russia). These evaluated nuclear data are described with the ENDF-6 format.⁶⁾ MVP photonuclear data library is generated with the modified LICEM system. The lower energy range of the photonuclear data depends on nuclides; the lowest one is about 1 MeV. The upper energy range depends on the institutes. They are 140 MeV, 140 MeV, 30 MeV, 150 MeV and 20 MeV for JENDL, KAERI, CNDC, LANL and BOFOD, respectively.

Photonuclear reactions addressed in MVP are listed in Table H.3 The number of the reactions (MT number) is 135 and is based on the ENDF-6 format. This number includes reactions reserved for future assignment. The total reaction cross section is defined as MT=3. The MT number of 3 (MT=3) is assigned as the sum of non-elastic reactions in the ENDF-6 format. It is the actual total cross section for photonuclear reactions. MVP calculates the total reaction cross section (MT=3) as the sum of MT numbers from 5 to 95 (MT=5–97) and that from 102 to 111 (MT=102–111).

If an energy-angle distribution is given in File 6 of the ENDF-6 format, MVP can treat the following distributions (LAW format):

- unknown distribution (LAW=0),
- continuum energy-angle distribution (LAW=1),
- distribution for two-body scattering (LAW=2),
- distribution for two-body recoils (LAW=4),
- N-body phase space distribution (LAW=6),
- table-form energy-angle distribution in the laboratory system (LAW=7).

If File 6 is given with LAW=0 and 4, no secondary particles are generated.

4.1.4 Preparation of Neutron Cross Sections at Arbitrary Temperatures

Temperature dependence of neutron cross sections in the resolved and unresolved resonance and thermal energy regions is important for accurate prediction of nuclear properties, while the cross sections in the high energy region are independent upon temperature. In the resolved resonance region, Doppler-broadened cross sections at any temperature can be easily obtained by using the method of SIGMA1 code³⁷⁾ in the LICEM system. In the unresolved resonance region, the cross sections are obtained with the interpolation method for the tabulated values at several fixed temperatures. In the thermal energy region, the thermal elastic scattering data are obtained with the interpolation scheme given in the evaluated nuclear data. The thermal inelastic scattering data are obtained with the interpolation method for cross sections, energy and angular distributions, which have been tabulated at several temperatures given in the evaluated data. The free gas model is directly treated with a specified temperature.

The MVP code uses the probability table method in the unresolved resonance region. In this method, the probability distribution is tabulated with appropriate probability bins selected for each energy and each temperature. For the interpolation, the probability tables are prepared in advance for each incident energy at 11 temperatures (293, 300, 393, 600, 900, 1200, 1500, 1800, 2100, 2500 and 3000 K). The MVP libraries have the restructured probability tables with common probability bins at the temperatures (See Fig 4.1(a)). The table at an arbitrary temperature is obtained by interpolation for cross sections of each probability bin as shown in Fig 4.1(b). The Akima method³⁸⁾ is used with a linear-log interpolation function; namely σ is a function of $\ln(T)$, where σ is the cross section and T is the temperature.

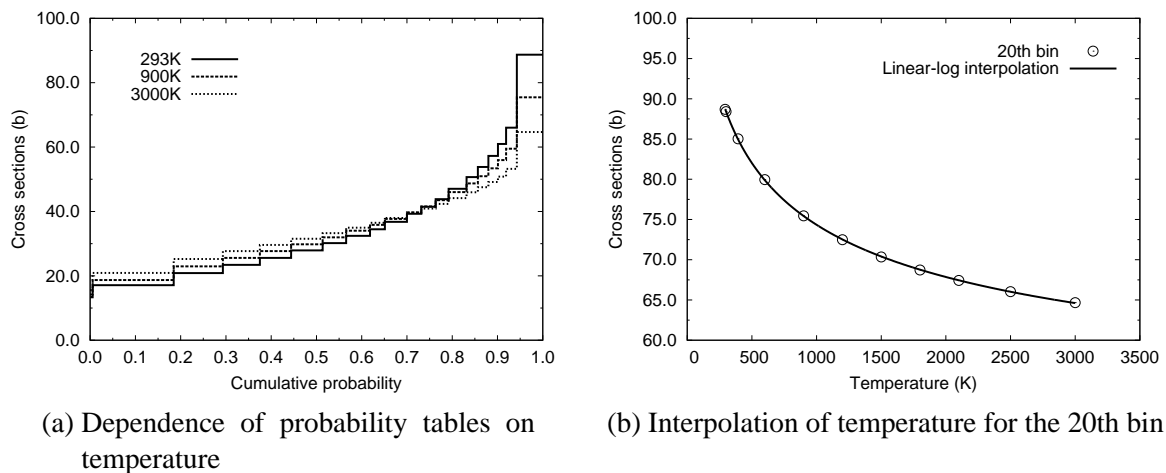


Fig. 4.1 Probability table of the U-235 capture cross section around 500 eV

The inelastic thermal scattering data are represented by cross sections, a secondary energy distribution and an angular distribution for each energy transfer from E_i to E_j . Common energy points are used for the energy distribution in all MVP libraries. The angular distribution is given as probabilities for five independent events which are isotropic scattering in forward and backward directions and scattering in three discrete directions ($\mu = 1, 0$ and -1) and the probabilities are calculated with the half-range moments of the scattering kernel.³⁹⁾ For the interpolation, the above data are prepared in advance at the temperatures at which the evaluated nuclear data are given. Secondary energy and angular distributions at an arbitrary temperature are obtained by interpolating cumulative probabilities for secondary energy and the moments of the angular distribution, respectively. The Akima method with a linear-linear function is used for the interpolation of the cross sections, the energy and angular distributions.

4.2 Analysis of Neutron Collision

Figure 4.2 shows the neutron reactions in the fast and thermal energy regions. Figure 4.3 shows the calculation flow of the collision analysis.

4.2.1 Selection of Collision Nuclide

In the continuous-energy method, the reaction of a particle with a matter is treated as the reaction with one of the constituent nuclides. Suppose that a neutron collides with the nuclide k in the medium

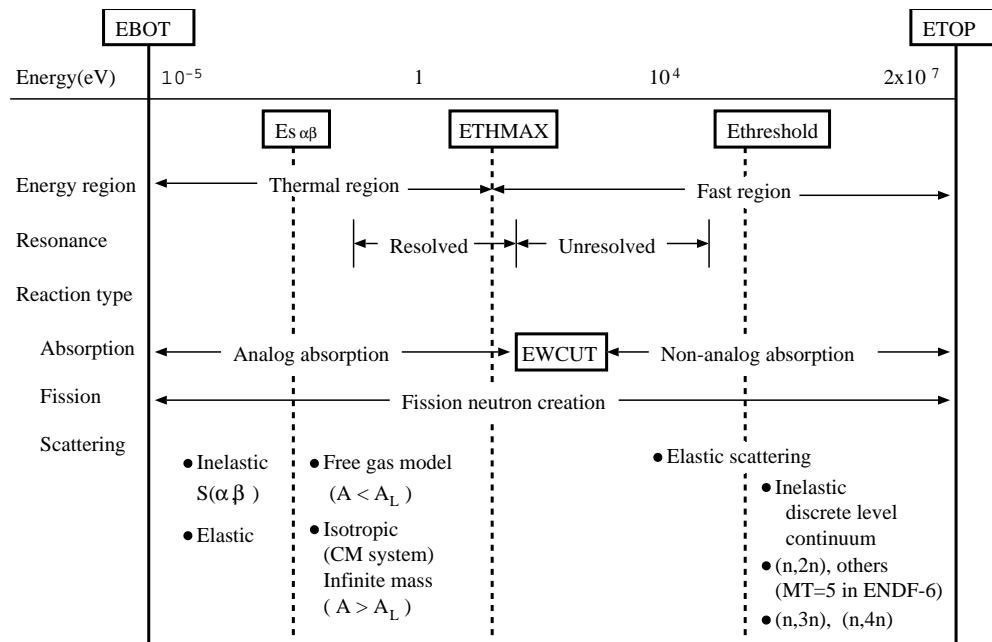


Fig. 4.2 Neutron reactions in collision treated in MVP

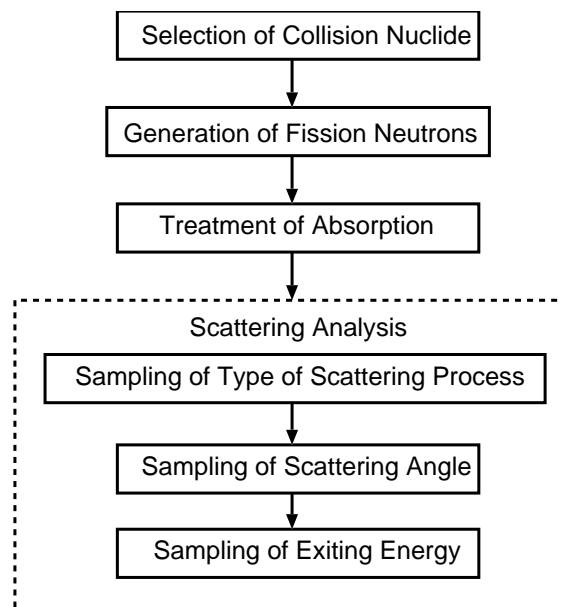


Fig. 4.3 Calculation flow of the collision analysis in MVP

comprised of K nuclides with the probability p_k ,

$$p_k = \frac{\rho^k \sigma_t^k}{\sum_{j=1}^K \rho^j \sigma_t^j} = \frac{\rho^k \sigma_t^k}{\Sigma_t}, \quad (4.8)$$

where $\rho^k, \sigma_t^k, \Sigma_t$ are the atomic number density, microscopic and macroscopic total cross section, respectively.

There are some codes that calculate the cumulative probability density distribution for the nuclide and determine a collision nuclide by using the cumulative probabilities and a random number. But this scheme is inefficient for MVP/GMVP from the viewpoint of memory storage and vector/parallel processing because MVP/GMVP performs this process for more than one particle simultaneously and thus the cumulative probabilities for all the particles must be stored.

In order to improve the efficiency, a new method has been developed. In the method, when the partial sum $\Sigma^{(k)} \equiv \sum_{j=1}^k \rho^j \sigma_t^j$ is obtained in the calculation process of the macroscopic total cross section, a random number ξ_k is generated for each particle. If the random number satisfies the condition $\xi_k \Sigma^{(k)} < \rho^k \sigma_t^k$, then the nuclide k is stored as a candidate of the collision nuclide. This process is repeated K times and the collision nuclide is determined as the last nuclide that satisfies the above condition. This method requires a lot of random numbers but the partial sum $\Sigma^{(k)}$ is not needed to be stored and all calculations are suitable for vector/parallel processing.

4.2.2 Generation of Fission Neutrons

The expectation value of the total weight for all fission neutrons in collision is $W\nu\sigma_f/\sigma_t$ in the medium including a fissile nuclide. W is the weight of the neutron before collision. The weight of all fission neutrons is W_f given by the input data WGTF. Then the number of generated neutrons n is determined probabilistically so that the expectation value $E[n]$ is conserved.

$$E[n] = \frac{W\nu\sigma_f}{\sigma_t} \frac{1}{W_f} \quad (4.9)$$

Letting n_1 be the maximum integer less than $E[n]$, $n = n_1 + 1$ is selected with the probability $p = E[n] - n_1$ and $n = n_1$ with the probability $1 - p$.

The energy of fission neutrons is sampled from the fission spectrum given in the table form or by the probability distribution function. In the latter case, the energy is determined uniquely with more than one random number. For example, in the case of the Maxwell distribution, it is determined with 3 random numbers ξ_1, ξ_2, ξ_3 ,

$$E = -R(E') \left[\ln \xi_1 + \ln \xi_2 \cdot \cos^2 \left(\frac{\pi}{2} \xi_3 \right) \right], \quad (4.10)$$

where $R(E')$ is the constant dependent on the incident energy E' . The angular distribution is assumed to be isotropic.

4.2.3 Absorption

There are two methods to treat absorption used in the continuous-energy Monte Carlo codes including MVP. They are “analog absorption” and “non-analog absorption”. Analog absorption is usually used in the low-energy region and the random walk of particles is terminated with the absorption probability σ_a/σ_t . On the other hand, non-analog absorption is used in the high-energy region or the multigroup

method and the particles continue the random walk with the weight reduced by the absorption probability instead of finishing the history.

The boundary energy of these 2 methods is given by the input data EWCUT. The default value of EWCUT is 275 eV (the same as VIM²¹) in the current MVP code but it is desirable to change EWCUT depending on problems to improve calculation efficiency. For example, EWCUT should be smaller when a system has a large absorption probability in the low energy region and the information in that region is important. On the other hand, the calculation efficiency may be improved with larger EWCUT for a system where scattering is mainly important in the high energy region.

4.2.4 Scattering

The probability that the reaction x occurs $p_x = \sigma_x / (\sigma_t - \sigma_a)$ is used to determine the type of scattering. x is sampled from the cumulative probability table and determined so that x satisfies

$$\sum_{j=1}^{x-1} p_j < \xi \leq \sum_{j=1}^x p_j, \quad (4.11)$$

where ξ is a random number. If the (n,mn) reaction ($m > 1$) is selected, $(m - 1)$ neutrons are newly generated.

In the case of elastic scattering, the energy and the flight direction Ω after scattering are determined by the conservation laws of energy and momentum in collision (kinematics). At first, the cosine of the scattering angle in the center-of-mass system μ_{cm} is determined. It is sampled from the probability table in the form of equally probable cosine bins in the MVP library that is generated from the angular distribution at each energy point in the evaluated nuclear data files. Since the probability table at an arbitrary incident energy E' is required in general, either of the tables at two adjacent energy points to E' is used in the same way as described in Section 4.1.1. The uniform distribution is assumed in an angular bin. After μ_{cm} is determined, the energy after scattering is determined by the conservation laws of energy and momentum,

$$E = \frac{E'(A^2 + 2A\mu_{\text{cm}} + 1)}{(A + 1)^2}, \quad (4.12)$$

where A is the ratio of the mass of the atom to that of the neutron. In order to transform μ_{cm} into the scattering angle μ_{lab} in the laboratory system, the following equation is used.

$$\mu_{\text{lab}} = \frac{A\mu_{\text{cm}} + 1}{\sqrt{A^2 + 2A\mu_{\text{cm}} + 1}} \quad (4.13)$$

Since the scattering is isotropic about the azimuthal angle ω , it is determined with a random number ξ by

$$\omega = 2\pi\xi. \quad (4.14)$$

The obtained μ_{lab} is the relative angle to the flight direction of the particle, and then the flight direction is calculated in the (x, y, z) coordinates.

In the case of discrete level inelastic scattering, μ_{cm} is sampled from the probability table in the same way as the elastic scattering. The energy after scattering E is determined by the following equation.

$$E = \frac{E'}{(A + 1)^2} [\gamma^2(E') + 2\gamma(E')\mu_{\text{cm}} + 1], \quad (4.15)$$

where

$$\gamma(E') = A \sqrt{1 + \frac{A+1}{A} \frac{Q_i}{E'}}, \quad (4.16)$$

Q_i is the Q-value and the negative value in this case. The scattering angle in the laboratory system μ_{lab} is obtained by

$$\mu_{\text{lab}} = \frac{\gamma(E')\mu_{\text{cm}} + 1}{\sqrt{\gamma^2(E') + 2\gamma(E')\mu_{\text{cm}} + 1}}. \quad (4.17)$$

In the case of continuum inelastic scattering, the energy distribution is usually given in the table form or the evaporation model. In the former case, the energy E is determined from the probability table with a random number. In the case of the evaporation model, the energy distribution is given by

$$f(E; E') = C \cdot E e^{-E/T(E')}, \quad (4.18)$$

where E' is the incident energy and $T(E')$ the nuclear temperature. E is thus determined with two random numbers ξ_1 and ξ_2 by

$$E = -T(E') \ln(\xi_1 \cdot \xi_2), \quad 0 \leq E \leq E' - U, \quad (4.19)$$

where U is the constant given by the evaluated nuclear data files. If $E > E' - U$, then E is sampled again. The scattering angle is sampled from the probability table in the form of equally probable cosine bins in the laboratory system.

In the case of the (n,mn) reaction, the energy after scattering and the scattering angle are sampled in the same way as the continuous level inelastic scattering. m neutrons are emitted with the same weight as the one before collision so that the neutrons are uncorrelated with each other.

4.2.5 Thermal Neutron Scattering

The scattering of a thermal neutron with a matter is treated as a special scattering in MVP. In the thermal neutron scattering, the following models can be selected by the input parameters and nuclear data as shown in Fig 4.2.

- Free gas model (free monatomic gas model)
- Infinite mass model
- Elastic scattering
- Inelastic scattering with $S(\alpha, \beta)$

When the thermal scattering cross section data $S(\alpha, \beta)$ is not given in the library, $E_{S\alpha\beta} = 0$ and the thermal neutron scattering is treated by the free gas or infinite mass models for such nuclides.

In the free gas model, the velocity of the target nucleus (the speed and moving direction) is determined from the probability distribution. Then the energy and flight direction of the scattered neutron are determined by the kinematics of the two-body elastic collision (conservation laws of energy and momentum).

The infinite mass model is used when the atomic mass of the target nucleus is larger than the specified value. In this model, scattering is isotropic in the center-of-mass system. Neutrons slow down by elastic

scattering in the energy region of not less than 0.625 eV* and the energy of the neutrons do not change in the energy region of less than 0.625 eV.

The thermal elastic scattering includes coherent and incoherent elastic scattering models used in the thermal library of the ENDF-6 format files. MVP can treat both the models. The energy of neutrons does not change before and after the scattering. The scattering angle is determined from the angular distribution given in the laboratory system in the incoherent scattering, while the Bragg edges and structure factors are used in the coherent one.

In order to take into account the scattering from an atom bound in a molecule or crystal lattice, the scattering law $S(\alpha, \beta)$ is provided for some nuclides. The law is represented by the probability of the energy transfer and the corresponding angular distribution table in the MVP libraries. The energy bin after scattering is determined by the discrete conditional sampling method⁴⁰⁾ and the scattering angle is determined from the corresponding angular distribution.

4.2.6 Resonance Elastic Scattering

As has already been stated in Section 4.2.4, an energy and a flight direction after elastic scattering are determined with Eqs. (4.12), (4.13) and (4.14). This elastic scattering model is called the asymptotic slowing-down scattering model (asymptotic model), which includes the assumption that the thermal motion of a target nucleus is not taken into account. This model has been conventionally used for the elastic scattering in the resonance energy region. However, recent studies^{41, 42, 43, 44)} reveal that this approximation gives a significant impact on the calculation of reactor physics parameters including Doppler reactivity worth. To overcome the drawback, an exact resonance scattering model has been implemented into the MVP code.⁹⁾ The theoretical background of the model is described in the following.

Taking into account the thermal motion of the target nucleus, the probability that a neutron with velocity \mathbf{v} suffers scattering with a target nucleus with velocity \mathbf{V} can be written as

$$P(V, \mu) dV d\mu = \frac{v_r \sigma_s(v_r)}{2v \sigma_s^{\text{eff}}(v)} M(V) dV d\mu, \quad (4.20)$$

where

$$v = |\mathbf{v}|, V = |\mathbf{V}|, \mu = \frac{\mathbf{v} \cdot \mathbf{V}}{|\mathbf{v}| |\mathbf{V}|}.$$

σ_s and σ_s^{eff} are the scattering cross section at 0 K and the Doppler-broadened scattering cross section, respectively; $v_r = |\mathbf{v} - \mathbf{V}|$ is the relative speed between the neutron and the target; $M(V)$ is the Maxwellian distribution of the target speed.

The outgoing velocity \mathbf{v}' of the neutron from the scattering can be written as

$$\mathbf{v}' = \frac{1}{A+1} (\mathbf{v} + A\mathbf{V}) + \frac{A}{A+1} |\mathbf{v} - \mathbf{V}| \boldsymbol{\Omega}'_{\text{cm}}, \quad (4.21)$$

where \mathbf{v} and \mathbf{V} are the velocities of the neutron and the target before scattering in the laboratory system; $\boldsymbol{\Omega}'_{\text{cm}}$ is the flight direction of the scattered neutron in the center-of-mass system; A is the atomic mass of the target nucleus to the neutron. One can calculate the energy-angle distribution of the scattered neutron with Eqs. (4.20) and (4.21).

If one can sample V and μ from the probability density function of Eq. (4.20), no approximation is required. However, σ_s cannot be expressed in a general analytical form and the integral for σ_s^{eff} cannot

*The value is defined internally in the code.

be performed analytically. A conventional approximation assumes that the scattering cross section at 0 K (σ_{s0}) is constant. The Doppler-broadened scattering cross section can, then, be written as

$$\sigma_s^{\text{eff}}(v) = \frac{\sigma_{s0}}{\beta^2} \left[\left(\beta^2 + \frac{1}{2} \right) \text{erf} \beta + \sqrt{\frac{1}{\pi}} \beta \exp(-\beta^2) \right], \quad (4.22)$$

where

$$\beta^2 = \frac{AM_n v^2}{2kT}. \quad (4.23)$$

erf is the error function; M_n , E , k and T are the neutron mass, the neutron energy, the Boltzmann constant and the equilibrium temperature of the target, respectively. Then Eq. (4.20) can be rewritten as

$$P_{\text{const}}(V, \mu) dV d\mu = \frac{v_r}{\frac{2v}{\beta} \left[\left(\beta^2 + \frac{1}{2} \right) \text{erf} \beta + \sqrt{\frac{1}{\pi}} \beta \exp(-\beta^2) \right]} M(V) dV d\mu. \quad (4.24)$$

This probability density function is used in the free gas model described in Section 4.2.5. Many Monte Carlo codes use this model for thermal neutron scattering together with the assumption of isotropic scattering for Ω'_{cm} in Eq. (4.21). In the MVP code, the model which uses $P_{\text{const}}(V, \mu) dV d\mu$ without the assumption of isotropic scattering is referred to as the free gas model with constant cross section (constant cross section model).

The approximation of the constant cross section model is valid for light nuclides but not for heavy nuclides because resonances exist down to about 1 eV. To address the exact probability density function of Eq. (4.20), the weight correction method⁴³⁾ is used in the MVP code. In this method, the thermal motion of the target nucleus, V , is sampled from Eq. (4.22) and the neutron weight is corrected with a factor f :

$$f = \frac{P(V, \mu)}{P_{\text{const}}(V, \mu)} = \frac{\sigma_s(v_r)}{\sigma_s^{\text{eff}}(v)} g(\beta), \quad (4.25)$$

where

$$g(\beta) = \frac{\left(\beta^2 + \frac{1}{2} \right) \text{erf} \beta + \sqrt{\frac{1}{\pi}} \beta \exp(-\beta^2)}{\beta^2}. \quad (4.26)$$

The factor $g(\beta)$ is actually nearly equal to unity. The MVP code thus uses the following correction factor

$$f = \frac{\sigma_s(v_r)}{\sigma_s^{\text{eff}}(v)} \quad (4.27)$$

The largest difference between Eq. (4.25) and Eq. (4.27) is less than 0.005% for the U-238 resonance elastic scattering at 1,000 K. In the MVP code, this model is referred to as the free gas model with resonance cross section (exact model).

The correction factor sometimes fluctuates the neutron weight significantly from unity. MVP automatically applies the weight window method to suppress the fluctuation if the exact resonance scattering model is used. It should be noted that this weight window method is applied independently of the weight

window specified by the WEIGHT-WINDOW option. (See Section 2.9.4 and Chapter 7.) The weight window parameters for the resonance scattering model are set internally in the code such that the neutron weight after scattering should be not less than half of the weight before scattering and should be not larger than twice of it (The upper limit of the weight is 2.0). The user should pay attention to the fact that the statistics for the weight window in the event monitor includes both the results with the weight windows for the resonance scattering model and the normal WEIGHT-WINDOW option. (See Section 16.5.4 for the output of the event monitor.)

4.2.7 Speed of Neutrons

The speed of neutrons is required for time-dependent tallies. In the case of MVP, the speed of neutrons is calculated from the following equation based on the special relativistic theory:

$$V_n(E_n) = \frac{c\sqrt{E_n(E_n + m_0c^2)}}{E_n + m_0c^2}, \quad (4.28)$$

where

$$\begin{aligned} V_n(E_n) &: \text{neutron speed (cm/s) of energy } E_n \text{ (eV)} \\ m_0c^2 &= \frac{m_0(\text{g})[c(\text{cm/s})]^2}{e(\text{erg/eV})} \\ c &= 2.99792458 \times 10^{10} : \text{speed of light (cm/s)} \\ m_0 &= 1.6749286 \times 10^{-24} : \text{rest mass of neutron (g)} \\ e &= 1.60217733 \times 10^{-12} \text{ (erg/eV).} \end{aligned}$$

In the case of GMVP, the speed of neutrons is calculated as follows:

$$\begin{aligned} V_{n,g} &= \frac{\int_{\Delta E_g} \frac{1}{E} dE}{\int_{\Delta E_g} \frac{1}{V_n(E)E} dE} \quad (4.29) \\ &= c \left| \ln \frac{E_u}{E_\ell} / \left[\ln \left(E_u + m_0c^2 + \sqrt{E_u(E_u + 2m_0c^2)} \right) - \frac{\sqrt{E_u(E_u + 2m_0c^2)}}{E_u} \right. \right. \\ &\quad \left. \left. - \ln \left(E_\ell + m_0c^2 + \sqrt{E_\ell(E_\ell + 2m_0c^2)} \right) + \frac{\sqrt{E_\ell(E_\ell + 2m_0c^2)}}{E_\ell} \right] \right|, \quad (4.30) \end{aligned}$$

where $V_{n,g}$ is the neutron speed of the g -th group. E_u and E_ℓ are the upper and lower energies of the g -th group, respectively.

4.3 Estimation of Eigenvalue

The eigenvalue k_{eff} (multiplication factor) is estimated by two different methods in the eigenvalue problems and finally the combined results (estimated eigenvalue and its variance) is output. One is calculated from the definition of neutron multiplication as follows:

$$\begin{aligned} k_{\text{eff}} &= \frac{\sum \langle \text{Weight of neutrons created by fission reaction} \rangle}{\sum \langle \text{Weight of source neutrons} \rangle} \\ &= \frac{\int \nu \Sigma_f \phi dE dr}{\sum \langle \text{Weight of source neutrons} \rangle}. \quad (4.31) \end{aligned}$$

The other is based on the equation of neutron balance and the eigenvalue is estimated by

$$\begin{aligned}
 k_{\text{eff}} &= \frac{\sum \langle \text{Weight of neutrons created by fission reaction} \rangle}{\sum \langle \text{Leakage} \rangle + \sum \langle \text{net absorption} \rangle} \\
 &= \frac{\int \nu \Sigma_f \phi dE dr}{\sum \langle \text{Leakage} \rangle + \int \Sigma_{\text{loss}} \phi dE dr},
 \end{aligned} \tag{4.32}$$

where

$$\Sigma_{\text{loss}} = \Sigma_a - \Sigma_{n,2n} - 2\Sigma_{n,3n} - 3\Sigma_{n,4n}. \tag{4.33}$$

The terms including the cross sections in the above equations are calculated by three different estimators of the track length, collision and analog estimators in MVP. The analog estimator is the same as the collision estimator but the microscopic cross section of only the collision nuclide are used for the estimation of the macroscopic reaction rates. The other estimators use the macroscopic cross sections calculated from the microscopic cross sections of all nuclides in the collision material.

No analog estimator is used in GMVP but the track length and collision estimators are used.

As a result, the eigenvalue is estimated by six and four different methods in MVP and GMVP, respectively. In MVP/GMVP, the most confident eigenvalue is obtained by the method of maximum likelihood with these estimators, where they are combined with weights so that the variance is minimum. Since these estimators are correlated with each other, the combined results are calculated by the following equations.

$$k_{\text{eff}} = \frac{\sum_i \sum_j T_{ij} X_j}{\sum_i \sum_j T_{ij}}, \tag{4.34}$$

$$\sigma = \frac{1}{\sum_i \sum_j T_{ij}}, \tag{4.35}$$

where X_i is the result of each estimator and T_{ij} is the element of the inverse matrix of covariance estimated by the results in each batch. MVP/GMVP outputs the estimated value by each estimator, correlation coefficients, results by the maximum likelihood method for the neutron production and neutron balance methods and the eigenvalues and their variances obtained by the method of maximum likelihood for all the estimators.

The user can calculate not only the eigenvalue but also other physical quantities with options. The fluxes are estimated by the track length and collision estimators. The reaction rates, effective microscopic and macroscopic cross sections are estimated by either of the estimators.

4.4 Estimation of Real Variance in Eigenvalue Problems

It has been pointed out that the variance of the eigenvalue etc. is theoretically underestimated by statistical analysis used in many Monte Carlo codes because the calculated result in each batch is assumed to be uncorrelated between batches in the eigenvalue calculations. In order to estimate this underestimation quantitatively, Ueki et al. proposed the method to estimate the variance including the correlation between batches.⁴⁵⁾ The MVP/GMVP code uses this method to estimate the “real variance”, that is, real statistical uncertainty and provide the information to judge the underestimation of the variance obtained by the conventional method.

The physical quantities are estimated by the statistically averaged values of the results in the batches (neutron generations in the eigenvalue problems). In the case of eigenvalue,

$$k = \frac{1}{N} \sum_{i=1}^N k_i, \quad (4.36)$$

where k_i is the neutron multiplication factor in each batch and N is the number of batches.

The real variance (σ_R^2) of k is the variance of the population from which eigenvalues are sampled in the independent Monte Carlo calculations:

$$\sigma_R^2 = E[(k - E(k))^2] = E[k^2] - E[k]^2, \quad (4.37)$$

where $E[X]$ is the expectation value of X .

On the other hand, the sample variance σ_S^2 , which is used in many Monte Carlo codes including MVP/GMVP, is estimated in a single Monte Carlo calculation as follows:

$$\sigma_S^2 = \frac{1}{N(N-1)} \sum_{i=1}^N (k_i - k)^2 = \frac{1}{N(N-1)} \left(\sum_{i=1}^N k_i^2 - Nk^2 \right). \quad (4.38)$$

Here the apparent variance σ_A^2 is defined as the expectation value of the sample variance,

$$\sigma_A^2 = E[\sigma_S^2]. \quad (4.39)$$

Unless the sample variance σ_S^2 is biased, the apparent variance σ_A^2 is identical with the real variance σ_R^2 . But this is not true in general because the correlation between batches is neglected in the estimation of the sample variance.

Assuming that the values of $E[k_i]$ and $E[k_i^2]$ are independent of batch i , the difference between the real and apparent variances is obtained as follows:

$$\sigma_A^2 - \sigma_R^2 = -\frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^N \text{cov}[k_i, k_j], \quad (4.40)$$

where $\text{cov}[k_i, k_j]$ is the intercycle covariance of k_i and k_j . If the calculation is in equilibrium, we can assume that the intercycle covariance between batches depends only on the cycle difference. Thus we introduce the real lag i covariance $C_R[i]$:

$$C_R[i] = \text{cov}[k_m, k_{m+i}], \quad m = 1, \dots, N-i. \quad (4.41)$$

Then the difference between the real and apparent variances is as follows:

$$\sigma_A^2 - \sigma_R^2 = -\frac{2}{N(N-1)} \sum_{i=1}^{N-1} (N-i)C_R[i]. \quad (4.42)$$

We define the following sample lag i covariance $C_S[i]$:

$$C_S[i] = \frac{1}{N-i-1} \sum_{j=1}^{N-i} (k_j - k)(k_{j+i} - k) \quad (4.43)$$

and the apparent lag i covariance $C_A[i]$ as the expected value of the sample lag i covariance:

$$C_A[i] = E[C_S[i]]. \quad (4.44)$$

Then we can obtain the following relation:

$$C_R[i] = \frac{N-i-1}{N-i} C_A[i] + \sigma_R^2 - \frac{2}{N(N-i)} \sum_{j=1}^i \left(\sum_{m=1}^{j-1} + \sum_{m=1}^{N-j} \right) C_R[m] + \frac{2i(N-1)}{N(N-i)} (\sigma_R^2 - \sigma_A^2). \quad (4.45)$$

When $i \ll N$, the above equation becomes

$$C_A[i] - C_R[i] \approx -\sigma_R^2. \quad (4.46)$$

By using these equations, we estimate the real variance including the correlation between batches in the following procedure. First, we specify N_0 and assume that the lag i covariance for a lag of more than N_0 can be neglected. Then,

$$C_A[i] = C_S[i] = C_R[i] \approx 0, \quad i > N_0. \quad (4.47)$$

On the basis of this assumption, we estimate σ_R^2 that satisfies the relation between the apparent and real variances (Eq. (4.42)) and the approximated relation between the apparent and real covariances (Eq. (4.46)).

As is clear in the above discussion, it is assumed that the number of batches is sufficiently large and the estimation should be performed after the source distribution becomes the fundamental mode in this method. Unless the assumptions are satisfied, the estimation of the real variance is not reliable.

4.5 Reactor Noise Analysis

Noise analysis methods are often used to measure the subcriticality of nuclear systems. The MVP/GMVP code has the capability of the Feynman- α method⁴⁶⁾, which is one of time-domain noise analysis methods. In the method, variance-to-mean ratios (Feynman's Y values) are calculated for the distribution of neutron counts with the gate width. To use the method, the analog Monte Carlo simulation where the neutron weight is always unity must be performed because the method is based on the correlation of neutrons. Therefore, the following functions are implemented into the code:

- sampling the number of fission neutrons from the distribution given by Terrell,⁴⁷⁾
- scoring analog events for scattering, capture, fission and absorption,
- periodic-time boundary condition,
- generation of multiple source neutrons at the same time for spontaneous fission source.

The number of fission neutrons from Terrell's distribution is sampled as follows. The cumulative probability distribution for the number of neutrons per fission is given by the following equation:

$$\begin{aligned} \sum_{n=0}^{\nu} P(n) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{(\nu-\bar{\nu}+1/2+b)/\sigma} \exp\left(-\frac{1}{2}t^2\right) dt \\ &= \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{(\nu+1/2)} \exp\left(-\frac{(x-\bar{\nu}+b)^2}{2\sigma^2}\right) dx, \end{aligned} \quad (4.48)$$

where n is the number of fission neutrons, $\bar{\nu}$ is the average number of neutrons and σ is the root-mean-square width of the initial total excitation-energy distribution. b is the adjustment factor such that the average value of n is $\bar{\nu}$. Namely b must satisfy the following equation:

$$\sum_{n=0}^{\infty} nP(n) = \bar{\nu}. \quad (4.49)$$

Then b is approximated as

$$b = \frac{1}{\sqrt{2\pi}} \int_{(\bar{\nu}+1/2)/\sigma}^{\infty} \exp\left(-\frac{1}{2}t^2\right) dt. \quad (4.50)$$

Since Eq.(4.48) is a normal distribution with a mean value of $\bar{\nu} - b$ and a standard deviation of σ , the number of fission neutrons is determined as follows:

$$n = \left[\bar{\nu} + \frac{1}{2} \right], \quad (4.51)$$

where the Gauss symbol $[x]$ means the maximum integer not larger than x .

To calculate Feynman's Y values, the MVP/GMVP code obtains a time spectrum for each detector tally with time bins corresponding to a minimum gate width. Then the code calculates the mean counts and mean-square counts for the different gate widths from time spectra obtained by bunching the original spectrum (See Fig 4.4.). The mean count with the gate width τ_n is obtained by

$$\overline{C(\tau_n)} = \frac{1}{N_n} \sum_{i=1}^{N_n} C_i^{\tau_n} \quad (4.52)$$

and the mean-square count is obtained by

$$\overline{C^2(\tau_n)} = \frac{1}{N_n} \sum_{i=1}^{N_n} (C_i^{\tau_n})^2, \quad (4.53)$$

where N_n is the number of time bins for the bunched spectrum of the gate width τ_n and $C_i^{\tau_n}$ is the number of counts for the i -th bin of the gate width τ_n . Using the above counts, the Y value is calculated as follows:

$$Y(\tau_n) = \frac{\sigma^2(\tau_n)}{\overline{C(\tau_n)}} - 1 \quad (4.54)$$

$$\sigma^2(\tau_n) = \frac{N_n}{N_n - 1} \left[\overline{C^2(\tau_n)} - \overline{C(\tau_n)}^2 \right]. \quad (4.55)$$

The statistical uncertainty of the Y value is estimated by the batch method.

4.6 Photon Reaction

The generation of photons in MVP is divided into two cases; one is generation from fixed sources and the other is from the nuclear reaction of neutrons. For the latter case, the yield, energy and angular distributions at each incident energy of a neutron are given in the neutron cross section libraries.

The photon interaction with matter is addressed in photon transport. There are two types of interactions: interaction of photons with atoms (photoatomic reaction) and that with nuclei (photonuclear

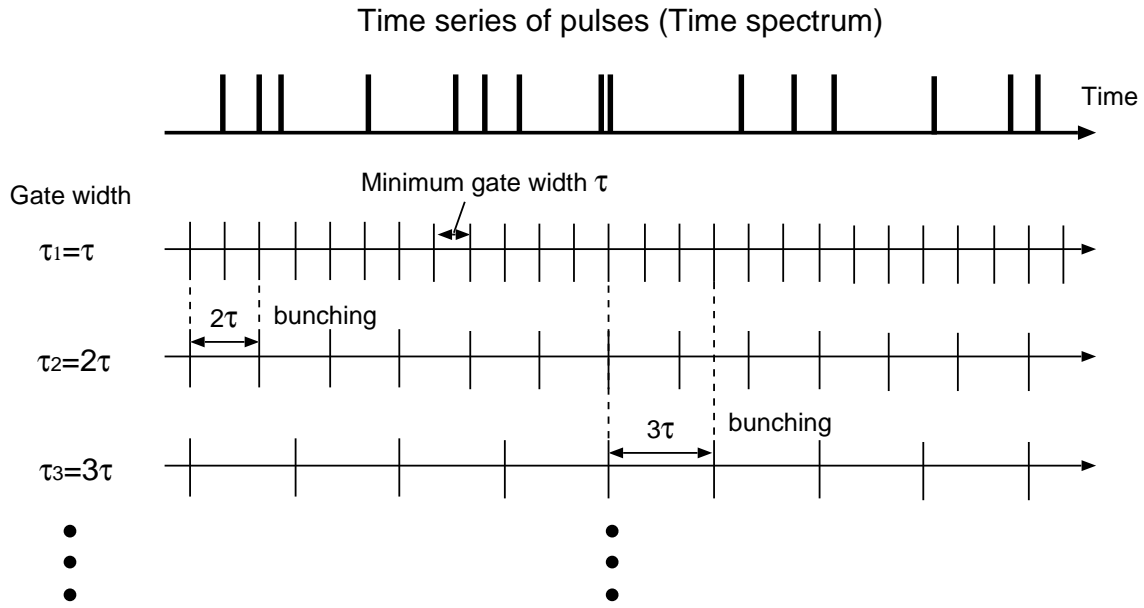


Fig. 4.4 Time spectra for different gate widths

reaction). The energy range from 10 MeV to 20 MeV is favorable for photonuclear reactions and the photonuclear cross sections are considerably smaller than the photoatomic ones. The photonuclear reactions are, thus, not necessary for reactor analysis but there are some cases where the photonuclear reactions are addressed such as non-destructive assay with photonuclear reactions. MVP does not consider the photonuclear reactions by default but they are considered if the PHOTO-NUCLEAR option is specified and the photonuclear data library is available.

The macroscopic total cross section is required to determine the flight length of photons through a material. The cross section is calculated as the sum of the photoatomic and photonuclear cross sections.

$$\Sigma_t(E) = \Sigma_t^{pa}(E) + \Sigma_t^{pn}(E), \quad (4.56)$$

where $\Sigma_t^{pa}(E)$ and $\Sigma_t^{pn}(E)$ are the photoatomic and photonuclear macroscopic total cross sections, respectively. After a photon collides with a material and the collision nuclide is determined with a random number, MVP determines which reaction occurs. MVP selects the photoatomic reaction with the probability of $p = \sigma_t^{pa} / (\sigma_t^{pa} + \sigma_t^{pn})$ and the photonuclear reaction with $1 - p$. σ_t^{pa} and σ_t^{pn} are the photoatomic and photonuclear microscopic total cross sections, respectively.

The photoatomic cross sections are given for each reaction and each energy point as well as the neutron ones. The energy range is from 1 keV to 100 MeV. The cross sections at an arbitrary energy point are obtained by the log-log interpolation.

As is well known, photoatomic reactions are classified into the photo-electric effect, coherent (Thomson) scattering, incoherent (Compton) scattering and the electron pair production. Thus the total cross section of photons is obtained by the sum of each reaction cross section as follows:

$$\sigma_t^{pa}(E) = \sigma_{pe}(E) + \sigma_{coh}(E) + \sigma_{inc}(E) + \sigma_{pp}(E) \quad (4.57)$$

The photonuclear cross sections are also given for each reaction and each energy point. The energy range depends on nuclides and evaluation institutes. See Section G.3 for the details. The total cross section σ_t^{pn} for photonuclear reactions is the same as the sum of the non-elastic cross sections (MT=3).

4.6.1 Detailed and Simple Models

The user can select either the detailed or simple model for the treatment of the photoatomic reactions in MVP. The latter is the model for high energy photons, where the generation of fluorescent X-rays accompanying the photo-electric effect, the coherent scattering and the correction factor of the Klein-Nishina differential cross section for the incoherent scattering are neglected. Therefore, the former model is preferable for low-energy photons or in the case where high accuracy is required for large Z-number nuclides or deep-penetration problems. In both models, Bremsstrahlung photons by slowing down of positrons and electrons by the pair production, Compton recoil electrons and photoelectrons etc. can be optionally taken into account.

4.6.2 Photoelectric Effect

The photoelectric effect is treated as the non-analog absorption in the simple model. Thus the weight of photons is multiplied by $(1 - \sigma_{pe}/\sigma_t)$ in each collision. In the detailed model, the history of the photon is terminated when this reaction is selected. Instead, not more than one fluorescent X-ray is emitted isotropically in the laboratory system depending on the incident photon energy.

4.6.3 Electron Pair Production

This is the reaction of which threshold value is 1.022 MeV. A photon annihilates and a photon of 0.511 MeV with double the weight is emitted isotropically in the laboratory system at the annihilation point.

4.6.4 Incoherent (Compton) Scattering

The differential cross section of the incoherent (Compton) scattering for the photon of the energy E' is given by the following equation. The unit of the photon energy is the electron mass ($m_0c^2 = 0.511$ MeV).

$$\frac{d\sigma_{inc}(\mu, E; E')}{d\mu} = \frac{d\sigma_c(\mu, E; E')}{d\mu} \times S(q, Z) \quad (4.58)$$

$$q = E' \sqrt{1 + \left(\frac{E}{E'}\right)^2 - 2\mu \left(\frac{E}{E'}\right)}, \quad (4.59)$$

where

μ, E	: cosine of the scattering angle in the laboratory system and the energy after scattering
$d\sigma_c/d\mu$: Klein-Nishina differential cross section
$S(q, Z)$: incoherent scattering function ($0 \leq S \leq Z$)
q	: momentum of the recoil electron
Z	: atomic number

The cosine of the scattering angle μ and the energy after scattering E is determined by sampling from the above equation. First, they are sampled from the Klein-Nishina differential cross section and then the rejection method with the scattering function S is applied for the results. Since the correction for the scattering function is not taken into account in the simple model, the rejection method is not applied and the sampled results from the Klein-Nishina differential cross section is used.

The angular distribution of the scattered photon is given by the Klein-Nishina formula:

$$p(\mu; E') d\mu = \frac{\pi r_0^2}{\sigma_c(E')} \frac{1}{\{1 + E'(1 - \mu)\}^2} \left[E'(1 - \mu) + \mu^2 + \frac{1}{1 + E'(1 - \mu)} \right] d\mu, \quad (4.60)$$

where r_0 is the classical radius of the electron (2.817938×10^{-13} cm).

The distribution of the azimuthal angle is uniform and the energy of the scattered photon E is derived from the conservation laws of energy and momentum:

$$E = \frac{E'}{1 + E'(1 - \mu)}. \quad (4.61)$$

Using this equation, Eq. (4.60) can be rewritten as a function of energy after scattering:

$$p(E; E') d\mu = \frac{\pi r_0^2}{\sigma_c(E')} \frac{1}{E'^2} \left[\frac{E}{E'} + \left(\frac{2}{E'} + \frac{1}{E'^2} \right) + \left(E' - \frac{2}{E'} - 2 \right) \frac{1}{E} + \frac{1}{E^2} \right] dE, \quad (4.62)$$

$$\frac{E'}{1 + 2E'} \leq E \leq E'.$$

The cosine of the scattering angle is as follows:

$$\mu = 1 + \frac{1}{E'} - \frac{1}{E}. \quad (4.63)$$

In MVP, Koblinger's direct sampling method⁴⁸⁾ is used in the energy region of $E' \geq 1.5$ MeV and the non-uniform rejection method by Horowitz et al.⁴⁹⁾ is used below 1.5 MeV.

4.6.5 Coherent (Thomson) Scattering

Photons do not lose their energy but only the flight direction is changed in this scattering. Since the distribution of the azimuthal angle is uniform, the scattering angle is determined from the following angular distribution:

$$\frac{d\sigma_{\text{coh}}(\mu; E')}{d\mu} = \pi r_0^2 (1 + \mu)^2 [C(q, E', Z)]^2 \quad (4.64)$$

$$[C(q, E', Z)]^2 = [F(q, Z) + F_1(E')]^2 + F_2(E')^2, \quad (4.65)$$

where

$C(q, E', Z)$: correction factor of the differential cross section independent of the energy
$F(q, Z)$: form factor ($0 \leq F \leq Z$)
$F_1(E')$: real anomalous scattering factor
$F_2(E')$: imaginary anomalous scattering factor
q	: momentum of the recoiled atom ($q = E' \sqrt{2(1 - \mu)}$)

This scattering is not taken into account in the simple model.

4.6.6 Bremsstrahlung Photon

The photoatomic interaction yields electron-positron pairs, Compton recoil electrons or photo-electrons (Auger electrons). The Bremsstrahlung photons are generated when the electrons slow down in the electric field of nuclei. But the generation point, energy, etc. of the photons cannot be determined precisely without the transport calculation of the electrons and positrons. MVP uses the thick target approximation where the photons are generated at the same points as the electrons and positrons without the transport of them. The used data and model are the same as in the MCNP code³⁴⁾ and so on.

4.6.7 Analysis of Photonuclear Reactions

If a photonuclear reaction is selected, the type of the photonuclear reaction (MT number) is determined with each cross section and a random number. MVP determines the type of generated particles and the number of them based on the determined MT number, and then determines the energy and angle of the particles from the energy-angle distribution. The number of generated particles depends on the yield and the energy-angle distribution depends on the type of the distribution (for cases of LF=1, 5, 7, 9, 11, 61, 62, 66, 67. LF is the parameter specified in the MVP library). If the unknown distribution (LF=60) or the distribution for two-body recoils (LF=64) is used, no secondary particles are generated. MVP treats the fission reaction (MT=18) as a neutron generation process like other inelastic scattering processes, unlike the absorption process for neutron eigenvalue calculations.

If the discrete-level inelastic scattering representation (MT=50–90) is used, MVP samples a scattering angle from a specified angular distribution and determines an energy with the two-body kinematics. The photoneutron energy E_n in the center-of-mass (CM) system is obtained with the following equation:⁵⁰⁾

$$E_n = \frac{A-1}{A} \left[E_\gamma - Q - \frac{E_\gamma^2}{2m_n c^2 (A-1)} + \frac{E_\gamma}{A} \sqrt{\frac{2(A-1)}{m_n c^2 A (E_\gamma - Q)}} \cos \theta \right], \quad (4.66)$$

where each parameter is defined as follows.

- A : mass of the target nucleus,
- E_γ : photon energy,
- Q : threshold energy,
- m_n : neutron mass,
- c : speed of light,
- θ : angle between photon and neutron flight directions.

Only the CNDC evaluation uses the discrete-level inelastic scattering representation for the (γ, n') reaction and the angular distribution is given as isotropic. It should be noted that the above equation holds only for an outgoing neutron.

There are some cases where the energy-angle distribution of LAW=1 for MF=6/MT=5 is used, a small cross section of ~ 0.01 mb is given from the threshold energy to 20 MeV, and its distribution has the range from 0.0 eV to 1.0×10^{-5} eV (for example, the LANL evaluation for O-16). In this case, the sampled neutron energy would be like 6.0×10^{-6} eV in the CM system and become an energy of the keV order in the laboratory system. Currently, MVP allows this conversion.

Only neutrons and photons are generated from the photonuclear reactions in the current version of MVP. If production data for other charged particles such as protons, alphas, etc. are given, MVP calculates the production number of the charged particles but does not generate the particles. MVP addresses

the photonuclear reactions whose MT numbers are not more than 117 ($MT \leq 117$) and neglects the ones for MT numbers of more than 117. It should be noted that MVP forcibly terminates the calculation if the sum of the numbers of generated neutrons, generated photons and particles in transport becomes larger than the bank size (NBANK).

4.6.8 Speed of Photons

The speed of photons is required for time-dependent tallies. Since the speed V_p of photons is independent of its energy or energy group, it is given as

$$V_p = c \quad (4.67)$$

for both MVP and GMVP.

5. MVP/GMVP Geometry

The features of the MVP/GMVP geometry are as follows.

- (1) MVP/GMVP employs the CG (Combinatorial Geometry)¹⁸⁾ that has been used in the MORSE-CG and VIM codes and thus the user can model the complicated 3-dimensional geometry easily.
- (2) MVP/GMVP has a LATTICE option with rectangular parallelepiped and right hexagonal prism lattices to describe easily a repeated geometry with a hierarchical structure often seen in reactor cores and so on.
- (3) In the LATTICE geometry, the user can distinguish partial spaces in the LATTICE filled with the same unit cell by naming the space (SUBSPACE name).
- (4) Physical quantities such as fluxes etc. can be tallied in each partial region and also tallied in arbitrary combination of the partial regions. (Specification of TALLY REGIONS.)
- (5) The statistical geometry model is available to simulate the random distribution of coated fuel particles, pebble bed fuels, plutonium spots, etc.

In this chapter, the basic concept is described for each feature.

5.1 Combinatorial Geometry

In combinatorial geometry, the geometry model is described by the concepts of “BODY”, “ZONE” and “REGION”.

(1) BODY

BODY is the object of a unit shape such as a rectangular parallelepiped, a sphere, etc. A lot of objects are prepared to describe the calculation geometry. BODIES are defined by one curved surface or more, and the interior and exterior of the BODIES are clearly distinguished.

(2) ZONE

ZONE is the minimum unit of the geometry models in MVP/GMVP and is defined by the combination of BODIES (set operations of union, intersection, etc.). A ZONE consists of a single material in which the composition and density are uniform. A ZONE belongs to only one REGION.

(3) REGION

REGION is the unit of a spatial region where fluxes etc. are tallied and is defined by the union of some ZONES. A REGION can include more than one ZONE each of which has a different material. The parameters for the variance reduction are set by the REGION.

5.2 BODIES

ZONES are defined by the combination (set operations) of BODIES. The interior and exterior of the BODIES are determined uniquely. In the current MVP/GMVP, there are following 17 BODIES:

- RPP (rectangular parallelepiped)
- SPH (sphere)
- RCC (right circular cylinder)
- CYL (cylinder of which central axis is parallel to the z-axis)
- TRC (truncated right cone)
- TEC (truncated elliptic cone)
- WED or RAW (wedge)
- BOX (parallelepiped)
- RHP (right hexagonal prism of which central axis is parallel to the z-axis and of which one side is perpendicular to the x-axis)
- HEX (hexagonal prism)
- ARB (arbitrary polyhedron with 4, 5 or 6 faces)
- ELL (ellipsoid by rotation)
- GEL (general ellipsoid)
- ELT (elliptical torus)
- HAF (half space with a planer surface)
- RCL (right cylinder with a specified radial direction)
- GQS (general quadratic surface).

These BODIEs are distinguished by the above 3-character string and the size and position of each BODY are defined by the input data. In the input data, each BODY has a BODY ID and the IDs are used to specify the BODIEs to define ZONEs. The detail of the BODY data is described in Section 10.2.

As a special usage of BODIEs, the combination of BODIEs can be treated as a BODY. This can be specified by the BODY type BBC (BODY-by-BODY combination). There are some cases where BBC makes it easy to describe the combination of BODIEs.

5.3 Definition of ZONEs

5.3.1 Operation of BODY Combination

The ZONE is the minimum unit of the geometry models and is defined by the combination of BODIEs (set operations). The available set operations are following two operators,

- intersection (AND): No operator is necessary.
- union (OR) : The operator is “OR”.

and the signs with the BODY ID. The signs mean

- “+” : inside of a BODY (It can be omitted.)
- “-” : outside of a BODY.

There are no restrictions for the number of BODIES to define a ZONE. But there are some cases where computation time becomes large if a number of BODIES are used to define a ZONE.

Example:

“+1 +2” : “inside of BODY 1” and “inside of BODY 2”

“+1 -2” : “inside of BODY 1” and “outside of BODY 2”

“-1 +2” : “outside of BODY 1” and “inside of BODY 2”

“+1 OR +2” : “inside of BODY 1” or “inside of BODY 2”

“-1 -2 +3” : “outside of BODY 1” and “outside of BODY 2” and “inside of BODY 3”

Figure 5.1 shows these set operations to define a ZONE.

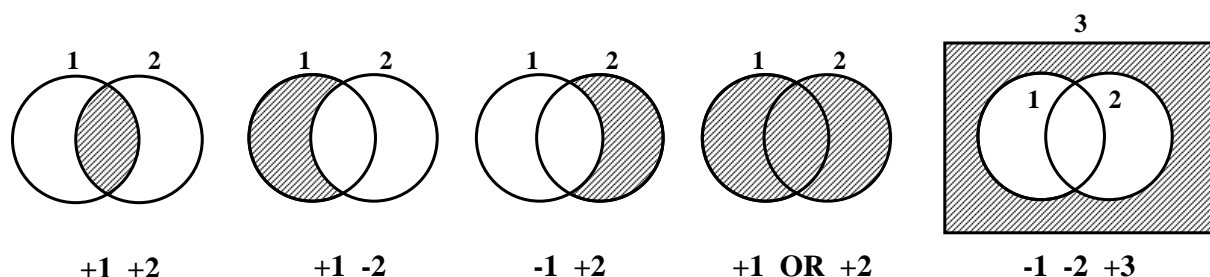


Fig. 5.1 ZONE definition by combination of BODIES

5.3.2 Unsupported BODY Operation

“AND” has priority over “OR” in the operation order in the combinatorial geometry of the current MVP/GMVP code. Even if the “OR” operator should be performed in prior to “AND”, parentheses “()” cannot be used. For example, the expression

+1 (+2 OR +3) “inside of BODY 1” and “inside of BODY 2 or BODY 3”

is not allowed to define the ZONE specified in the shaded portion in Fig 5.2. In this case, the expression

+1 +2 OR +1 +3 “inside of BODIES 1 and 2” or “inside of BODIES 1 and 3”:

must be used.

5.3.3 OR at the Head of BODY Operations

If an “OR” operator is used, it can be located at the top of BODY operations. For example, the expression “OR +1 OR +2 OR +3” is equivalent to “+1 OR +2 OR +3”. Both expressions indicate the shaded region (inside of BODY 1 or 2 or 3) in Fig 5.3. It seems that the expression “+1 OR +2 OR +3” is more natural but “OR +1 OR +2 OR +3” is used in the MORSE code etc. Thus both expressions are allowed because of the compatibility of the input data.

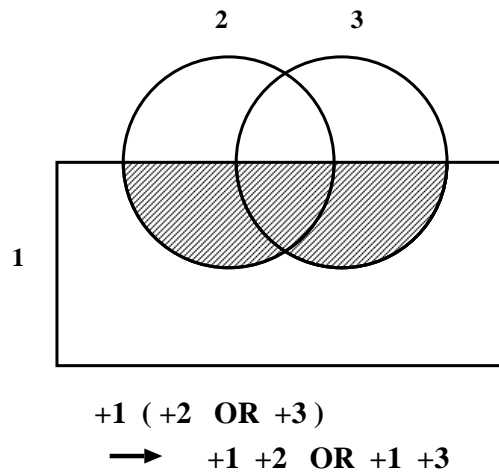


Fig. 5.2 Unsupported parentheses and alternative expression

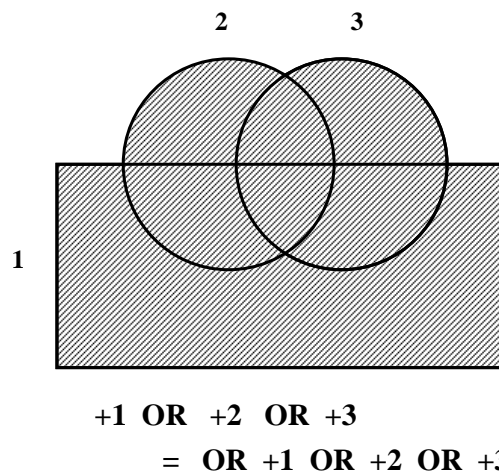


Fig. 5.3 Equivalent OR usage

5.3.4 INPUT-ZONE and ZONE

To be precise, a partial space (ZONE) defined by a set of BODY operations is an “INPUT-ZONE” inside the MVP/GMVP code. The INPUT-ZONE is a unit specified by the material ID and the REGION name in the input data.

When “OR” operators are included in the definition of an INPUT-ZONE, parts of the INPUT-ZONE divided by the “OR” operators are defined as different ZONES in the code. Unless an “OR” operator is included, the INPUT-ZONE is a ZONE itself. In general, operations in the codes are performed by the ZONE unit. Thus the user should note that numbers in messages output during calculations are sometimes the ZONE identification numbers, not the “INPUT-ZONE” identification numbers.

5.3.5 Multiply Defined ZONE

If a partial space is included in more than one ZONE, it is called a “multiply defined ZONE”. Particles in the multiply defined ZONE are judged to be in different ZONES (one of the ZONES in the multiply defined ZONE) depending on their history. For example, suppose that two ZONES are defined as the insides of BODY #1 and #2, respectively, and the shaded portion is multiply defined as shown in Fig 5.4. Considering the path between A and B, the bold line in the figure is judged as follows:

- 1) When $A \rightarrow B$: The particle belongs to ZONE #1.
- 2) When $B \rightarrow A$: The particle belongs to ZONE #2.

Thus the particle belongs to the different ZONES depending on the path. In the case where a particle enters the multiply defined ZONE from other region than the two ZONES, it is more complicated to judge which ZONE the particle belongs to. It depends on the definition order of the ZONES and the histories of some particles.

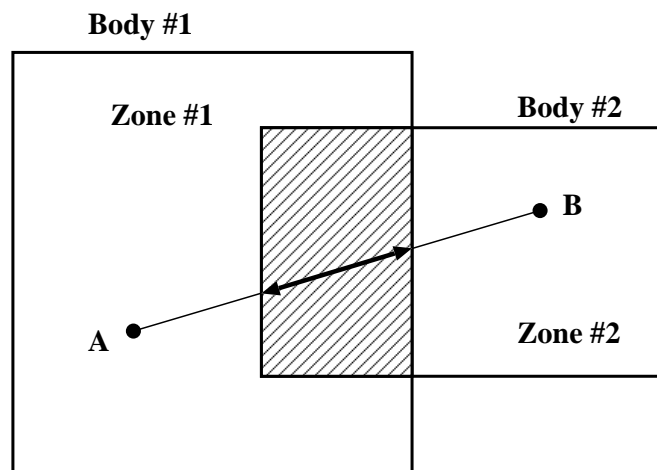


Fig. 5.4 Multiply defined ZONE

If the multiply defined ZONE satisfies the conditions:

- the material of the ZONE is the same for the multiple definitions,
- REGION to which the ZONE belongs is the same for the multiple definitions,

then there are no problems in the calculation because the material of the spatial region and the REGION are defined uniquely. For example, we consider the geometry shown in Fig 5.4. The INPUT-ZONE “+1 OR +2” defined by the overlapped BODIES #1 and #2 is divided into two different ZONES (inside of BODY #1 and inside of BODY #2) and the overlapped part belongs to both ZONES. Then, the material and REGION are given to the INPUT-ZONE and thus the two ZONES satisfy the above conditions.

On the other hand, if multiply defined regions exist and they do not satisfy the above conditions, the geometry model is not determined uniquely and is meaningless. Even in this case, the MVP/GMVP code does not check the multiply defined regions and the user should pay attention to it. But the user can check whether or not multiply defined regions satisfy the above conditions by using the option of MVP/GMVP to print the cross-sectional view or the geometry checking tool CGVIEW (Appendix B).

5.3.6 Identification of ZONES and Undefined Regions

All partial spaces where particles may exist must be defined as ZONES. Particles that enter undefined regions cannot be tracked and the histories are terminated as “lost particles” because ZONES to which the particles belong cannot be determined. In MVP/GMVP, when a particle reaches the boundary of a spatial region (spatial coordinates \mathbf{X} , flight direction $\mathbf{\Omega}$), the code finds a ZONE including the spatial position

$$\mathbf{X}' = \mathbf{X} + \epsilon \mathbf{\Omega},$$

where ϵ is the minimum distance that the code recognizes (input data DEPS : default value = 10^{-5} cm) and $\mathbf{\Omega}$ the unit vector of the particle direction. ZONES are searched in the following order:

- in the order of ZONES to be entered with possibilities estimated from the previous tracking,
- in the input order of ZONES except for the above ZONES.

There are some cases where an undefined region may be unexpectedly created when the boundary surface of the adjacent two ZONES is defined by each BODY of the ZONE but the surface is not always identical with each other because of the lack of numerical accuracy. There are other cases where the code fails to track particles as if undefined regions exist because of the numerical error. The histories of these particles are terminated and are treated as “lost particles”. These problems can be avoided by defining the region in the vicinity of the boundary multiply to the extent where it is physically insignificant.

5.3.7 Finiteness of ZONES

The spatial ZONES where particles exist must be finite, not infinite. Even if only the energy change of particles is calculated in the infinite geometry, the finite ZONES with the perfect reflective boundary condition must be defined. This is because the histories of particles are terminated when the distance between the current position and the ZONE boundary cannot be calculated. On the other hand, ZONES that particles never enter can be infinite, as ZONES including the fictitious material to indicate boundary conditions described below. Such ZONES can be defined as the outside of a single BODY.

5.4 Specification of Materials and Boundary Conditions

The material of each ZONE is specified by the material ID defined in the CROSS SECTION data block of input data. In MVP/GMVP, the boundary conditions are defined by fictitious ZONES including fictitious materials that are located in the outside of the boundaries. The material IDs including the fictitious materials are listed in Table 5.1.

There are some restrictions for the use of the fictitious materials. The user should pay attention to the following.

- (1) The fictitious materials with their ID's ≤ -1001 cannot be used for calculations with point detectors. Regions of internal or outer void can be specified for the calculations. MVP/GMVP cannot calculate the flight path of pseudo-particles correctly when the fictitious materials exist in the calculation geometry.
- (2) The fictitious materials cannot be used for the cases where particles possibly enter the endless loop. One example is the case where the outside of a region is filled with the perfect reflective material and the inside is vacuum. In this case, some particles may continue flying without absorption or scattering.

Table 5.1 Material IDs

Material ID	Corresponding material
> 0	Normal material of which composition is given in the CROSS SECTION data block.
0	Internal void. (Particles pass through this region without collisions.)
−1 to −998	ZONE filled with a small “universe” defined separately (a CELL or a LATTICE comprised of regularly arranged CELLS). This ZONE is called a “FRAME”. The absolute value is the ID number of a small universe (usually LATTICE).
−999	Fictitious ZONE that defines the outside of CELLS (unit of repeated geometry).
−1000	Outer void. (It corresponds to the vacuum boundary condition. Particles that enter the ZONES are regarded as leaked from the system and the random walk is terminated.)
−1001 to −1999	Albedo reflective material. (currently not available.)
−2000	Perfect reflective material. (It corresponds to the mirror reflective boundary condition.)
−3000	Isotropic reflective material. (It corresponds to the white reflective boundary condition.)
−4000	Material to specify the periodic boundary condition.

There are additional restrictions for the material of the periodic boundary condition. A ZONE with the fictitious material ID of −4000 must be defined for the periodic boundary condition but the ZONE must satisfy the following conditions.

- (1) The definition of the ZONE filled with the fictitious material −4000 must include the outside of a BODY that has a pair of parallel planes such as RPP, BOX, RHP etc. That is, the pair of parallel planes is the set of periodic boundaries.
- (2) The ZONE with the material ID of −4000 must be defined by a single ZONE. The outside of the parallel planes that are periodic boundaries must not be defined by separate ZONES.

In the calculation process, when a particle reaches the boundary of the ZONE of which material ID is −4000, the spatial position of the particle is displaced to the corresponding position on the other plane of the pair with the same flight direction.

5.5 Repeated Geometry

5.5.1 Concept of LATTICE Geometry

MVP/GMVP has a capability to represent repeated geometry easily. It is called LATTICE geometry in the MVP/GMVP terminology. Figure 5.5 shows the concept of the LATTICE geometry. The following terms are used in the LATTICE geometry. (These specific terms are written in upper case.)

- CELL : Unit of the repeated geometry. The inside of the CELL is defined in the CELL-specific coordinates.

- LATTICE : Three-dimensional regular arrangement of CELLS and the frame of the LATTICE (LATTICE FRAME).
- FRAME ZONE : ZONE where a LATTICE is placed in the geometry model.
- SUBFRAME : Partial space where each CELL is placed in a LATTICE.

The terms “LATTICE — FRAME ZONE” and “CELL — SUBFRAME” may make the user confused. “LATTICE” and “CELL” are “substances” that give the arrangement of CELLS and the unit of repeated geometry. On the other hand, “FRAME ZONE” and “SUBFRAME” are locations where “LATTICE” and “CELL” are placed.

The LATTICE geometry of MVP/GMVP has the following functions. The functions can be used with the LATTICE option. (The statistical geometry model is treated as one of the LATTICES in programming but it is explained in the next section.)

- (1) A rectangular parallelepiped lattice comprised of rectangular parallelepiped CELLS and a right hexagonal prism lattice of right hexagonal prism CELLS are available as a LATTICE.
- (2) Any number of LATTICES can be defined and placed anywhere in space with any direction of them. The same LATTICE can be placed at different positions.
- (3) A LATTICE has a frame that defines the boundary of the LATTICE (LATTICE FRAME). The FREE-LATTICE-FRAME option enables the user to define any shape of the LATTICE FRAME. Without this option, only limited LATTICE FRAMES can be used. The frame of a rectangular parallelepiped LATTICE must be the rectangular parallelepiped of the outermost surface of the CELL arrangement. A right hexagonal prism, a circular cylinder and rectangular parallelepiped can be used for a right hexagonal prism LATTICE.
- (4) A LATTICE consists of any number of SUBFRAMEs. Some SUBFRAMEs may not be filled with CELLS but they are filled with the ZONEs in the same universe as the FRAME ZONE where the LATTICE is placed.
- (5) CELLS are defined by combinatorial geometry.
- (6) CELLS can include other LATTICES. (This is called the nested LATTICE geometry.)
- (7) CELLS can be placed with the direction reversed or rotated at SUBFRAMEs in the LATTICE.

5.5.2 Hierarchical Structure of Nested LATTICE Geometry

In the LATTICE geometry model of MVP/GMVP, CELLS can include other LATTICES. Figure 5.6 shows the geometry model in this case. The rank of the hierarchical structure is called “level” in MVP/GMVP. The relation of each object in Fig 5.6 is represented as follows.

- : The left side is divided into the right side or is represented by the combination of the right side.
- || : The lower universe is placed in the upper space.
- { : The left side is classified into the right side.

The meaning of terms is shown in the following.

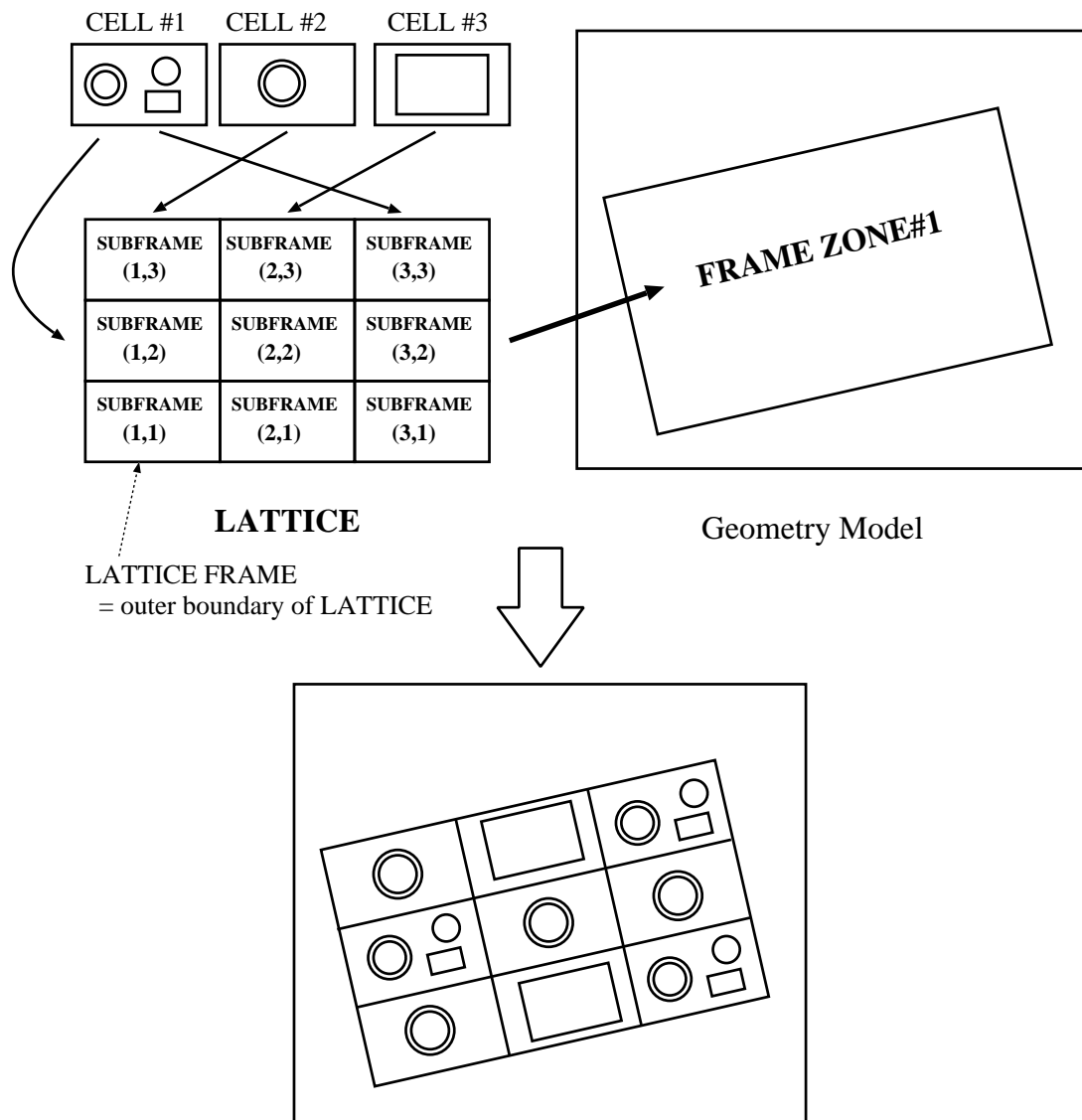


Fig. 5.5 Geometry model defined with LATTICE capability

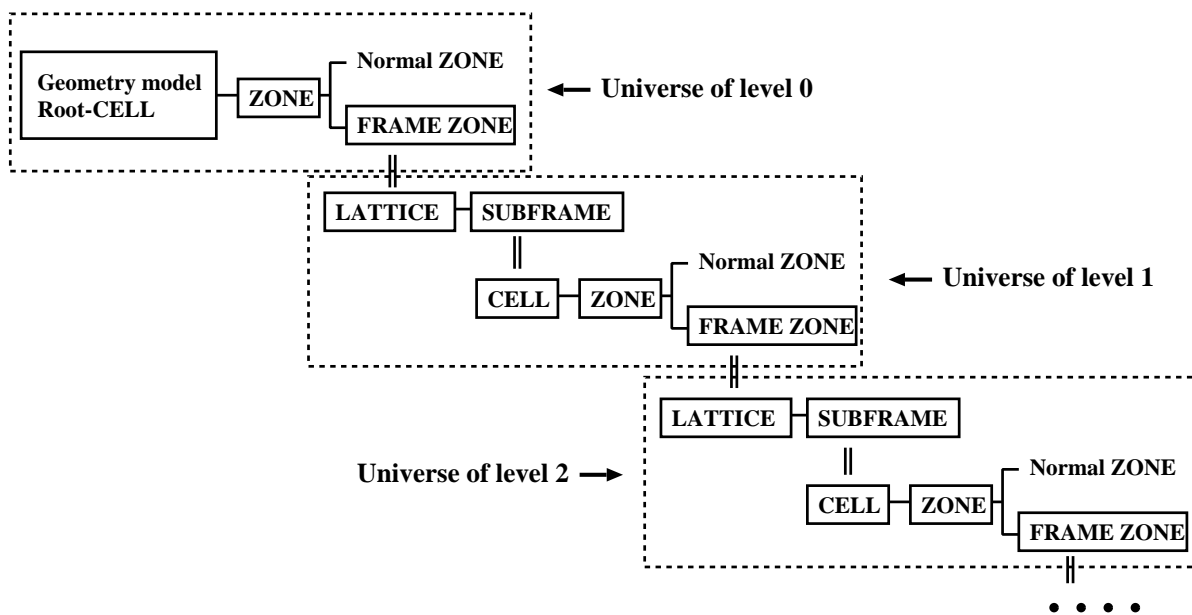


Fig. 5.6 Concept of nested LATTICE geometry

- Normal ZONE

ZONE with a normal material (material ID number ≥ 0) or a fictitious material that defined the boundary condition or the outer boundary of CELLS (material ID number ≤ -999). No LATTICE is defined in the ZONE.

- FRAME ZONE

ZONE where a LATTICE is placed. $-998 \leq \text{material ID number} \leq -1$ (The absolute value of the material ID number is the LATTICE ID number of the LATTICE.). The form of “LAT=LATTICE ID” in the input data is allowed instead of the above material ID number.

- LATTICE

Universe defined by the LATTICE geometry. The space in the universe is divided into SUB-FRAMES.

- SUBFRAME

Partial space where each CELL is arranged.

- CELL

Universe defined in the coordinate system specific to CELLS.

The nested LATTICE geometry is based on the “CELL-ZONE” structure and CELLS can include other CELLS through LATTICES placed at FRAME ZONES. Logically, the infinite level of the hierarchical structure can be described but currently the 4th level (Level 4) or less of the universe can be treated.

[Note: ROOT-CELL]

The whole calculation geometry itself can be placed at other SUBFRAME by modifying the boundary conditions. Therefore, it can be regarded as a CELL in the uppermost level of the current model and is called “ROOT-CELL”. The space where the ROOT-CELL is placed can be regarded as “ROOT-SUBFRAME” (=ROOT-FRAME). This interpretation holds from the viewpoint of the hierarchical structure but actually there is a restriction that only rectangular parallelepiped and right hexagonal prism CELLS are available in the current MVP/GMVP code.

5.5.3 Definition of CELLS

Rectangular parallelepiped or right hexagonal prism CELLS are defined inside the BODY RPP or RHP shown in Fig 5.7 as well as normal ZONES. The BODY RPP is a rectangular parallelepiped and each surface is perpendicular to the coordinate axis. The BODY RHP is a right hexagonal prism that has a central axis parallel to the Z -axis and one side is perpendicular to the X -axis. The BODY to define the LATTICE FRAME (FRAME BODY) and BODIES to define ZONES in the CELLS can be defined at any position. The inside of the FRAME BODY must be defined as ZONES and must not include undefined regions. The outside of the FRAME BODY must be defined by the ZONE with the fictitious material that specifies the outside of CELLS (material ID = -999) as shown in Fig 5.8. The CELLS defined like this are distinguished by CELL ID numbers.

Particles in the CELLS are tracked in the coordinates (X, Y, Z) where the CELLS are defined through the CELL-coordinates (X_c, Y_c, Z_c) , the origin of which is at the center of the CELL shown in Fig 5.7 and the CELL-array coordinates (X_a, Y_a, Z_a) described in Section 5.5.4.

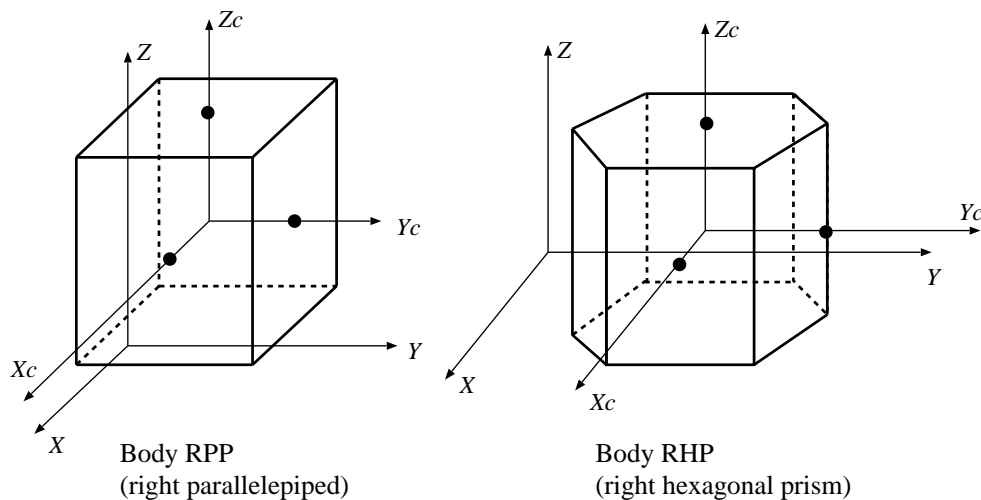


Fig. 5.7 CELL-coordinates (X_c, Y_c, Z_c)

5.5.4 Definition of LATTICES and Coordinate Transformation

LATTICES are defined as follows. (This section describes rectangular parallelepiped and right hexagonal prism LATTICES. The statistical geometry model is described in another section.)

(1) Specification of the LATTICE type

The following 4 types of LATTICES are available. The LATTICE type ID is specified by the input data LTYP.

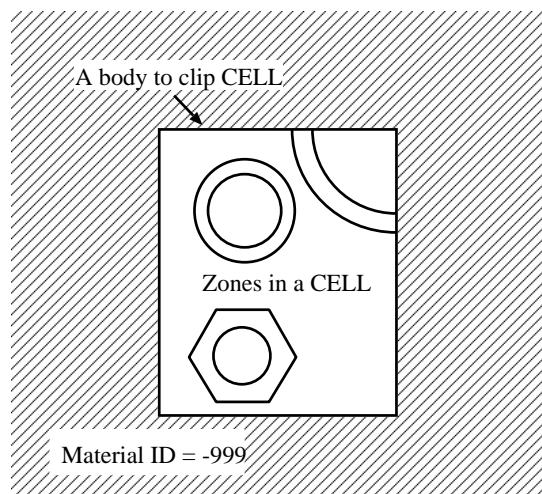


Fig. 5.8 Definition of CELL

- LATTICE type 1 : Rectangular parallelepiped LATTICE with a rectangular parallelepiped FRAME
- LATTICE type 2 : Right hexagonal prism LATTICE with a right hexagonal prism FRAME
- LATTICE type 3 : Right hexagonal prism LATTICE with a right circular cylinder FRAME
- LATTICE type 4 : Right hexagonal prism LATTICE with a rectangular parallelepiped FRAME

(2) Size of CELL array and CELL-array coordinates

Figure 5.9 shows the examples (cross-sectional views) of the CELL arrays for rectangular parallelepiped and right hexagonal prism LATTICES. The CELL-array coordinates (X_a, Y_a, Z_a) are defined for the CELL arrays as shown in Fig 5.9. The Z_a -axis is perpendicular to the figures with the upward direction and the origin is at the bottom layer ($k = 1$) of SUBFRAMEs. Each SUBFRAME in the CELL array is indicated by 3 indices (i, j, k) . The size of the CELL array is determined by the number of SUBFRAMEs arranged in each direction (variable NVLAT). In the examples of Fig 5.9, the size of the CELL arrays is 4 in the X_a -direction and 3 in the Y_a -direction for both cases.

(3) Size of SUBFRAMEs (variable SZLAT)

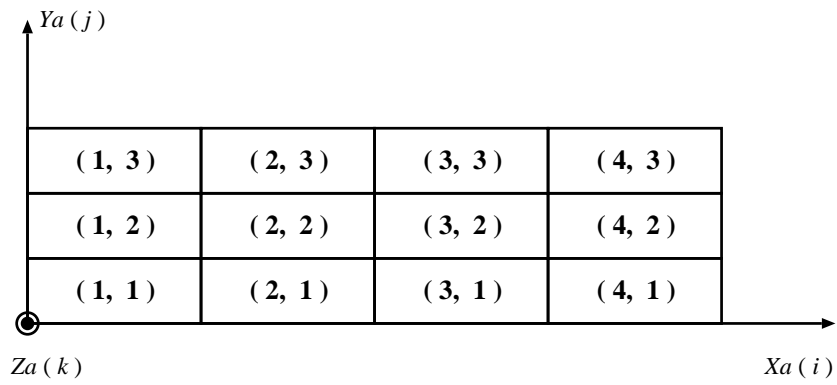
The size of SUBFRAMEs that comprise a single LATTICE is constant and defined as follows.

- Rectangular parallelepiped LATTICE : Pitches in the X_a -, Y_a - and Z_a -axis.
- Right hexagonal prism LATTICE : Width and height (Pitches) in the X_a - and Z_a -axis, respectively.

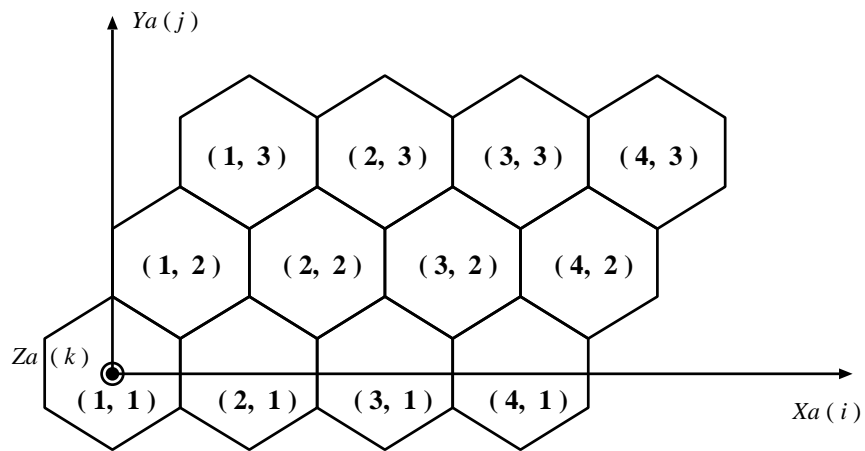
(4) Relation between CELL-array and CELL coordinates

The size of SUBFRAMEs is used for the coordinate transformation. Basically the direction of each coordinate axis is related as

$$X_a \parallel X_c, Y_a \parallel Y_c, Z_a \parallel Z_c.$$



(a) Rectangular lattice



(b) Hexagonal lattice

Fig. 5.9 CELL array and CELL-array coordinates (X_a, Y_a, Z_a)

CELLs can be placed with the direction reversed in each axis. In case of rectangular LATTICEs, the coordinates of X_a and X_c are related by the following equations (Note that the origin of CELL-coordinates is the center of the CELL.):

$$i = \text{INT}(X_a/P_x) + 1 \quad (5.1)$$

$$X_c = \{X_a - P_x \times (i - 1/2)\} \times f \quad (5.2)$$

$$X_a = f \times X_c + P_x \times (i - 1/2), \quad (5.3)$$

where

P_x : Pitch in the X_a -axis,

$\text{INT}(x)$: Maximum integer less than or equal to x ,

f : Sign of reverse allocation of CELLs in the SUBFRAMEs,

1 : Not reversed,

-1 : Reversed.

Likewise other 2 coordinates are related. A little complicated but similar relation is used for the right hexagonal prism LATTICE.

(5) Size of SUBFRAMEs and CELLs

The size of SUBFRAMEs must be the same as that of CELLs to be placed in the SUBFRAMEs. If the former is larger than the latter, there exist undefined regions. Otherwise, there exist multiply defined regions.

(6) LATTICE FRAME and LATTICE coordinates (X_ℓ, Y_ℓ, Z_ℓ)

- Rectangular parallelepiped LATTICE : The minimum size of the rectangular parallelepiped that includes all SUBFRAMEs is the LATTICE FRAME. It is uniquely determined by the size of the CELL array and SUBFRAMEs. The direction of each axis of the LATTICE coordinates (X_ℓ, Y_ℓ, Z_ℓ) is the same as that of the CELL-array coordinates.
- Right hexagonal prism LATTICE : Three kinds of shapes (right hexagonal prism, circular cylinder and rectangular parallelepiped) are available for the LATTICE FRAME. The LATTICE coordinates (X_ℓ, Y_ℓ, Z_ℓ) are defined for each FRAME as shown in Fig 5.10. The size of the FRAME is specified by the variable SZHEX. The FRAME (LATTICE coordinates) and the CELL arrangement (CELL-array coordinates) can be defined independently and the relation between them is determined as follows.

1) The origin is defined so that the direction of the Z_a -axis is the same as that of the Z_ℓ -axis and the plane of $Z_a = 0$ coincides with the plane of $Z_\ell = 0$. The relation of (X_a, Y_a) and (X_ℓ, Y_ℓ) is determined from the following 2) and 3) as shown in Fig 5.11.

2) Distance between the center of the LATTICE FRAME and that of the reference SUBFRAME:

This is specified by the vector from the center of the FRAME (■ in Fig 5.11) to that of the reference SUBFRAME (□ in Fig 5.11) : (components in the (X_ℓ, Y_ℓ) coordinates : variable SZHEX).

Center of the FRAME	:	Center of the base of the FRAME (regular hexagon, circle, rectangle)
Reference SUBFRAME	:	SUBFRAME indicated by indices (i, j) (variable RCELL, RCELL(5 4) in Fig 5.11)
Center of the reference SUBFRAME	:	Center of the regular hexagonal base

- 3) θ : Rotation angle of (X_a, Y_a) from (X_ℓ, Y_ℓ) in the counter-clockwise direction.

The rotation angle is -30 degree ($\theta = -30$) for the example in Fig. 5.11, which is often seen in fast breeder reactors. But $\theta + 30$ must be input for the variable SZLAT(2) so that the arrangement of the example is the standard ($\theta = 0$).

Note 1 Only the inside of the LATTICE FRAME is defined as the LATTICE in the CELL array. That is, particles that get out of the FRAME are judged to be out of the LATTICE. The inside of the LATTICE FRAME must be filled with the CELL array. Otherwise, there exist undefined regions inside the FRAME.

Note 2 The universe in a LATTICE (CELL-array coordinates) is related to the universe out of the LATTICE through the LATTICE coordinates. (See Allocation of LATTICE at FRAME ZONE (p. 66).)

5.5.5 Specification of CELLS Placed at SUBFRAMES

CELLs placed at SUBFRAMES are specified by the ID numbers (variable KLATT) and the directions of the CELLS are also specified (variable KSLAT).

The SUBFRAMES of KLATT=0 are not included in the LATTICE and the spatial regions of the SUBFRAMES are defined by other ZONES in the same level as the LATTICE FRAME. It enables us to describe the geometry comprised of 2 LATTICES with different pitches as shown in Fig 5.12 . For simplicity, we consider only the 2-dimensional array.

LATTICE #1: Rectangular parallelepiped LATTICE of 5×4 of which SUBFRAMES of 3×2 are not included in the LATTICE.

```
KLATT(  1  1  1  1  1
        1  0  0  0  1
        1  0  0  0  1
        1  1  1  1  1 )
```

LATTICE #2: Rectangular parallelepiped LATTICE of 4×3

FRAME ZONE where LATTICE #1 is allocated = +1 -2

FRAME ZONE where LATTICE #2 is allocated = +2

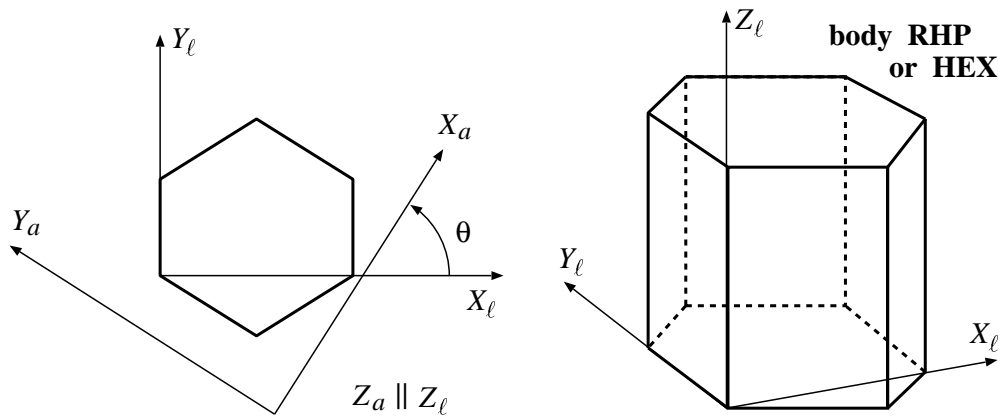
When a CELL is placed at a SUBFRAME, the direction of the CELL is determined so that the axes of the cell coordinates satisfy

$$X_a \parallel X_c, Y_a \parallel Y_c, Z_a \parallel Z_c$$

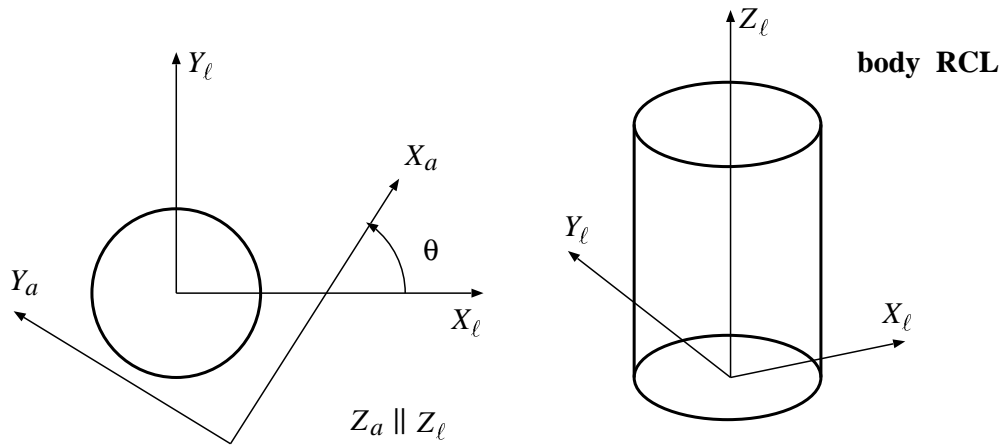
by default. The following transformation can be applied.

- Rectangular parallelepiped LATTICE : Reverse in the X_a -, Y_a - and Z_a -directions
- Right hexagonal prism LATTICE : (x) Reverse in the X_a -direction.
(t) Rotation around the Z_a -axis (central axis of right hexagonal prism CELLS) by 60 degrees.
(z) Reverse in the Z_a -direction
(The order of the transformation is (x) \rightarrow (t) \rightarrow (z).)

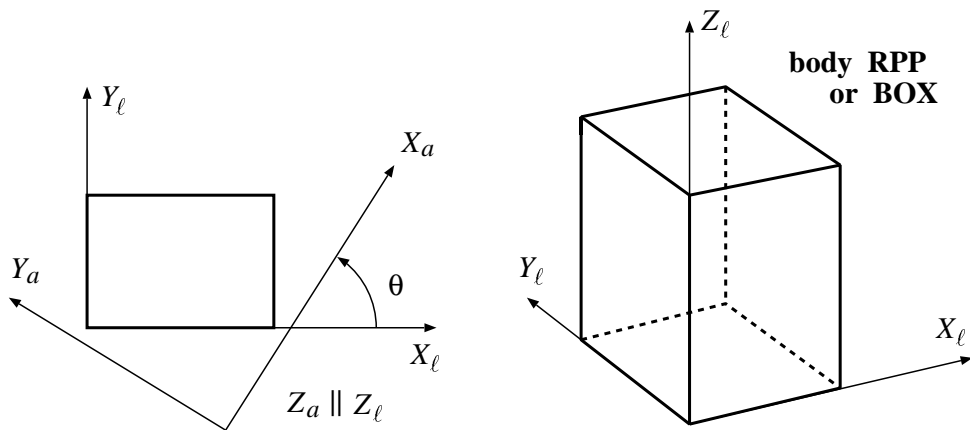
LATTICES defined like this are specified by the LATTICE ID numbers.



(a) Frame is a hexagonal prism (LTYP=2)



(b) Frame is a right cylinder (LTYP=3)



(c) Frame is a right parallelepiped (LTYP=4)

Fig. 5.10 FRAME of a right hexagonal prism LATTICE and LATTICE coordinates (X_ℓ , Y_ℓ , Z_ℓ)

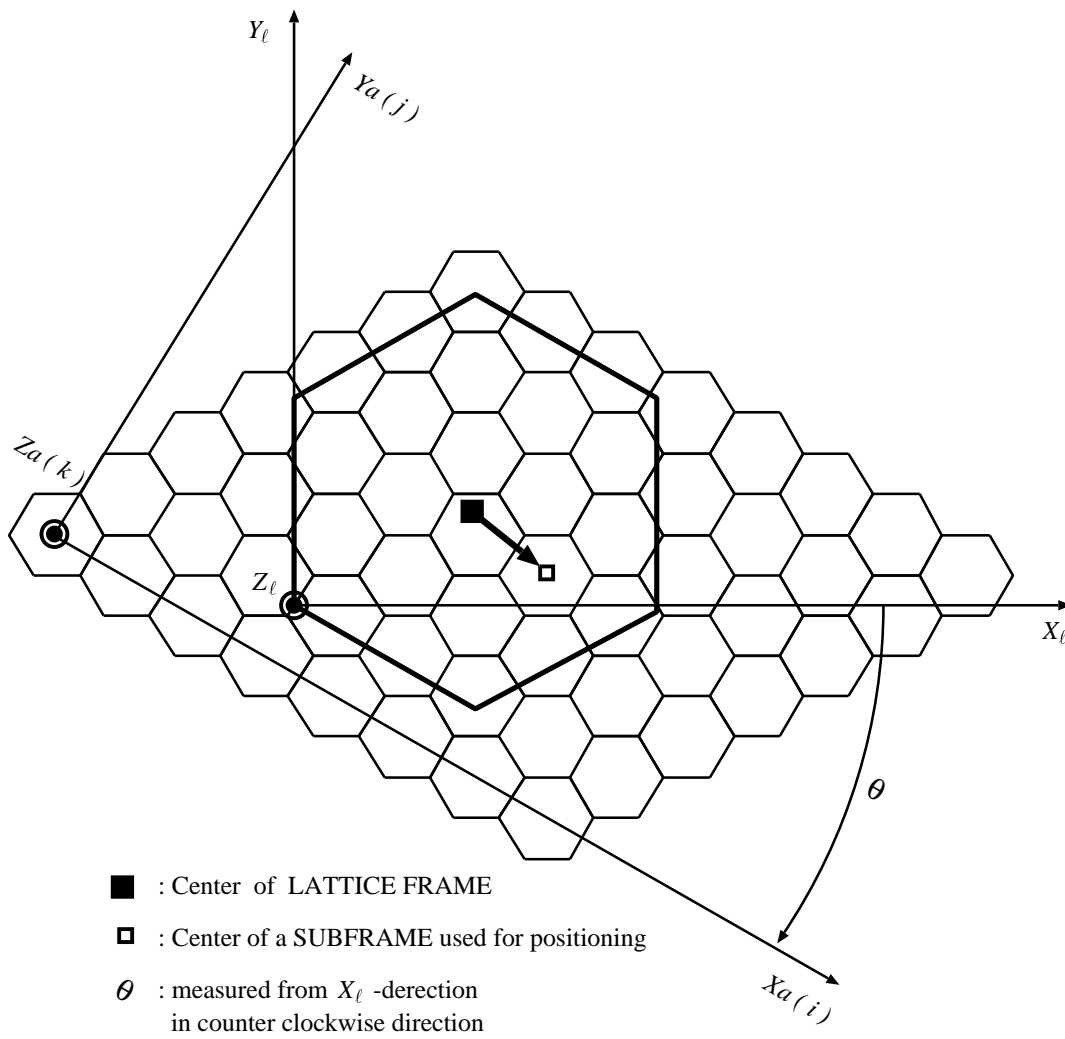


Fig. 5.11 LATTICE coordinates (X_ℓ, Y_ℓ, Z_ℓ) of a right hexagonal prism LATTICE and CELL-array coordinates (X_a, Y_a, Z_a)

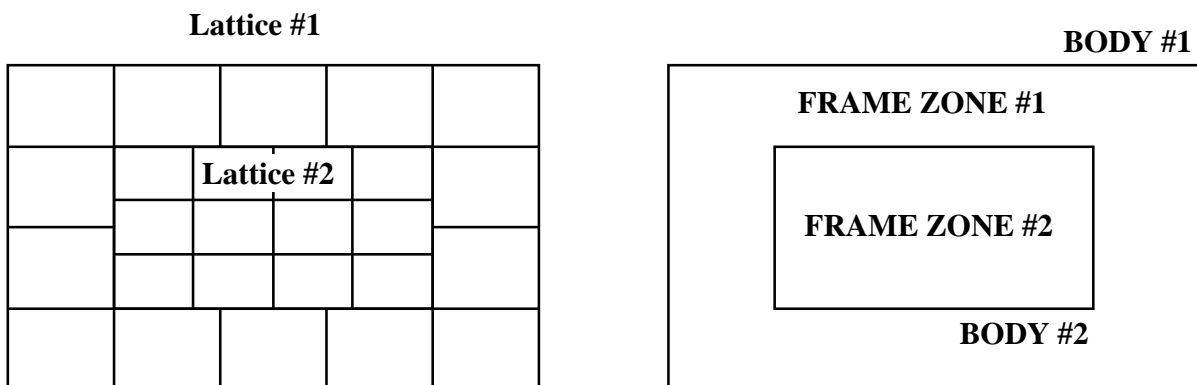


Fig. 5.12 Two LATTICES with different pitches

5.5.6 Allocation of LATTICE at FRAME ZONE

The FRAME ZONE is specified by the negative material ID numbers ($-998 \leq \text{KMAT} \leq -1$: The absolute value of this material ID number is the LATTICE ID number to be placed.) or the expression “LAT=LATTICE ID” or “L=LATTICE ID”.

The position and direction of a LATTICE placed at a FRAME ZONE is determined by the first BODY in the definition of the FRAME ZONE or the BODY defined as “FRAME=BODY ID”. The direction of each axis in the LATTICE coordinates (X_ℓ, Y_ℓ, Z_ℓ) is determined as listed in Table 5.2.

Table 5.2 Direction of the axes in the LATTICE coordinates

BODY	X_ℓ -axis	Y_ℓ -axis	Z_ℓ -axis	Remarks
RPP	X-direction	Y-direction	Z-direction	
BOX	Direction of the first side vector \mathbf{A}_1	Direction of the second side vector \mathbf{A}_2	Direction of the third side vector \mathbf{A}_3	See Fig. 10.5 for the definition of \mathbf{A}_1 , \mathbf{A}_2 and \mathbf{A}_3 .
RHP	X-direction	Y-direction	Z-direction	
HEX	Direction of vector \mathbf{A}	Direction of $\mathbf{H} \times \mathbf{A}$	Direction of the central axis vector \mathbf{H}	See Fig. 10.10 for the definition of \mathbf{A} and \mathbf{H} .
RCL	Direction of vector \mathbf{A}	Direction of $\mathbf{H} \times \mathbf{A}$	Direction of the central axis vector \mathbf{H}	See Fig. 10.11 for the definition of \mathbf{A} and \mathbf{H} .

The origin in the LATTICE coordinates is defined as shown in the left figures of Fig 5.10. The FRAME BODY specified by “FRAME=BODY ID” in the definition of the FRAME ZONE is used only for the allocation of the LATTICE. It has no meaning in the definition and does not affect the shape of the FRAME ZONE.

The coordinate transformation from the absolute coordinates (coordinates of the space where the FRAME ZONE is defined) $\mathbf{X}^t = (X, Y, Z)$ to the LATTICE coordinates $\mathbf{X}_\ell^t = (X_\ell, Y_\ell, Z_\ell)$ is performed as follows:

$$\mathbf{X}_\ell = \begin{bmatrix} X_\ell \\ Y_\ell \\ Z_\ell \end{bmatrix} = \begin{bmatrix} \mathbf{e}_{x_\ell}^t \\ \mathbf{e}_{y_\ell}^t \\ \mathbf{e}_{z_\ell}^t \end{bmatrix} (\mathbf{X} - \mathbf{V}_0),$$

where \mathbf{V}_0 is the position vector of the origin of the LATTICE coordinates; $\mathbf{e}_{x_\ell}, \mathbf{e}_{y_\ell}, \mathbf{e}_{z_\ell}$ are the unit vectors in the X_ℓ, Y_ℓ, Z_ℓ -directions, respectively; the superscript t means transpose.

5.5.7 Shape of FRAME ZONE

The shape of the FRAME ZONE depends on the FREE-LATTICE-FRAME option.

- (1) Without the FREE-LATTICE-FRAME option:

The shape of the FRAME ZONE must be fitted to the LATTICE FRAME. If the FRAME ZONE is larger than the LATTICE FRAME, there exist undefined regions. Otherwise, there exist multiply defined regions.

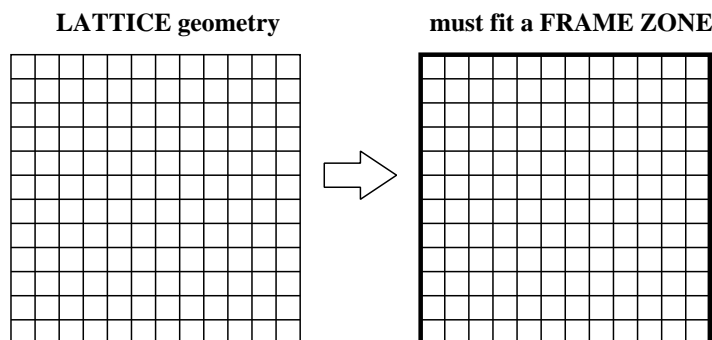
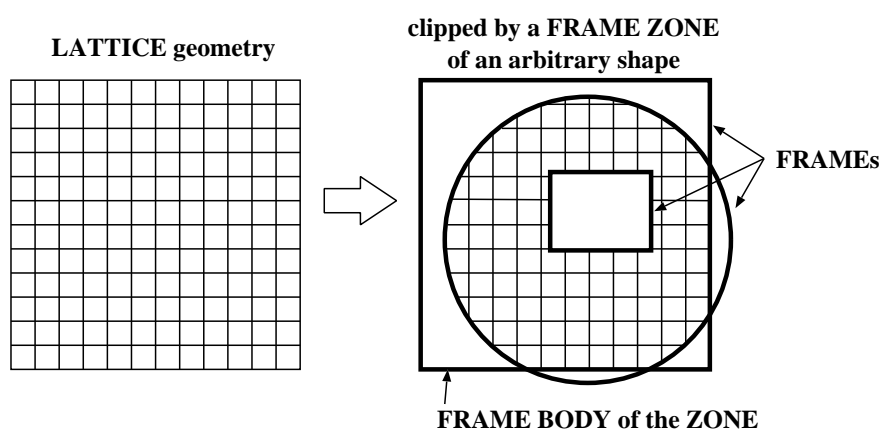
(a) LATTICE geometry mapping of non-FREE-LATTICE-FRAME**(b) LATTICE geometry mapping of FREE-LATTICE-FRAME**

Fig. 5.13 Free and conventional LATTICE FRAMEs

(2) With the FREE-LATTICE-FRAME option:

The shape of the FRAME ZONE is arbitrary. It is allowed that CELLS are cut out at the boundary of the FRAME ZONE for both rectangular parallelepiped and right hexagonal prism LATTICES. In order to allocate the LATTICE in the FRAME ZONE, the definition of the FRAME ZONE must include the BODY to define the LATTICE position corresponding to the LATTICE type (FRAME BODY). The specification of this BODY is described in Section 5.5.6.

Figure 5.13 shows the difference of treatment for the LATTICE geometry with and without the FREE-LATTICE-FRAME option. In the conventional LATTICE geometry of a right hexagonal prism LATTICE, there are some cases where geometrical modeling is incorrect for the nested LATTICE geometry. Figure 5.14 shows such an example, where the FRAME in level #1 crosses the CELLS in level #2 (Line A-B in Fig 5.14) and the CELLS are partially cut out. In this example, the FRAME in level #1 has been neglected in level #2. Such a LATTICE geometry can be modeled correctly with the FREE-LATTICE-FRAME option.

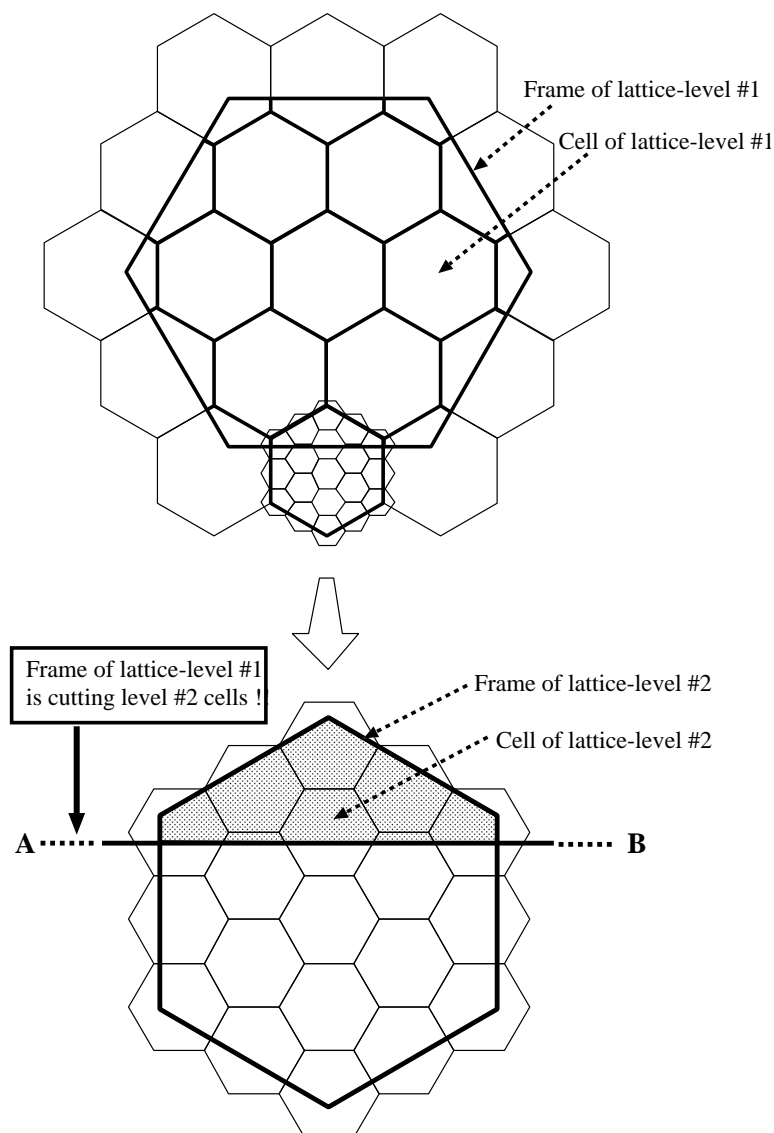


Fig. 5.14 Nested LATTICE geometry where the LATTICE FRAME crosses the CELLS

5.6 Statistical Geometry Model

The statistical geometry model is implemented into MVP/GMVP to analyze accurately the heterogeneity effect of randomly distributed coated fuel particles used in High Temperature Engineering Test Reactor (HTTR) and so on. Murata et al. firstly applied the statistical geometry model to Monte Carlo codes and developed the MCNP-CFP⁵¹⁾ and MCNP-BALL⁵²⁾ codes. In the statistical geometry model, the position of the fuel particles is determined probabilistically from the nearest neighbor distribution (NND) in the process of random walk. The NND is determined from the packing fraction of fuel particles in the fuel region.

The MVP/GMVP code have the statistical geometry model based on the principle similar to MCNP-CFP. Figure 5.15 shows the schematic view of the statistical geometry model in MVP/GMVP.

The features of the statistical geometry model are as follows.

- (1) The arbitrary number of spatial regions where coated fuel particles are placed probabilistically (STGM regions) can be defined with arbitrary shapes.
- (2) The arbitrary number of coated fuel particles (STG particles) can be defined as spherical spatial regions. Any spatial structure can be defined in the STG particles and furthermore STGM regions can be defined in the STG particles.
- (3) The multiple STG particles which have different materials and spatial structures can be included in a STGM region but their radii must be the same. The STG particles are not necessarily fuels.
- (4) The STGM regions can be used with the LATTICE geometry of MVP/GMVP and be placed in any level of the nested LATTICE geometry.
- (5) The distribution calculated by the MCRDF code⁵³⁾ developed for MCNP-CFP and the analytical distribution function based on the statistically uniform distribution are available for NNDs. The NND with the different packing fraction can be given in each STGM region.
- (6) In the MCNP-CFP code, STG particles cannot be placed on the boundary of the STGM region but it is possible in the statistical geometry model of MVP/GMVP.
- (7) The FREE-LATTICE-FRAME option is automatically set for the MVP/GMVP input data where STGM regions are defined, though the LATTICE option must be specified by the user.

The eigenvalue, fixed-source and neutron-photon coupled calculations can be performed with STGM regions. The restart option is also available with the STGM regions. Currently there are, however, following restrictions.

- Eight types of STG particles can be distributed in one STGM region.
- Not only spherically symmetric geometry but also any geometry can be defined in a STG particle. The direction of the STG particle is not determined at random but it is fixed such that the directions of the axes in the coordinates of the STGM region are the same as those in the coordinates where the STG particles are defined.
- The torus shape cannot be used to define STGM regions and STG particles.
- There is an option by which the fission source data can be stored in a file in an eigenvalue problem but no information about the allocation of STG particles in which fissions occur is included in the file.

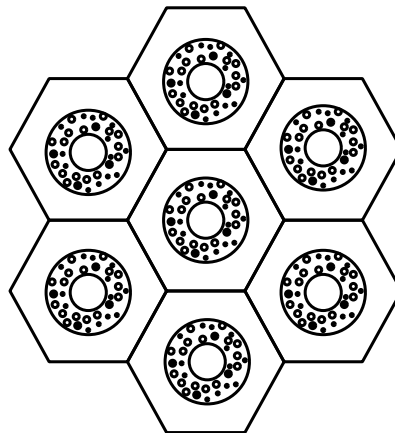
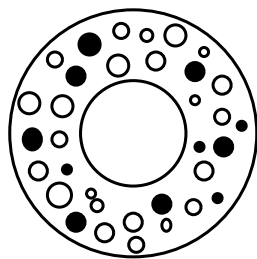
5.6.1 How to Input the Statistical Geometry Model

In order to use the statistical geometry model, the following must be specified in the input data:

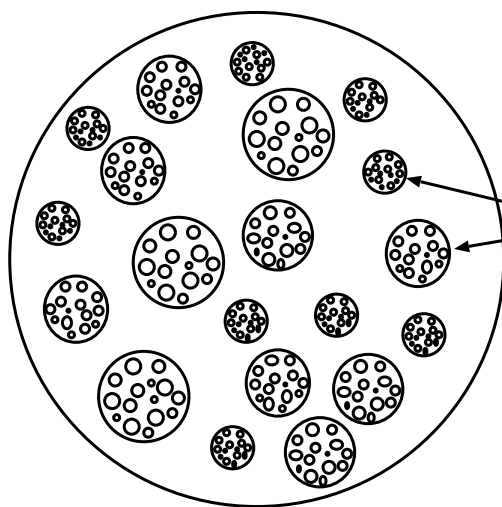
- The LATTICE geometry must be used (LATTICE option) even if there are no repeated structures in the geometry.
- A STGM region must be specified as a LATTICE of TYPE 10. The LATTICE consists of STG particles and the medium (matrix) in the other space of the STGM region than STG particles.
- A STG particle must be defined as a CELL (TYPE “STGP”). This type of CELL is called STGP CELL.

STGM region of arbitrary shape

STGM region in CELLS



STG particles in STG particle



The radius of particles
are the same.

Fig. 5.15 Schematic view of the statistical geometry model in MVP/GMVP (STGM region and STG particle)

- A fictitious CELL (TYPE “MBASE”) must be defined to specify the medium (matrix) in the STGM region and a tally region. This type of CELL is called MBASE CELL.

The details are described in Section 10.8 “Input Data for Statistical Geometry Model”.

5.7 Hierarchical Expression of REGION Names

In combinatorial geometry, each ZONE is labeled a REGION name and ZONES with the same REGION name belong to the identical REGION. Thus each ZONE in the same CELLS placed in some SUBFRAMEs occupies the different partial space in the absolute space but it cannot be distinguished from others. The TALLY-LATTICE option (or FRAME-DEPENDENT-TALLY option) enables us to distinguish such ZONES.

When this option is specified, different partial spaces filled with the same LATTICE can be distinguished by labeling FRAME ZONES. The name of a FRAME ZONE is called a FRAME name. In addition, each SUBFRAME in the CELL array is distinguished by the SUBFRAME name labeled each SUBFRAME. As a result, the partial space (SUBSPACE) occupied by each SUBFRAME in the universe

of Level 1 is distinguished by

FRAME name + SUBFRAME name .

It is called "SUBSPACE name" and is expressed as

! FRAME name : SUBFRAME name

in MVP/GMVP. Thus, taking the hierarchical structure into account, a ZONE in a CELL placed in the partial space specified by this SUBSPACE name has the following REGION name:

! FRAME name : SUBFRAME name! REGION name of each ZONE,
(Level 0) (Level 1)

where the REGION name of each ZONE is the REGION name given in the ZONE definition and the symbol "!" is a delimiter that represents the beginning of a REGION name for a ZONE or a FRAME ZONE in each level. In the universe of Level 2, the REGION name is

! FRAME name : SUBFRAME name! FRAME name : SUBFRAME name
(Level 0) (Level 1)
! REGION name of each ZONE
(Level 2)

and it is expressed similarly in any level. The form,

! FRAME name : SUBFRAME name ,

which appears repeatedly in the above expression, is called "SUBSPACE name in each level" and the SUBFRAMEs in a level are distinguished with the FRAME name in the upper level. Thus, the SUBFRAMEs in an arbitrary level belong to the partial space with the following SUBSPACE name. (It is similar to hierarchical directories in UNIX.)

SUBSPACE name = SUBSPACE name in the upper level ! FRAME name : SUBFRAME name ,

where the SUBSPACE in the upper level is the space where the FRAME of the SUBSPACEs in the current level is defined. As is found in this form, the SUBFRAMEs of the same "! FRAMEname : SUBFRAME name" belong to the different SUBSPACEs when the SUBSPACEs are different in the upper level.

MVP/GMVP considers that there are no SUBSPACE names in the universe of Level 0, since ROOT-FRAME = ROOT-SUBFRAME where ROOT-CELL is allocated from the viewpoint of the hierarchical structure. Finally, the REGION name of an arbitrary ZONE in an arbitrary level is as follows:

REGION name = SUBSPACE name ! REGION name in each ZONE .

5.8 TALLY REGION

"TALLY REGION" is a spatial region where physical quantities (fluxes etc.) are tallied by estimators such as the track length estimator, the collision estimator and so on. The TALLY REGION can be specified with the combination of more than one REGION. In the codes, the physical quantities are tallied in each REGION labeled a REGION name ("SUBSPACE name ! REGION name in each ZONE" if the TALLY-LATTICE option is used, otherwise "REGION name in each ZONE"). Without the definition of TALLY REGIONS, each REGION labeled a REGION name is a TALLY REGION.

There are two modes to specify TALLY REGIONS in MVP/GMVP. The TALLY REGIONS can be defined as the sum of REGIONS in both modes but there is the following difference.

- (1) ADD mode : A new TALLY REGION is added. In this case, tallied results of each REGION in the TALLY REGION are output.
- (2) DEFINE mode : A new TALLY REGION is defined. No tallied results of each REGION in the TALLY REGION are output.

The specification of TALLY REGIONS makes the following possible.

- The averaged value of physical quantities over TALLY REGIONS with those in each REGION (e.g., each pin power in a fuel assembly and the averaged power over the fuel assembly) can be calculated simultaneously in the ADD mode.
- Physical quantities can be tallied in a TALLY REGION comprised of ZONES in different levels with the TALLY-LATTICE option. For example, when a fuel assembly (Level 1 in the whole core calculation) is defined by the ZONES of a wrapper tube, coolant (Level 1), and the LATTICE of fuel pin cells (Level 2), ZONES in the different levels cannot be defined as the same REGION in the specification of FRAME-SUBFRAME but can be defined as the same TALLY REGION. Unless the TALLY-LATTICE option is used, physical quantities are tallied in the REGION of the same REGION name regardless of the level.
- TALLY REGION names can be used for the specification of REGION-wise data such as parameters for variance reduction. In this case, input data is specified for all REGIONS included in the TALLY REGION.
- If some REGIONS must be defined distinctively for the input parameters of variance reduction etc. but physical quantities need not to be tallied in each REGION, the amount of output data and required memory can be reduced by defining a TALLY REGION comprised of the REGIONS in the DEFINE mode.
- Since the results in specified TALLY REGIONS are output in prior to those in normal REGIONS, it is easy to understand the output data.

5.9 REGION Name and REGION Number

The numbering of REGION names (REGION number) depends on the TALLY-LATTICE option. Unless the TALLY-LATTICE option is used, the REGION name of each ZONE is first ordered according to the following procedure and then the REGION number is determined.

1. First, the REGION names are ordered by the number of characters. → A REGION name of less characters has priority.
2. Next, the order of characters is determined as follows (“<” means the left side has priority.):

Special characters < Lowercase letters (a < b < ... < z) < Uppercase letters (A < B < ... < Z) < Numbers (0 < 1 < ... < 9).

The REGION names are case sensitive.

3. The REGION names ordered in the above procedure are numbered from high to low priority. But FRAMES and ZONES with fictitious materials for boundary conditions are not numbered. Even if the REGION name is given to such a FRAME or a ZONE, it is ignored.

Examples :

- If REGION names are specified with positive integers only, then the order of the names is the ascending order of the number.
- A < 1
- A2 < A3
- B2 < A10
- aAA < AAA
- A_CD < ABCD

On the other hand, when the TALLY-LATTICE option is used, the REGION names are numbered in the order defined by the ZONE data (including REGION name, FRAME name and SUBFRAME name). In this case, it is difficult to know the REGION number for each ZONE from the input data. MVP/GMVP prints the table of REGION numbers and REGION names after reading and processing the geometry data.

[Note] Results by the REGION are output in the order of the REGION number.

6. Input Instructions

The user should prepare MVP/GMVP input data in the text form that can be read from standard input on unit 5. We call it just “input data” in the following chapters. The user should also assign cross section index files, a restart file, etc.

6.1 Features

Special features for the MVP/GMVP input data are as follows.

- Input with a variable name
Data should be input with the variable name, so it is easy to understand values assigned to variables.
- Free format
All input data are input in the free format. The user does not have to care about the column-adjustment of data values.
- Description of comments
The user can insert arbitrary comment lines or comments in the input data in addition to the data that is related to calculation directly.
- Description of numeric data by equations or symbolic parameters
The four basic operations of arithmetic or basic mathematical operations are available for data input. In addition, the user can define “labeled values” (symbolic parameters) and use them in the input data. It enables us to label values and modify the values all together.
- Labeling of data
The user can label regions in space, geometrical objects, material compositions, etc. by character strings and arbitrary numbers.

By using the above features, the user can prepare the flexible input data that can be easily understood or modified.

6.2 Structure of Input Data

The input data consist of the following data blocks.

```
Title 1 <The first two lines are title ones which are used to describe a problem, etc.>
Title 2
options .... <Character strings to control the program (options)>
options ...
...
options
```



```

    <Terminate the options by more than one blank lines>
data ...
data ... <Data to describe the problem in detail>
data ...
.....
data
...
/ <Terminate the input data by a character "/">

```

6.3 Effective Columns and Comments in a Line

Both effective data for MVP/GMVP input and comments should be given in accordance with the following rules.

- The 1st through 72nd columns are active in each input line. The 73rd column and beyond are ignored.
- Lines whose first column is "*" are ignored except for the title lines. These are read as comment lines.
- Characters after "/" in a line are ignored except for the title lines. They are also used as comments.
- Blank lines are also ignored except for the title lines and the option block.

Example:

`Framed` parts are active.

```

MVP test problem
This data is only for data/comment part explanation
FISSION PICTURE
* .... '*' on the first column means comment line
NEUTRON /* neutron problem is solved

* .... option section ended with the blank line above
$GEOMETRY
RPP( 1  0 1 0 10 -1 100 )
END

VOID :  : -1000: -1
HAKE : 1 : 1 : 1
$END GEOMETRY
/

```

6.4 Option Block

The user should specify the following character strings in the option block. The characters can be written at any location in the 1st through 72nd columns. Blanks are the delimiters for the options.

[NO-]*option-string*[(*parameter* ...)]

Characters in [] are specified when required. *option-string* is a character string which shows the options and *parameter* ... is a parameter required for the option. If “NO-” is added to the head of the character string for the option, this option is switched off (ineffective).

Example:

```

NEUTRON      /* NEUTRON CALCULATION
NO-RESTART  LATTICE                      /* NECESSARY FOR LATTICE GEOMETRY
DYNAMIC-MEMORY( 3000000 )
*
* --- VARIANCE REDUCTION CONTROL ---
*
NO-RUSSIAN-ROULETTE  WEIGHT-WINDOW  NO-IMPORTANCE
*
* --- TALLY ----
*
NO-UNIVERSE-DEPENDENT-TALLY      /* UNIVERSE-DEPENDENT-TALLY
RESPONSE                        /* REACTION RATE USING RESPONSE FUNCTION.
EDIT-BY-TRACK-LENGTH           /* ESTIMATOR FOR TALLY OTHER THAN FLUX & KEFF
**EDIT-BY-COLLISION            /* ESTIMATOR FOR TALLY OTHER THAN FLUX & KEFF
FLUX-PRINT                     /* PRINTOUT OF FLUX

```

The DYNAMIC-MEMORY option is an example for the parameter (numeric data in this case) specified in the parentheses following the option string.

Attention should be paid for the comment description by using “/*” in the option block. If there are only blanks before “/*” or a line begins with “/*”, then the line is interpreted as a blank line and thus the option block is terminated.

6.5 Format of Input Data Description

The user should input the following data after the blank line which shows the termination of the option block.

- (1) Data to control random walk
- (2) Cross section data and material composition
- (3) Geometry data
- (4) Source data
- (5) Data for variance reduction
- (6) Data for secondary particles

(7) Tally data

(8) Data about printout to check geometry data

The user should pay attention to what follows for preparation of these data.

- The number of data to be specified should be input in prior to the data in principle. Templates for input data are prepared in the `template` directory for the UNIX version, so the user can prepare an input data by modifying the template data in most cases.
- Some data should be given in a special “data block” such as the CROSS SECTION data block which is located between `$CROSS SECTION` and `$END CROSS SECTION`. On the contrary, the user can input the data in any place outside “data blocks” unless a “data block” is specified for it.
- It is allowed that unnecessary data are included in the input data. These data are ignored or not used in the calculation. On the contrary, a fatal error occurs when necessary data for some options are not input and even the default values are not defined.
- If the same data is input more than once, the last data is usually effective.

6.5.1 Labeled Free Format Input

In general, the user should input the data in the following format. The 1st through 72nd columns in a line is available.

data-name (*data data ...*)

data-name is the name of a data set. *data data ...* are a series of data, each of which is a numeric value or a character string. One or more blanks are necessary between the data as a delimiter. It is allowed that a series of data is specified by more than one line but the label of data and the following “(” must be in the same line.

6.5.2 Data Block

Some data should be input by the following “data block”.

\$data-block (“\$” should be in the first column in the line.)
 ...
\$END [data-block] (“\$” should in the first column in the line.)

or

#data-block (“#” should be in the first column in the line.)
 ...
#END [data-block] (“#” should be in the first column in the line.)

In general, the data block starting with a symbol “#” is usually used as a data sub-block in the “\$” data block.

6.5.3 Labeled Value and Arithmetic Operation

In the format of “*data-name* (*data data ...*)”, each ‘*data*’ is generally a value or a character string. Furthermore, mathematical expressions using variables (labeled values: symbolic parameters) and numerical values are available for the input of values as described below.

● Direct specification of values

```
NGROUP( 26 )
ENGYB(          /* ENERGY BIN (UPPER BOUNDARIES + LOWEST BOUNDARY)
1.6487E+7 4.7236E+6 1.0540E+6 1.8315E+5 5.2475E+4 1.5034E+4 5.5307E+3
5.8293E+2 1.3007E+2 2.9022E+1 6.4758E+0 1.8554E+0 1.2752E+0 8.7643E-1
6.8256E-1 4.1399E-1 3.4206E-1 2.7699E-1 2.1878E-1 1.6742E-1 1.2293E-1
8.5295E-2 5.4519E-2 3.0601E-2 1.3542E-2 3.3419E-3 1.0E-5   )
```

- The floating-point data should be specified like “3.0601E-2” or “3.0601e-2” (“3.0601D-2” or “3.0601d-2” for double precision) as a usual FORTRAN expression.
- A value data should not include blanks and two data should be separated by one or more blanks or different lines.

< wrong example >

```
- 1.2345 E-6      (including blanks)
1.2345-10        (E or D must be used for the powered format.)
1.2345E-10-2.354 (Blanks are necessary between input data.)
```

● **Expression by arithmetic operation** The user can use the following input form instead of specifying values directly.

```
C000000J40(< 0.9226 / (12.011 + 2*1.00794) * 6.022137E23 * 1.0E-24 >)
```

A mathematical expression must be specified in the region parenthesized by “< >” and the result of the operation is taken as the input value. Nested parentheses “()” can be used with the following operations and basic functions.

- available operations:

five basic operations: +, -, *(multiplication), /(division), *(power)

- basic functions:

The user can make use of the mathematical functions listed in Table 6.1.

● **Definition and reference of labeled value (symbolic parameter)** The user can define a labeled value by assigning a specific value to a character string called a ‘symbolic parameter’ and refer it in input data. The following input data is an example.

```
% NG = 64, NGN = 5
NGROUP( <NG> )
NGROUP.N( <NGN> ) NGROUP.P( <NG-NGN> )
```

The above input data is the same as the next example.

Table 6.1 Basic functions and logical/relational operators available in input data

Function	Meaning
SQRT(x)	square root
SIN(x), COS(x), TAN(x)	trigonometric functions (unit : radian)
ASIN(x), ACOS(x), ATAN(x)	inverse trigonometric functions
SIND(x), COSD(x), TAND(x)	trigonometric functions (unit : degree)
EXP(x)	exponent
LOG(x)	natural logarithms
LOG10(x)	common logarithms
ATAN2(x, y)	$\arctan(x/y)$
SINH(x), COSH(x), TANH(x)	hyperbolic functions
INT(x)	integer part of a real number
NINT(x)	the nearest integer
ABS(x)	absolute value
MOD(x, y)	residue (residue of x divided by y)
MAX(x, y, z, \dots)	maximum value (arbitrary number of arguments)
MIN(x, y, z, \dots)	minimum value (arbitrary number of arguments)
IF(x, y, z)	y if $x \neq 0$, z otherwise.
AND(x, y)	logical product supposing that 0 is false, otherwise true
OR(x, y)	logical sum supposing that 0 is false, otherwise true
NOT(x)	logical denial supposing that 0 is false, otherwise true
EQ(x, y)	1 if $x = y$, 0 otherwise.
NE(x, y)	1 if $x \neq y$, 0 otherwise.
GT(x, y)	1 if $x > y$, 0 otherwise.
GE(x, y)	1 if $x \geq y$, 0 otherwise.
LT(x, y)	1 if $x < y$, 0 otherwise.
LE(x, y)	1 if $x \leq y$, 0 otherwise.

NGROUP(64) NGROUP.N(5) NGROUP.P(59)

The user must obey the following rules for the use of symbolic parameters.

1. A symbolic parameter must be a string of not more than 8 characters. Alphabetical letters, numbers and periods(.) can be used for the symbolic parameter but the first character must be an alphabetical letter. Symbolic parameters are case sensitive. For example, “NG” and “ng” are different symbolic parameters. The maximum number of available symbolic parameters is 4,096 and this value is fixed in the codes.
2. Symbolic parameters can be defined in the following 2 ways:
 - Parameter lines (The first column is “%”)

$\% \text{ name1} = \text{value1} [, \text{name2} = \text{value2} \dots]$

The user can define symbolic parameters in lines whose first column is “%”. These lines are called “parameter lines” in this manual. Any number of parameter lines can be defined and they can be inserted in any position except for the title section. Multiple symbolic parameters can be defined in a line but each definition must be set apart by a comma “,”. Blanks have no

meaning in the parameter lines. A symbolic parameter must be defined in a single parameter line (not be defined in multiple lines).

- Parentheses “< >” in labeled free form input data

data-name(<*name1* = *value1*>)

In this case, the value of the symbolic parameter *name1* is set to *value1* and the variable *data-name* is also set to *value1* at the same time. The rule in the parentheses “< >” is the same as the one in parameter lines. If more than one parameter are defined in the parentheses, the value of the last parameter is set to the variable.

3. Symbolic parameters can be used only for arguments of mathematical operations in “< >” or the parameter lines*. The symbolic parameters must be defined in prior to their use.

If some input values are defined with symbolic parameters, some other values may be expressed by using these symbolic parameters as shown in the above example. This function can make parametric-survey calculations easy by defining the parameters are defined with the symbolic parameters. This is also helpful to clarify the meaning of input values and to decrease errors in input data.

Note :

- A pair of “< >” should be in the same line in the current MVP/GMVP version.

6.5.4 Repeated Values and Equally Spaced Values

The user can input the repeated or incremental data values by using simple notation.

● **Repeated values** The user can specify the number of repetitions for specific values in the labeled free-format input as follows.

n (*data data ...*) or R (*data data ...*)

n is the number of repetitions. “*data data ...*” shows one or more input values which are repeated.

- *n* must be a positive integer or 0. If it is not specified, the MVP/GMVP code interprets as 1. If *n* = 0, *data data ...* is ignored.
- An expression with symbolic parameters (“< >”) can be used for *n*.
- There must not be blanks between *n* and the following “(”.
- The nested structure is allowed. (max. by 15 times)
- If a character “R” is specified instead of the number of repetitions, the rest of data is filled with the values in “()” following “R”. In this case, the number of data to be input must be defined in prior to the “R” input.

Example : The following data are the same. (Suppose the data name is “M”.)

*There are some exceptional cases in the source input where symbolic parameters can be referred to without “< >”.

```

M( 2( 1 3( -1 8) ) )
M( 1 3(-1 8) 1 3(-1 8) )
M( 1 -1 8 -1 8 -1 8
    1 -1 8 -1 8 -1 8 )

```

--> M(1)=1, M(2)=-1, M(3)=8, M(4)=-1, M(5)=8, M(6)=-1, M(7)=8,
M(8)=1, M(9)=-1, M(10)=8, M(11)=-1, M(12)=8, M(13)=-1, M(14)=8

Example : Repetition with “R” (Suppose the data name is “L” and the number of necessary data is 11.)

```
L( 3(1) R( 4 3 2) )
```

```

-->      L(1)=L(2)=L(3)=1,
          L(4)=4, L(5)=3, L(6)=2,
          L(7)=4, L(8)=3, L(9)=2,
          L(10)=4, L(11)=3

```

● **Equally spaced values** Equally spaced values are specified as follows.

$n : data$

In the above example, n is the number of data generated by adding the increment to the preceding data and *data* is the increment.

- n should be a positive integer. If n is not specified or 0, “ $n : data$ ” is ignored.
- An expression with symbolic parameters (“< >”) can be used for n .
- There must not be blanks between n and “:”[†].
- “ $n : data$ ” is not allowed for the first data of the free form input data.

Example : The following expressions are the same.

```

IA( 1 9:2 )
IA( 1 3 5 7 9 11 13 15 17 19 )

```

6.5.5 Operations and Functions Applicable to Data Series

The user can apply a list of operations and functions to multiple input values (data series).

Example :

```

DATA( 10 20 30 {X; X - SIN(X/3.14159265)}(1 2 3 2(3 4 5) 8:2)
      R(0) )

```

In this example, the data list “(1 2 3 2(3 4 5) 8:2)” is evaluated first and then the input values are generated from the expression “X - SIN(X/3.14159265)” where the evaluated data are used successively as the variable X.

The general form is written as follows:

[†]Blanks can be inserted between “:” and *data*. It is, however, not recommended for preventing input errors.

$$\{ \text{var} ; \text{expression} \} (\text{data data} \dots)$$

where *var* is the variable that expresses the value in the following data list and *expression* is the function which involves the variable. The variable and the function are delimited by “;” (semi-colon) and they must be closed by “{ }”. The expressions, functions and symbolic parameters explained in “Labeled Value and Arithmetic Operation (p. 78)” are available to define the functions. Blanks have no meaning in “{ }” as in “< >”. The user can insert blanks between “{ }” and the following “(”, but they must be specified in a single line.

Example : The following example specifies energy boundaries with equal lethargy width. The total number of energy bins is NG.

```
% EMAX=2.0E7, EMIN=1.0E-5, NG=4
ENGYB( { X ; <EMAX>*( <EMAX/EMIN> )**(-X/<NG>)} ( 0 <NG>:1 ) )
```

This input data is the same as the following one.

```
ENGYB( 2.0E7 1.68179E4 1.41421E1 1.18921E-2 1.0E-5 )
```

6.5.6 Reference of Variables Defined in the Codes as Symbolic Parameters

The user can refer to variables defined internally in the codes as the symbolic parameters. A character “%” must be attached to the head of the variable name to refer to. The currently available variables are listed in Table 6.2. However the user must pay attention to the reference in prior to the definition of the variable in the codes. In such a case, the value of the variable is not guaranteed.

Example : Usage in parameter lines

```
% NR = %NREG
% NG = %NGROUP
WKIL( <NR*NG>(0.1) )
```

Example : Direct reference in the labeled free-format input

```
WKIL( <%NREG*%NGROUP>(0.1) )
```

6.6 Data Input by REGION

REGION numbers are not input by the user directly but the codes set them automatically in the order which corresponds to the REGION name, so it is difficult for the user to input the data dependent on REGIONs in order of REGION numbers. In addition, it is sometimes difficult to find the number of REGIONs in case of the TALLY-LATTICE option. In order to overcome this difficulty, the following two types of input formats with the REGION or TALLY-REGION name are available in addition to the standard labeled free format.

Table 6.2 Variables which can be referred to as symbolic parameters

Variable	Meaning	Time when the variable is set
NGP1	Number of groups for neutron (=NGROUP.N)	NGP1 is input.
NGP2	Number of groups for photon (=NGROUP.P)	NGP2 is input.
NGROUP	Number of groups for all particles	After NGP1, NGP2 are input.
NTIME	Number of time bins	Tally data are input.
NHIST	Number of source particles per a batch	NHIST is input.
NINPZ	Number of INPUT ZONES	Geometry data are input.
NMAT	Number of materials for macroscopic cross section	Cross section data are input.
NZONE	Number of zones	Geometry data are input.
NREG	Number of regions	Geometry data are input.
NTREG	Number of tally regions	Geometry data are input.
NPKIND	Number of particle types	Option data are input.
DEPS	Minimum distance to judge in codes	DEPS is input.
DINF	Distance interpreted as infinity in codes	Set by default.
NUC	Number of nuclides (MVP only)	Cross section data are input.
NPATOM	Number of atomic elements (MVP only)	Photon cross section data are input.
ETOP	Highest energy for neutron (MVP only)	ETOP is input.
EBOT	Lowest energy for neutron (MVP only)	EBOT is input.
ETOPP	Highest energy for photon (MVP only)	ETOPP is input.
EBOTP	Lowest energy for photon (MVP only)	EBOTP is input.
NGPX	Number of groups for primary particles (GMVP only)	Cross section data are input.
NGGX	Number of groups for secondary particles (GMVP only)	Cross section data are input.
NTGX	Number of groups for multigroup cross section (GMVP only)	Cross section data are input.

6.6.1 REGION-dependent Data

The data to be input by REGIONs are as follows.

- XIMP(%NGROUP, %NREG) : Importance by energy groups and REGIONs
- WKIL(%NGROUP, %NREG) : Parameter for Russian-roulette by energy groups and REGIONs
- WSRV(%NGROUP, %NREG) : Parameter for splitting by energy groups and REGIONs
- WGTF(%NREG) : Weight for fission neutrons by REGIONs
- WGTP(%NREG) : Weight for secondary photons by REGIONs
- WGTPNR(%NREG) : Weight for secondary photoneutrons by REGIONs
- PASLP(%NREG) : Path stretching factor

- TRVOL(%NTREG) : TALLY-REGION volume
- RVOL(%NREG): REGION volume

6.6.2 Input Format - 1

$$data-name(!region-name(data data \dots) \dots)$$

Example for XIMP

```
XIMP( !region1( data ... ) !region2( data ... ) ... )
```

In this format, the user inputs the REGION-dependent data for a specified REGION by its REGION name according to the following rules.

- The user can specify REGIONS by assigning REGION names or TALLY-REGION names (See Section 10.9 for the TALLY-REGION names.). The same data is assigned to all REGIONS included in the TALLY-REGION.
- Wild card characters “*” and “?” (See Section 6.6.4.) are available for the specification of REGION names.
- If some data are doubly defined for the same REGION, the last defined data is effective.
- The user can input the data for the labeled data more than once. The following two expressions are the same.

```
XIMP( !region1( data ) !region2( data ) )
XIMP( !region1( data ) ) XIMP( !region2( data ) )
```

- Unless the first character to specify the REGION name is “!” or “@”, all data must be input in the order of REGION numbers. (ordinal labeled free format input)

Example : Input without REGION names (not recommended)

```
XIMP( <NG>(2.0) <NG>(0.5) R(1.0) ) /* Number of input data
                                     /* is NGROUP*NREG.
```

REGION numbers are assigned by the codes automatically. So the user should not use this format except for the cases where REGION numbers are easily identified or the same data is assigned for all REGIONS.

6.6.3 Input Format - 2

$$!region-name(data-name(data data \dots) \dots)$$

Example for XIMP

```
!region1( XIMP( data ... ) )
!region2( XIMP( data ... ) )
```

6.6.4 Specification of REGIONS by Wild Card Characters * and ?

Wild card characters (meta-characters) are available for the specification of REGIONS as is used in UNIX or MS-DOS.

- “?” : arbitrary one character
- “*” : zero or more arbitrary characters

(These are effective within REGION names for each ZONE, FRAME and SUBFRAME names.)

Example :

! * or *	→	All REGIONS belonging to ROOT-CELL (Level 0 universe)
! * : * ! * : * ! *	→	All REGIONS in all CELLS of Level 2 universe (Other levels are not included.)
!LAT1:SPEC1!*A	→	All REGIONS whose last name is “A” belonging to a SUBSPACE specified by !LAT1:SPEC1.

If the data for the same REGION is defined more than once because of the specification by wild card characters, the last data is effective. This feature is convenient to define the same data for all REGIONS as default and then modify the data for some REGIONS as shown below.

```
!*( WKIL(<%NGROUP>(0.001)) WSRV(<%NGROUP>(1.0)) )
!REGION8( WKIL(<%NGROUP>(0.1)) WSRV(<%NGROUP>(0.2)) )
```

or

```
WKIL( !*( <%NGROUP>(0.001)) !REGION8(<%NGROUP>(0.1)) )
WSRV( !*( <%NGROUP>(1.0)) !REGION8(<%NGROUP>(0.2)) )
```

7. Title and Options

7.1 Title for Input Data

The first two lines for input data must be title lines where the user can input any character strings to specify the problem, etc. This data is output to the standard output file (I/O unit 6), the binary result file (I/O unit 30), the restart file (I/O unit 20) and other files.

7.2 Rule for Option Block

The option block must be input following the title lines. Each option is specified by the following form of character strings.

`[NO-]word1 [-word2- ... -wordn] [(parameter ...)]`

word1, *word2*, ... are (sub-)character strings to describe options defined below. In case that more than one character string are required, each string should be connected by “-” (hyphen or minus). By adding “NO-” at the head, the option is switched off (inactive). The option block should be terminated by a blank line. The character strings for each option can be located anywhere in the 1st to 72nd columns but each option cannot be specified over two lines.

7.3 List of Available Options

The available options are as follows. Details of each option are described in Section 7.5. The options with ● mark are effective by default and those with ○ not by default. (See Table 7.1.)

○ ADAPTIVE-MICRO-CALCULATION

By default, microscopic cross sections for all nuclides are calculated after particle energy changes. But the cross sections for each nuclide are prepared when they are necessary with this option.

○ ADJOINT

Solve an adjoint problem. (Only for GMVP)

○ ANALOG-FISSION

The number of fission neutrons is determined by Terrell’s equation in a fixed source problem of MVP. (See Section 4.5.)

○ AUTO-RESTART

Restart a calculation automatically if a restart file exists on I/O unit 10. Unless the codes can find a restart file, an initial run will be performed. The restart file should be output on I/O unit 20 in the previous calculation.

○ BETA-EFFECTIVE

Calculate the effective delayed neutron fraction (β_{eff}) by approximating the adjoint neutron flux with the next fission probability.

○ BREMSSTRAHLUNG

Treat Bremsstrahlung X rays in a photon transport calculation.

○ COLLISION-LESS

Make generated particles fly without collisions by taking account of spatial attenuation. Uncollided fluxes can be calculated by this option.

● DELAYED-NEUTRON

Generate delayed neutrons in fission reactions.

○ DOPPLER-SCATTERING(*mode*)

Select an elastic scattering model in the resonance region. See Section 4.2.6 for details.

- *mode* = NONE : asymptotic slowing-down scattering model (asymptotic model).
- *mode* = NORES : free-monatomic-gas model with constant cross section (constant cross section model).
- *mode* = EXACT : free-monatomic-gas model with resonance cross section (exact model) for nuclides with 0 K elastic cross section. The asymptotic model is used for the other nuclides.
- *mode* = EXACT+ : free-monatomic-gas model with resonance cross section (exact model) for nuclides with 0 K elastic cross section. The constant cross section model is used for the other nuclides.

○ DYNAMIC-MEMORY(*size*)

Memory size for dynamic allocation in MVP/GMVP calculations. The sufficient but available size should be specified by word. A word is a memory unit to describe an integer data in FORTRAN language. (Normally 4 bytes.)

○ EDIT-BY-COLLISION

Use the collision estimator for cross-section and reaction rate calculations in MVP.

● EDIT-BY-TRACK-LENGTH

Use the track length estimator for cross-section and reaction rate calculations in MVP.

○ EDIT-MACROSCOPIC-DATA(*flag*)

Option to output (or print) macroscopic cross sections and reaction rates in MVP. The energy-group structure is specified with the input data ENGYB (p. 223). “*flag*” is an 8-digit integer. Each digit and its value show the reaction and process types, respectively.

$$\begin{aligned} \text{flag} = & 10^7 \cdot N_1 + 10^6 \cdot N_2 + 10^5 \cdot N_3 + 10^4 \cdot N_4 + 10^3 \cdot N_5 + 10^2 \cdot N_6 \\ & + 10 \cdot N_7 + N_8 \end{aligned}$$

Reaction types (digit of the flag)

- N_1 : total

- N_2 : nu-fission
- N_3 : fission
- N_4 : elastic scattering
- N_5 : capture (absorption-fission)
- N_6 : inelastic scattering
- N_7 : (n,2n)
- N_8 : absorption-(n,2n)-2(n,3n)-3(n,4n)

Process types (Value of N_i)

- 0: Not calculated.
- 1: Calculated. Output to the file (I/O unit 30) and not printed.
- 2: Calculated. Output to the file (I/O unit 30) and only cross sections are printed.
- 3: Calculated. Output to the file (I/O unit 30) and only reaction rates are printed.
- 4: Calculated. Output to the file (I/O unit 30) and the both are printed.

If “*flag*” is omitted, this option is not effective.

○ EDIT-MICROSCOPIC-DATA(*flag*)

Print (or output the binary result file) microscopic cross sections and reaction rates in MVP. The energy-group structure is specified with the input data ENGYB (p. 223). The input format and input data for *flag* are the same as the EDIT-MACROSCOPIC-DATA option.

○ EIGEN-VALUE

Solve an eigenvalue problem. (Calculate a multiplication factor)

○ FISSION

Generate fission neutrons. If the EIGEN-VALUE option is specified, this option is set automatically. This option is necessary to generate fission neutrons as secondary particles in a fixed-source problem where fissionable materials exist.

○ FISSION-MULTIPLICITY(*mode*)

Use ν_t or ν_p to determine the number of neutrons generated in fission.

- *mode* = TOTAL : ν_t (default)
- *mode* = PROMPT : ν_p

● FIXED-SOURCE

Solve a fixed-source problem.

○ FLUX-PRINT

Print particle flux (per lethargy) for each REGION and each energy group.

○ FREE-LATTICE-FRAME

Specify arbitrary-shape frame for the lattice when the LATTICE (REPEATED-GEOMETRY) option is used.

○ IMAGINARY-PARTICLE

Perform a transport calculation of imaginary particles which do not interact with any material and just fly straight from a generation point. In this case, all regions are assumed to be void. But treatment of boundary conditions, leakage, etc is the same as real particles. Setting the tally-region volumes to 1.0 and using an appropriate source such as a parallel beam, the region volumes can be calculated by this option.

○ IMPORTANCE

Use the importance variance reduction technique. Russian roulette kill and splitting are performed according to importance assigned by each region and energy group. (See Sections 2.9.1 ~ 2.9.4 .)

○ LATTICE or REPEATED-GEOMETRY

Use the lattice (repeated-geometry) expression.

○ MONITOR

The number of reactions or events and the averaged weights are monitored by each region and energy group. This option is useful for test calculations to search for better variance reduction parameters.

○ MULTI-TASK(*n*)

The number of tasks in multitask processing. (See Section 3.3.) Usually, it should be specified in command line options.

● NEUTRON

Solve a neutron transport problem.

○ PATH-STRETCH

Use the path stretching (exponential transformation) variance reduction technique. The particles move according to the weight in a direction by modifying the transport kernel. (See Sections 2.9.3.)

○ PERIODIC-TIME

Set flight time of a particle to 0 when the flight time exceeds the specified time in a time-dependent calculation.

○ PERTURBATION

Perform perturbation calculations. Currently MVP can calculate the change in the effective multiplication factor (k_{eff}) and kinetics parameters of the effective delayed neutron fraction β_{eff} and the neutron generation time Λ with the perturbation capability.

○ PHOTON[(*mode*)]

Solve a photon transport problem. The photon reaction model can be selected by a character string *mode* in MVP.

- *mode* = PHOTON : detailed model (default)
- *mode* = GAMMA : simplified model for high-energy photon transport

The detailed model is selected when “(mode)” is not defined.

○ PHOTO-NUCLEAR

Generate secondary particles from photonuclear reactions in MVP coupled neutron-photon problems.

○ PICTURE

Output cross-sectional views of a geometrical model by character patterns.

○ PRINT-SUPPRESS(I_1 I_2 I_3 ...)

Option to suppress the output for each item specified by I_j .

Items that can be suppressed (I_j) are listed as follows.

- 1: flux by each group
- 2: angular flux
- 3: time-dependent flux
- 4: response
- 5: time-dependent response
- 6: averaged values for source positions, etc. in each batch
- 7: information for neutron multiplication factors in each batch
- 8-16: (not defined)
- 17: microscopic reaction rates (MVP)
- 18: microscopic cross sections (MVP)
- 19: macroscopic reaction rates (MVP)
- 20: macroscopic cross sections (MVP)

○ RELATIVE-WEIGHT

Parameters of weights for Russian roulette, weight-window and generation of secondary particles are interpreted as relative values to the initial values of generated particles.

○ RESPONSE

Calculate reaction rates by using multigroup response functions.

○ RESTART

Perform a restart calculation after reading a restart file from I/O unit 10. The restart file is output to I/O unit 20 in the previous calculation.

● RESTART-FILE

Output a restart file. This option is effective by default and thus a user should use this option to suppress the restart file output by specifying “NO-RESTART-FILE”.

○ RUN-MODE[(*mode*)]

Specify a special action. Currently, the following mode is available.

- *mode* = 1 : Only the input data is processed and anything else (random walk, etc.) are not.

● RUSSIAN-ROULETTE

Use the Russian roulette variance reduction technique. (See Section 2.9.1.)

○ SCATTERING-MATRIX[(*order*)]

Calculate group-to-group scattering rates and scattering matrix moments.

- *order* : The order up to which scattering matrix moments are calculated.

○ SCATTERING-MUBAR

Calculate the average scattering cosine in the laboratory system.

○ SOURCE-OUTPUT

Output the fission source data from the last batch in an eigenvalue calculation. This data includes the fission sites and so on and is available as the source data in another calculation. The source data has a binary form and is output on I/O unit 9.

○ TALLY-LATTICE or FRAME-DEPENDENT-TALLY

Edit tallies for the regions defined by the FRAME-SUBFRAME structure in the lattice (repeated-geometry) expression. (See Section 5.5.) The tallied region is located at the spatially different position but belongs to the same lattice cell.

○ TIME-DEPENDENT

Solve a time-dependent problem.

○ WEIGHT-WINDOW

Use the weight-window variance reduction technique. (See Sections 2.9.1 through 2.9.4.)

7.4 Conflict between Options

There are some options which cannot be specified at the same time because of their incompatibility. Table 7.1 shows the incompatible options and default values. If the incompatible options are specified or the same options with and without “NO-” are specified, the last input data is effective.

Table 7.1 Default values and incompatibility for options (1/3)

Option	Default	Incompatible options	Comments
ADAPTIVE-MICRO-CALCULATION	off	EIGEN-VALUE	Only for MVP. See p. 93.
ADJOINT	off		Only for GMVP. See p. 93.
ANALOG-FISSION	off		Only for MVP. See p. 94.
AUTO-RESTART	off		See p. 94.
BETA-EFFECTIVE	off		Only for MVP. See p. 94.
BREMSSTRAHLUNG	off		Only for MVP (with PHOTON option). See p. 95.

(Cont'd)

Table 7.1 Default values and incompatibility for options (2/3)

Option	Default	Incompatible options	Comments
COLLISION-LESS	off		See p. 95.
DELAYED-NEUTRON	on		Only for MVP. See p. 95.
DOPPLER-SCATTERING	NONE		Only for MVP. See p. 96.
DYNAMIC-MEMORY	-		Compilation option. See p. 97.
EDIT-BY-COLLISION	off	EDIT-BY-TRACK-LENGTH	Only for MVP. See p. 98.
EDIT-BY-TRACK-LENGTH	on	EDIT-BY-COLLISION	Only for MVP. See p. 98.
EDIT-MACROSCOPIC-DATA	off	FIXED-SOURCE	Only for MVP. See p. 99.
EDIT-MICROSCOPIC-DATA	off		Only for MVP. See p. 100.
EIGEN-VALUE	off		See p. 100.
FISSION	off		Automatically set to “on” in eigenvalue problem. See p. 100.
FISSION-MULTIPLICITY	TOTAL	EIGEN-VALUE	Only for MVP. See p. 101.
FIXED-SOURCE	on		See p. 102.
FLUX-PRINT	off		See p. 102.
FREE-LATTICE-FRAME	off		Automatically set to “on” in the STGM region. See p. 102.
IMAGINARY-PARTICLE	off	WEIGHT-WINDOW	See p. 102.
IMPORTANCE	off		See p. 103.
LATTICE	off		See p. 104.
(REPEATED-GEOMETRY)			
MONITOR	off		See p. 104.
MULTI-TASK	off		See p. 104.
NEUTRON	on		See p. 105.
PATH-STRETCH	off		See p. 105.
PERIODIC-TIME	off		See p. 106.
PERTURBATION	off		Only for MVP. See p. 106.
PHOTON	off		Detailed model if ‘(...)’ is omitted. See p. 106.
PHOTO-NUCLEAR	off		Only for MVP. See p. 107.
PICTURE	off		See p. 108.
PRINT-SUPPRESS	off		See p. 108.
RELATIVE-WEIGHT	off		See p. 109.
RESPONSE	off		See p. 109.
RESTART	off	WEIGHT-WINDOW	See p. 109.
RESTART-FILE	on		See p. 110.
RUN-MODE	off		See p. 111.
RUSSIAN-ROULETTE	on		See p. 111.
SCATTERING-MATRIX	off		Only for MVP. See p. 112.
SCATTERING-MUBAR	off		Only for MVP. See p. 112.

(Cont'd)

Table 7.1 Default values and incompatibility for options (3/3)

Option	Default	Incompatible options	Comments
SOURCE-OUTPUT	off		See p. 112.
TALLY-LATTICE (FRAME-DEPENDENT- TALLY)	off		See p. 113.
TIME-DEPENDENT	off		See p. 113.
WEIGHT-WINDOW	off	RUSSIAN- ROULETTE, IMPORTANCE	See p. 113.

7.5 Detailed Information for Each Option

We describe the detailed information of options available for MVP/GMVP. The options without “MVP only” or “GMVP only” are available for both MVP and GMVP.

ADAPTIVE-MICRO-CALCULATION

■ Function ■

By default, microscopic cross sections for all nuclides are calculated after particle energy changes. But the cross sections for each nuclide are prepared when they are necessary with this option.

■ Default ■

Not effective.

■ Description ■

This option is effective only in MVP. The microscopic cross sections at a certain energy point are obtained by interpolating the assigned cross section data. This process is performed when the energy of the particle changes by collisions or generation of a new particle. Without the ADAPTIVE-MICRO-CALCULATION option, the microscopic cross sections are prepared for all nuclides that are involved in the materials when a particle is generated or a particle collides with a nuclide.

When this option is specified, the microscopic cross sections for each nuclide are prepared when they are necessary. The user can save the computation time in the case where there are many nuclides in the calculation geometry and the material through which a particle passes does not always contain a large number of nuclides. However, note that the user cannot save the computation time for all cases. For example, it is not recommended to use this option for vector calculations.

If some nuclides are specified to tally microscopic reactions in the TALLY data block, the microscopic cross sections are prepared for the nuclides even if this option is specified.

ADJOINT

■ Function ■

An adjoint problem is solved in GMVP. Currently, the adjoint option is not available in MVP.

■ Default ■

Not effective.

■ Note ■

Note that only the sampled parameter G in the SOURCE data block must be input in reverse order in respect to energy groups. The results are also output in reverse order in respect to energy groups by using this option. The fission source is sampled from the distribution of χ , not $\nu\Sigma_f$.

ANALOG-FISSION**■ Function ■**

The number of fission neutrons is determined by Terrell's equation⁴⁷⁾ in the fixed source problem of MVP.

■ Default ■

Not effective.

■ Description ■

This is used for an option for simulation of the Feynman- α experiment. See Section 4.5.

■ Related options ■

FIXED-SOURCE (p. 102), FISSION (p. 100)

AUTO-RESTART**■ Function ■**

Restart automatically. If there exists a restart file on I/O unit 10, the codes restart a calculation automatically by reading the restart file. Unless there exists a restart file on I/O unit 10, the initial run is performed. The restart file should be output on I/O unit 20 in the previous calculation.

■ Note ■

See also description of the RESTART option. If the user specifies the AUTO-RESTART option and make an input data including NPART < 0, the user can perform the sequential calculations with constant increment of the histories by using the same input data.

■ Related options ■

RESTART-FILE (p. 110), RESTART (p. 109)

■ Related input data ■

NPART (p. 115), NRSINT (p. 118)

BETA-EFFECTIVE**■ Function ■**

This option calculates the effective delayed neutron fraction (β_{eff}) by approximating the adjoint neutron flux with the next fission probability. Three approximations are used for the calculation: non-adjoint weighted method, Meulekamp's method⁵⁴⁾ and Nauchi's method.⁵⁵⁾ See reference 56 for the formulation and the accuracy of the approximations.

■ Default ■

Not effective.

■ Related options ■

To calculate the effective delayed neutron fraction without the approximations, the user should specify BEFF for the tally parameter in the PERTURBATION data block. See Section 15.2.5 for the details.

BREMSSTRAHLUNG**■ Function ■**

This option generates Bremsstrahlung X rays due to electrons. This option is effective for MVP only.

■ Default ■

Not effective.

■ Related options ■

This option is effective only together with the PHOTON (p. 106) option.

■ Note ■

The index file for electron libraries must be input on FORTRAN I/O unit 27 in MVP.

COLLISION-LESS**■ Function ■**

Generated particles from sources fly without collisions by taking account of spatial attenuation.

■ Default ■

Not effective.

■ Description ■

This option is similar to the IMAGINARY-PARTICLE option but can estimate attenuation by scattering or absorption of particles without replacing the composite material with void. Therefore, the option makes it possible to estimate contribution of direct flight from the sources (uncollided flux) to a particle flux or any reaction rate. The modified track length estimator is used as the estimation. (But the output is written as track length estimator in the output list.)

■ Related options ■

EDIT-BY-TRACK-LENGTH (p. 98)

■ Note ■

- Some particles may be trapped in an infinite loop if this option is used for a geometry surrounded with reflective or periodic materials. It is necessary to terminate particle flights by putting a perfect absorption material at the end of them or using the Russian roulette option.
- Output by the collision estimator is meaningless when this option is specified.

DELAYED-NEUTRON**■ Function ■**

Generate delayed neutrons in neutron-induced fission reactions and photo-fission reactions. Prompt and delayed neutrons generated in their fission reactions are addressed separately. This option is effective only for MVP.

■ Default ■

Effective.

■ Related options ■

The option FISSION (p. 100) must be effective if the user wants to generate delayed neutrons from neutron-induced fission reactions. If the option TIME-DEPENDENT (p. 113) is effective, time delay for delayed neutron generation is considered.

■ Related input data ■

WGTF (p. 258), WGTPN (p. 260), PRODUCE-REACTION (p. 242).

DOPPLER-SCATTERING(mode)

■ Function ■

This option selects an elastic scattering model in the resonance region.

■ Format ■

A calculation mode can be specified for MVP as follows.

DOPPLER-SCATTERING (*mode*)

One of the models is selected by a character string *mode*.

DOPPLER-SCATTERING (NONE) : asymptotic slowing-down scattering model (asymptotic model).

DOPPLER-SCATTERING (NORES) : free-monatomic-gas model with constant cross section (constant cross section model).

DOPPLER-SCATTERING (EXACT) : free-monatomic-gas model with resonance cross section (exact model). This mode requires the special MVP libraries which include the elastic scattering cross section at 0 K. The special libraries are prepared only for heavy metals of U-235, -238, Pu-238 to -242 and Am-241. Unless the special libraries are specified for nuclide ID's, the asymptotic model is used for the nuclides.

DOPPLER-SCATTERING (EXACT+) : free-monatomic-gas model with resonance cross section (exact model). The same as above but the constant cross section model is used for the nuclides unless the special libraries are specified for nuclide ID's.

The asymptotic model is selected if “(mode)” is omitted.

■ Default ■

The asymptotic model is used: DOPPLER-SCATTERING (NONE).

■ Description ■

There are 4 modes for resonance elastic scattering. The details are described in Section 4.2.6.

DYNAMIC-MEMORY(size)**■ Function ■**

This option specifies a dynamically allocated memory size required to run a problem.

■ Format ■

DYNAMIC-MEMORY(*size*)

The unit of the memory size is ‘word’ used on the computer. The word is a memory unit to express an integer data in the FORTRAN language.

■ Default ■

The default size is determined in compilation. It can be also changed by specifying a command line parameter “MEMORY=*size*” in running the programs.

■ Note ■

- It is difficult to know the required memory size for a problem to be solved before execution. But once MVP/GMVP is executed, the user can know the accurate memory size required for the problem from the following output.

```

USED DYNAMIC MEMORY SIZE (TASK SHARED) =      4820405 (WORD)

*****
*      MEMORY UTILIZATION SUMMARY      *
*****

Item                Task shared (word)          Task local (word)
X-SEC DATA         4360587 ( 90.5%)           4360587 ( 90.5%)
GEOMETRY DATA       1996 ( 0.0%)              1996 ( 0.0%)
TALLY               14161 ( 0.3%)             14161 ( 0.3%)
EVENT STACK         24304 ( 0.5%)             24304 ( 0.5%)
PARTICLE BANK       311400 ( 6.5%)            311400 ( 6.5%)
WORKING AREA        92418 ( 1.9%)             92418 ( 1.9%)
Others              15539 ( 0.3%)             15539 ( 0.3%)

Total               4820405                    4820405
Available           6600000                    6600000

< Character Memory (4bytes/unit)>

Total               585
Available           60000

```

- Currently, 1 word is 4 bytes on many computers or operating systems but on some computers 1 word is 8 bytes such as a CRAY-MP series. MVP/GMVP assumes either of the following environments for the size of data units.
 - INTEGER=4 bytes, REAL=4 bytes, DOUBLEPRECISION(REAL*8)=8 bytes
 - INTEGER, REAL and REAL*8 are all 8 bytes.

There is a case where the word size is appropriate in the former environment but too large in the latter.

- This option is effective only when the programs are compiled to allocate memories dynamically.
- The information for the allocated memory size is output as follows.

```
**** MEMORY FOR VARIABLE-SIZED DATA :      ???????? WORDS ( 4 BYTES/WORD )
      (      ????.?? MEGA-BYTES )
```

It is not always possible to allocate the specified memory size in some environments of computers. If the memory allocation is failed in, the following message is output.

```
!!! FAILED TO ALLOCATE MEMORY DYNAMICALLY WITH SPECIFIED SIZE ( ???????? WORDS)
```

The possible causes for the failure are as follows.

- Too many processes are working on the computer.
The memory allocation is failed in with large probability when many programs that require much memory are working at the same time. A user should retry after finishing or terminating other processes.
- The upper limit of virtual memory is smaller.
It is common that UNIX-based machines can run programs which require more memory than the machines have by using virtual memory. But such programs may not run in some cases depending on the parameters related to the function (storage of a swap disk, upper limit of virtual memory per process, etc.) In such a case, a user has to consult a system administrator.

EDIT-BY-COLLISION

■ Function ■

The collision estimator is used to calculate microscopic or macroscopic cross sections and reaction rate in MVP.

■ Default ■

Not effective.

■ Related options ■

This option is applied to the contents specified in the EDIT-MICROSCOPIC-DATA (p. 100) , EDIT-MACROSCOPIC-DATA (p. 99) option. It is incompatible with the EDIT-BY-TRACK-LENGTH (p. 98) option.

EDIT-BY-TRACK-LENGTH

■ Function ■

The track length estimator is used to calculate microscopic or macroscopic cross sections and reaction rates in MVP.

■ Default ■

Effective.

■ Related options ■

This option is applied to the contents specified in the EDIT-MICROSCOPIC-DATA (p. 100) , EDIT-MACROSCOPIC-DATA (p. 99) option. It is incompatible with the EDIT-BY-COLLISION (p. 98) option.

EDIT-MACROSCOPIC-DATA(*flag*)**■ Function ■**

Macroscopic cross sections and reaction rates are output to a file or printed in MVP.

■ Default ■

Not effective.

■ Format ■

EDIT-MACROSCOPIC-DATA(*flag*)

■ Description ■

flag is an eight-digit integer.

$$flag = 10^7 \cdot N_1 + 10^6 \cdot N_2 + 10^5 \cdot N_3 + 10^4 \cdot N_4 + 10^3 \cdot N_5 + 10^2 \cdot N_6 + 10 \cdot N_7 + N_8$$

Each digit of the flag shows the type of reactions.

- N_1 : total
- N_2 : nu-fission
- N_3 : fission
- N_4 : elastic scattering
- N_5 : capture (absorption-fission)
- N_6 : inelastic scattering
- N_7 : (n,2n)
- N_8 : absorption-(n,2n)-2(n,3n)-3(n,4n)

Each value of the digit shows the process type.

- 0: Not calculated.
- 1: Calculated. Output to the file (I/O unit 30) and not printed.
- 2: Calculated. Output to the file (I/O unit 30) and only cross sections are printed.
- 3: Calculated. Output to the file (I/O unit 30) and only reaction rates are printed.
- 4: Calculated. Output to the file (I/O unit 30) and the both are printed.

If “*flag*” is omitted, this option is not effective.

■ Related options ■

The user can choose the track length estimator or the collision estimator by specifying EDIT-BY-TRACK-LENGTH (p. 98) or EDIT-BY-COLLISION (p. 98) , respectively.

EDIT-MICROSCOPIC-DATA(*flag*)**■ Function ■**

Averaged microscopic cross sections and reaction rates are printed (or output the binary result file) in MVP. Both are calculated when this option is effective.

■ Default ■

Not effective.

■ Format ■

EDIT-MICROSCOPIC-DATA(*flag*)

■ Description ■

The input format and input data for “*flag*” are the same as the EDIT-MACROSCOPIC-DATA option.

■ Related options ■

The user can choose the track length estimator or the collision estimator by specifying EDIT-BY-TRACK-LENGTH (p. 98) or EDIT-BY-COLLISION (p. 98), respectively.

EIGEN-VALUE**■ Function ■**

Solve an eigenvalue problem. (Calculate a multiplication factor)

■ Default ■

Not effective.

■ Related options ■

This option is incompatible with the FIXED-SOURCE (p. 102) option. The FISSION (p. 100) option is set automatically when this option is effective. Note that the FIXED-SOURCE (p. 102) option is effective by default.

■ Related input data ■

WGTF (p. 258), NHIST (p. 115)

FISSION**■ Function ■**

Generate fission neutrons.

■ Default ■

Not effective.

■ Related options ■

This option is set automatically when the EIGEN-VALUE (p. 100) option is effective.

■ Description ■

The number of generated particles is determined by comparing a weight of produced fission neutrons with a particle weight WGTF specified by each region in collision of a neutron with a fissionable nuclide. The treatment of fission neutrons is different between a fixed-source problem and an eigenvalue problem.

- Fixed-source problem

Track the generated particles in the same batch as the primary (parent) particle. The tracking can be terminated when the number of generations exceeds the limit (MXPGEN) to prevent a flood of neutrons in super-critical systems.

- Eigenvalue problem

The generated particles are not tracked in the current batch and stored in the 'fission neutron bank' temporarily. These stored particles are used as the source of the next batch. Sufficient number of fission neutrons should be stored for the next batch. It is recommended that it should be more than NHIST (the number of histories per batch). Otherwise, correlation between batches (generations) may become large and thus the confidence of evaluation of the eigenvalue may be spoiled. In order to prevent this situation, the user should not set large values to weights of fission neutrons WGTF. In general, it is desired that WGTF should be less than the presumed eigenvalue.

■ Related input data ■

WGTF (p. 258), NHIST (p. 115), MXPGEN (p. 121)

FISSION-MULTIPLICITY

■ Function ■

Option for fission multiplicity (yield). Fission neutrons are generated from total or prompt fission multiplicity. (MVP only)

■ Format ■

FISSION-MULTIPLICITY(TOTAL): Total fission multiplicity is used.

FISSION-MULTIPLICITY(PROMPT): Prompt fission multiplicity is used.

■ Default ■

Total fission multiplicity.

■ Description ■

Prompt and delayed neutrons are emitted in the fission reaction. Total fission neutrons must be considered for the usual criticality calculation. There are some cases where only the prompt fission neutrons are required to evaluate the prompt neutron attenuation factor α in the analysis of noise experiments. This option can be used for such a case. When only the prompt fission neutrons are generated by this option, the number of fission neutrons is sampled on the basis of ν_p instead of ν_t .

■ Related input data ■

FISSION (p. 100).

FIXED-SOURCE

■ **Function** ■

Solve a fixed-source problem (external source problem).

■ **Default** ■

Effective.

■ **Related input data** ■

The user must define fixed sources in the SOURCE data block.

■ **Related options** ■

This option is incompatible with the EIGEN-VALUE (p. 100) option.

FLUX-PRINT

■ **Function** ■

Calculated particle fluxes per lethargy by each region and energy group are printed.

■ **Default** ■

Not effective.

■ **Related options** ■

This option has higher priority than the PRINT-SUPPRESS (p. 108) option.

FREE-LATTICE-FRAME

■ **Function** ■

An arbitrary shape can be specified for a frame of a LATTICE when using the LATTICE (REPEATED-GEOMETRY) option.

■ **Description** ■

This option must be used together with the LATTICE option. When only the LATTICE option is used, the shape of a FRAME ZONE (ZONE where a LATTICE is placed) is restricted. (For a rectangular lattice, the same size of a rectangular parallelepiped. For a hexagonal lattice, a cylinder, a rectangular parallelepiped or a right hexagonal prism.) This option enables us to use the arbitrary shape of the FRAME ZONE.

This option is internally set “ON” when the statistical geometry model (STGM region) is selected.

■ **Related input data** ■

See input of zone shapes in the GEOMETRY data block.

IMAGINARY-PARTICLE

■ **Function** ■

Perform a transport calculation of imaginary particles which do not react with any material and just fly straight from the generation point. (In this case, all regions are assumed to be void.) But treatment of boundary conditions, leakage, etc is the same as real particles.

■ Default ■

Not effective

■ Description ■

The user can check geometry input data by using an appropriate source. In addition, the REGION volumes can be calculated by setting the TALLY-REGION volumes to 1.0 and using the appropriate source such as a parallel beam.

■ Note ■

- It is easier than this option to check the input geometry by using an auxiliary code CGVIEW practically. (See Appendix B for CGVIEW.)
- Some particles may be trapped in an infinite loop by using this option carelessly because they are reflected on the surface of the reflective material.

IMPORTANCE**■ Function ■**

The importance weight is used as a variance reduction technique. Russian roulette kill and splitting are imposed according to importance defined by region and energy group.

■ Default ■

Not effective.

■ Description ■

Suppose the ‘importance’ values are set as follows when a particle moves from a phase space to another phase space.

- I_{old} : Importance value for the past phase
- I_{new} : Importance value for the present phase

then defining the importance ratio as $\xi = I_{new}/I_{old}$,

- For $\xi < 1$:

The Russian roulette game is played with the survival probability ξ .

The weight of the survived particle is set to $W' = W/\xi$.

- For $\xi \geq 1$:

The particle is split into N particles, where N is calculated as follows by defining $[\xi]$ as an integer part of ξ .

$$N = [\xi] + k$$

$$k = 1 \text{ with probability } \xi - [\xi]$$

$$k = 0 \text{ with probability } 1 - (\xi - [\xi])$$

After being split, the particle weight is set to $W' = W/\xi$.

If this option is used together with the RUSSIAN-ROULETTE option, the Russian roulette game is played after the above game.

■ Related options ■

This option is incompatible with the WEIGHT-WINDOW option but compatible with the RUSSIAN-ROULETTE (p. 111) option.

■ Related input data ■

XIMP (p. 252).

LATTICE or REPEATED-GEOMETRY

■ Function ■

Lattice (repeated-geometry) expression is used.

■ Default ■

Not effective.

■ Related input data ■

See IDLAT, NVLAT, KLATT, etc. and the #CELL data sub-block to define zones in the GEOMETRY data block.

MONITOR

■ Function ■

The number and the averaged weights of reactions or events by each region and energy group are monitored. The monitoring events are collision, kill, splitting, etc.

■ Default ■

Not effective

■ Description ■

This is useful to determine parameters for variance reduction. Note that calculation time increases by the use of this option in some cases. (especially for vector computers)

■ Related options ■

WEIGHT-WINDOW (p. 113), IMPORTANCE (p. 103), RUSSIAN-ROULETTE (p. 111).

■ Related input data ■

PSALP (p. 254), PSXYZ (p. 254), WKIL (p. 253), WSRV (p. 253).

MULTI-TASK(n)

■ Function ■

This option specifies the number of processing elements (PEs) in multi-task processing.

■ Format ■

MULTI-TASK (n)

■ Description ■

n is the number of PEs. Usually, it should be specified in command line options 'option NTASK=...'.

NEUTRON

■ Function ■

Solve a neutron transport problem.

■ Default ■

Effective.

■ Note ■

The user must specify the index file for neutron cross section libraries in MVP. If the arbitrary-temperature libraries are used, the corresponding index file must be input on FORTRAN I/O unit 24. If the fixed-temperature libraries are used, the corresponding index file must be input on unit 25. See Section 16.3 for the index file.

■ Related options ■

The PHOTON (p. 106) option must be specified for the neutron- and photon-coupled problem. The 'NO-NEUTRON' option must be specified explicitly for a transport problem of photon only. The 'NEUTRON' option must be specified in an eigenvalue calculation (EIGEN-VALUE (p. 100)).

PATH-STRETCH

■ Function ■

This is a variance reduction technique with path stretching or exponential transformation. The particles are biased so that their flight distance is stretched or shortened. The biased flight distance depends on the direction of the flight. This option can be used to collect particles in the vicinity of the desired spatial point.

■ Default ■

Not effective.

■ Description ■

The details are described in Section 2.9.3. In this variance reduction technique, the total scattering cross section Σ_t is modified such that

$$\begin{aligned}\tilde{\Sigma}_t &= (1 - \beta)\Sigma_t, \\ \beta &= \alpha \boldsymbol{\Omega} \cdot \boldsymbol{\Omega}', \\ \boldsymbol{\Omega}' &= \frac{\boldsymbol{c} - \boldsymbol{r}}{|\boldsymbol{c} - \boldsymbol{r}|}.\end{aligned}$$

The user must then input the parameters α and c which are assigned by input data PSALP and PSXYZ.

■ Related options ■

WEIGHT-WINDOW (p. 113), IMPORTANCE (p. 103), RUSSIAN-ROULETTE (p. 111)

■ Related input data ■

PSALP (p. 254), PSXYZ (p. 254), WKIL (p. 253), WSRV (p. 253)

PERIODIC-TIME

■ Function ■

This option sets the flight time of a particle to 0 when it exceeds the specified time in a time-dependent calculation.

■ Default ■

Not effective.

■ Description ■

When the flight time of a particle exceeds the time cutoff (TCUT) , the particle continues random walk by setting the flight time $t - \text{TCUT}$ in the time-dependent problem (TIME-DEPENDENT option).

Note that the random walk must be terminated by analog absorption, leakage from the system, weight cutoff like Russian roulette etc. except for the time cutoff.

■ Related options ■

TIME-DEPENDENT (p. 113), RUSSIAN-ROULETTE (p. 111), WEIGHT-WINDOW (p. 113)

■ Related input data ■

TCUT (p. 121), WLLIM (p. 122)

PERTURBATION

■ Function ■

Perform perturbation calculations.

■ Default ■

Not effective.

■ Description ■ Currently MVP can calculate the change in the effective multiplication factor (k_{eff}) and kinetics parameters of the effective delayed neutron fraction β_{eff} and the neutron generation time Λ with the perturbation capability. The additional input data defined in the PERTURBATION data block is required for perturbation calculations. See Chapter 15 for the details of the input data.

PHOTON or PHOTON(mode)

■ Function ■

Solve a photon transport problem.

■ Format ■

Calculation mode can be specified in MVP as follows.

PHOTON (*mode*)

Either model is selected by a character string *mode*.

PHOTON (PHOTON) : Detailed model

PHOTON (GAMMA) : Simplified model for high-energy photon transport

The detailed model is selected if “(mode)” is omitted.

■ Default ■

The photon calculation is not carried out by default.

■ Description ■

There are two cases for photon generation. One is generation from photon sources and the other is from reactions between a nucleus and a neutron. In the latter case, the number of photons is decided by comparing a standard particle weight WGTP specified by each region and a total weight of generated photons in the neutron collision.

■ Related options ■

The NEUTRON (p. 105) option must be effective in the neutron/photon-coupled calculation. The 'NO-NEUTRON' option must be specified in the photon transport calculation only. The BREMS-STRAHLUNG (p. 95) option to treat Bremsstrahlung X rays generated by interaction between a photon and a nucleus is available.

■ Related input data ■

WGTP (p. 260).

■ Note ■

- The index file for a photon library must be input on FORTRAN I/O unit 26 in MVP.
- Both element ID in the photon library and nuclide ID in neutron libraries are available in specifying the material composition in an MVP photon calculation.
- No neutron transport calculation is performed for a problem with only photon sources given even if the options are specified for a neutron/photon-coupled problem. (Currently, (γ ,n) reaction cannot be treated.)

PHOTO-NUCLEAR

■ Function ■

Generate secondary particles from photonuclear reactions. This option is effective for MVP only.

■ Default ■

Not effective.

■ Description ■

Photonuclear reactions are treated explicitly and generated neutrons and photons from the reactions are tracked. The photonuclear library for MVP is required to treat the photonuclear reactions. The user must input an index file for the library on FORTRAN I/O unit 28 in executing MVP to use the library. In addition, the user must specify photonuclear reaction cross section ID's in the input data. See Section 9.1.4.

■ Related options ■

This option is effective only if so are the options NEUTRON (p. 105) and PHOTON (p. 106).

■ Related input data ■

Section 9.1.4 Use of Photonuclear Reaction Cross Sections, PNPRODUCE (p. 229), DIMENSION (p. 231), PRODUCE-REGION (p. 240), PRODUCE-NUCLIDE (p. 240), PRODUCE-REACTION (p. 242).

PICTURE

■ Function ■

Cross-sectional views of a geometrical model are output by character patterns.

■ Default ■

Not effective.

■ Description ■

There is the CGVIEW code to draw and check the geometry of MVP/GMVP input data but a user can check it without the graphics environment by using this option.

■ Related input data ■

NPICT (p. 125), PAPER (p. 126).

PRINT-SUPPRESS

■ Function ■

Option to suppress the print for each output subject.

■ Format ■

```
PRINT-SUPPRESS ( flag1 flag2 ... )
```

■ Description ■

If item numbers are specified, their items are not printed .

Items (I_j) that can be suppressed are listed as follows.

- 1: particle flux by each group
- 2: (not defined)
- 3: time-dependent flux
- 4: response
- 5: time-dependent response
- 6: averaged values for source positions, etc. in each batch
- 7: information for neutron multiplication factors in each batch
- 8-16: (not defined)
- 17: microscopic reaction rates (MVP)
- 18: microscopic cross sections (MVP)
- 19: macroscopic reaction rates (MVP)
- 20: macroscopic cross sections (MVP)

RELATIVE-WEIGHT

■ Function ■

Parameters of weights for Russian roulette, weight-window and generation of secondary particles are interpreted as relative values to the initial values of generated particles.

■ Default ■

Not effective.

■ Description ■

If this option is not active, the parameters of WSRV and WKIL for the variance reduction options RUSSIAN-ROULETTE or WEIGHT-WINDOW are compared with the weight of the particles. The parameters WGTF and WGTP which are initial weights for fission neutrons and secondary photons in the fixed source problem are also compared with it.

With the RELATIVE-WEIGHT option, the above parameters (WKIL, WSRV, WGTF and WGTP) are treated as the relative values to the initial weight of the source particle or the secondary particle generated from a reaction.

This treatment is effective in cases where the behavior of descendant particles is important but the initial weight of the particles must be small because it avoids immediate termination of secondary particles by Russian roulette or too small number of generations for descendant particles.

The user can also limit the flight time of a particle (TCUT) and the generation of a particle (MXPGEN) together in case that the calculation is never terminated because the number of descendant particles increases too much.

■ Related options ■

FIXED-SOURCE (p. 102), WEIGHT-WINDOW (p. 113), RUSSIAN-ROULETTE (p. 111)

■ Related input data ■

WKIL (p. 253), WSRV (p. 253), WGTF (p. 258), WGTP (p. 260), TCUT (p. 121), MXPGEN (p. 121)

RESPONSE

■ Function ■

Reaction rates are calculated by using multigroup response functions.

■ Default ■

Not effective.

■ Description ■

The user can estimate reaction rates by using the particle flux and the multigroup response function specified in the RESP data. MVP calculates the reaction rates by using continuous-energy cross section libraries, but the user must use this option to estimate any reaction rate in GMVP.

■ Related input data ■

NRESP (p. 228), RESP (p. 228)

RESTART

■ Function ■

This option shows a restart calculation. MVP or GMVP executes a calculation by reading a restart file on I/O unit 10. It is assumed that the restart file is produced in the previous run (I/O unit 20).

■ Note ■

In general, the user cannot use an MVP or GMVP restart file which is output by the different version of the code. (Strictly speaking, the version of the restart file) In addition, a restart file output from a different machine or on a different OS is not usable even if the version is the same. But if a restart file is output in the same binary format and the internal expression of the data is the same, it may be available. (Sparc CPU for SunOS and PA-RISC for HP-UX etc.)

The total number of histories in the restart calculation depends on the sign of NPART:

- Case of a positive integer : This value is the sum of histories for this run and previous runs.
- Case of a negative integer : The absolute value of NPART is the histories to be calculated in the present run.

For example, when a calculation have been performed for 100,000 histories in a previous run, the number of histories in the present run is as follows.

- $\text{NPART} = 250,000$: the calculation is performed for 100,001 to 250,000 histories. The histories to be calculated are $\text{NPART} - 100,000 = 150,000$.
- $\text{NPART} = -50,000$: the calculation is performed for 100,001 to 150,000 histories. The histories to be calculated in the present run are 50,000.
- $\text{NPART} = 50,000$: No calculation is performed for $0 < \text{NPART} \leq 100,000$. In this case, only the statistical process is performed and the restart file is output for the calculation result of 100,000 histories.

Some users may use the scripts (runmvp, rungmvp) to execute the MVP/GMVP code. In this case, the restart file is specified as follows:

```
runmvp ri=input-restart-file ro=output-restart-file
```

Unless the user specifies the “ro” option, the restart file is output to a scratch file. It is thus erased after the calculation finished. If users need a restart file, it is necessary to specify the “ro” option. The users should not specify the same name for the “ri” and “ro” options.

■ Related options ■

RESTART-FILE (p. 110), AUTO-RESTART (p. 94)

■ Related input data ■

NPART (p. 115), NRSINT (p. 118)

RESTART-FILE

■ Function ■

Output data necessary for restart on I/O unit 20.

■ Default ■

Effective.

■ Note ■

Use this option as “NO-RESTART-FILE” when a restart file is not necessary.

■ Related options ■

RESTART (p. 109), AUTO-RESTART (p. 94)

RUN-MODE[(mode)]

■ Function ■

A special action is specified.

■ Description ■

Currently, the following mode is available.

- *mode* = 1 : Only the input data is processed and anything else (random walk, etc.) are not. This mode can be used to check input data.

RUSSIAN-ROULETTE

■ Function ■

Use the Russian roulette variance reduction technique which terminates random walk of particles with a weight smaller than an assigned value.

■ Default ■

Effective.

■ Description ■

This option plays the Russian roulette game by using the following parameters defined for a phase space (region and energy) when a particle enters the space.

WKIL : Weight value below which the Russian roulette game is played.

WSRV : Weight given to a survived particle in the Russian roulette game. ($WSRV > WKIL$)

Suppose a particle weight is W .

- If $W \geq WKIL$, then the Russian roulette game is not played.
- If $W < WKIL$, then the following treatment is performed according to a random number ξ ($0 \leq \xi < 1$).
 - For $\xi < W/WSRV$, the particle weight is set to WSRV and the particle continues to be tracked.
 - For $\xi \geq W/WSRV$, tracking of the particle is terminated.

■ Related options ■

This option is incompatible with the WEIGHT-WINDOW option but compatible with the IMPORTANCE (p. 103) option. See also RELATIVE-WEIGHT (p. 109) option.

■ Related input data ■

WKIL (p. 253), WSRV (p. 253)

SCATTERING-MATRIX[(order)]**■ Function ■**

This option calculates group-to-group scattering rates (scattering matrix) and scattering matrix moments (scattering matrix tally). “*order*” is the order up to which scattering matrix moments are calculated. Unless “*order*” is specified, only group-to-group scattering rates are calculated (*order*=0). Currently, the maximum order is 7.

■ Default ■

Not effective.

■ Related input data ■

The scattering matrix tally depends on options of EDIT-MICROSCOPIC-DATA (p. 100)(which calculates microscopic cross sections and reaction rates) and EDIT-MACROSCOPIC-DATA (p. 99)(which calculates macroscopic cross sections and reaction rates). Unless the 4th flag (N4) for elastic scattering is specified, scattering matrix tallies for elastic scattering are not output. To calculate the group-to-group microscopic scattering rates, the user must specify the flags of EDIT-MICROSCOPIC-DATA as 1 or larger than 1 for N4(elastic), N6(inelastic) and N7((n,2n)). Likewise, the user must specify the flags of EDIT-MACROSCOPIC-DATA as 1 or larger than 1 for N4(elastic), N6(inelastic) and N7((n,2n)) to calculate the group-to-group macroscopic scattering rates.

The user can select either the track length estimator or the collision estimator for the calculation of the group-to-group scattering rates. The user specify the option EDIT-BY-TRACK-LENGTH (p. 98) and the option EDIT-BY-COLLISION (p. 98) for the track length and the collision estimators, respectively. The default estimator is the track length estimator.

Tally regions for the scattering matrix tallies are the same as for the EDIT-MICROSCOPIC-DATA and EDIT-MACROSCOPIC-DATA options. They are specified in the sub-block of #TALLY REGION at the end of the GEOMETRY data block.

SCATTERING-MUBAR**■ Function ■**

This option calculates the average scattering cosine in the laboratory system (scattering cosine tally).

■ Default ■

Not effective.

■ Related input data ■

Other input data related to the scattering cosine tally is the same as the scattering matrix tally. See the option SCATTERING-MATRIX (p. 112).

SOURCE-OUTPUT**■ Function ■**

Output fission source data at the last batch in an eigenvalue calculation. This data includes the fission sites, energy and so on. This is available for the source of another calculations.

■ Default ■

Not effective.

■ Related options ■

This option is effective only when the EIGEN-VALUE (p. 100) option is specified.

■ Related input data ■

See the SOURCE data block (p. 194) and the chapter of contents of the output files.

TALLY-LATTICE or FRAME-DEPENDENT-TALLY

■ Function ■

Tally by the REGION defined in the FRAME-SUBFRAME structure of the LATTICE (repeated-geometry) expression.

■ Default ■

Not effective.

■ Description ■

Only the tally averaged over all SUBSPACES consisting of the same cell is computed without this option. If this option is on, the user can edit for each SUBSPACE as the different tallied regions.

■ Related input data ■

See the #SUBFRAME and #TALLY REGION data sub-blocks in the GEOMETRY data block.

TIME-DEPENDENT

■ Function ■

Solve a time-dependent problem.

■ Default ■

Not effective.

■ Description ■

Specified this option, the user can estimate time-dependent physics parameters or terminate random walk of particles by time.

■ Related input data ■

NTIME (p. 224), TCUT (p. 121), TALLY data block.

WEIGHT-WINDOW

■ Function ■

Use the weight-window variance reduction technique.

■ Default ■

Not effective.

■ Description ■

Suppose the following values are defined for a phase space (region, energy) which a particle enters.

WKIL : Weight value below which the Russian roulette game is played.

WSRV : Weight given to a survived particle in the Russian roulette game. ($WSRV > WKIL$)

Suppose a particle weight is W ,

- for $W > 2 \cdot WSRV$:

The particle is split into $\xi = W/WSRV$ particles. ξ is not an integer in most cases, so the Russian roulette game is played with probability ($\xi - [\xi]$) as the IMPORTANCE option does. The new particle weight is set to $WSRV$.

- for $W < WKIL$:

The Russian roulette game is played with a probability $W/WSRV$. The weight for the survived particle is set to $WSRV$.

The particle weights are set in the range of $WKIL < W < 2 \cdot WSRV$ by the above procedure.

■ Note ■

The inequality $WSRV > WKIL$ must be satisfied for all phase space.

■ Related options ■

This option is incompatible with the RUSSIAN-ROULETTE (p. 111) and IMPORTANCE (p. 103) options.

■ Related input data ■

WKIL (p. 253), WSRV (p. 253)

8. Calculation Control

8.1 Control Data for Both MVP and GMVP

This chapter describes the input data to control random walk in MVP/GMVP (the number of histories, calculation time limit, physics model, etc.). All the following data must be input in the labeled free format.

NPART

■ Meaning ■

The number of total histories.

■ Description ■

The ‘history’ means a particle life that the particle is generated from a source, takes a random walk and finally is absorbed or leaks outside the system. The particle is sometimes split into more than one particle (splitting) during the random walk and the lives of the generated particles generated by splitting are also included in the history of the original particle. See Section 2.8. Thus, NPART is the number of total particles generated from sources.

- For $\text{NPART} > 0$:

NPART means the number of total histories for a calculation. In a restart calculation, the number of histories in the previous run must be included. The number of histories in the skipped batches before tallying must be included in the case of an eigenvalue calculation.

- For $\text{NPART} < 0$:

In a restart calculation, the absolute value of NPART is the number of histories of the present run. It does not include the number of histories of the previous runs. In an initial calculation, the absolute value of NPART is simply the number of histories.

■ Default ■

No default value.

■ Related input data ■

The number of histories per batch NHIST.

NHIST

■ Meaning ■

The number of histories per batch.

■ Default ■

No default value.

■ Note ■

- In a restart calculation, the same value must be specified as in the previous calculation.
- NPART must satisfy $\text{NPART} \geq \text{NHIST}$, but it is not required that NPART is an integral multiple of NHIST. Even in such a case, MVP/GMVP carries out statistical analysis correctly though the number of histories in the last batch may be smaller than NHIST.

■ Related input data ■

The number of total histories NPART, the bank size of particles NBANK.

NBANK

■ Meaning ■

Length of the particle bank.

■ Description ■

MVP/GMVP stores the multiple-particle data on the memory and processes them at a time in the random-walk process (vector calculation on vector computers). The array to store the attribute data for the multiple particles is called ‘particle bank’. NHIST particles are generated at the beginning of each batch but the number of particles possibly increases in the random-walk by the following processes.

- Generation of fission neutrons in a fixed-source problem
- Generation of photons from neutron reactions and neutrons from photonuclear reactions in a neutron-photon coupled calculation
- Splitting by use of the WEIGHT-WINDOW or IMPORTANCE option

In such cases, it is required that NBANK is larger than NHIST.

■ Default ■

Set to NHIST. But the value is set to $2 \times \text{NHIST}$ in case of a fixed-source problem including fission (FIXED-SOURCE + FISSION options) or use of the WEIGHT-WINDOW or IMPORTANCE option. For a neutron-photon coupled calculation by MVP, these values are multiplied with $1/(1-\text{BANKP})$.

■ Related input data ■

The number of total histories NPART, the number of histories per batch NHIST.

NGROUP

■ Meaning ■

The number of energy groups.

■ Description ■

It is the total number of energy groups for both neutron and photon.

- For GMVP:

NGROUP is the number of energy groups used for random walk in multigroup cross section data. The first group is the highest energy one. NGROUP is not always the same as the number of energy groups in the data but NGROUP must be equal to or less than it. If NGROUP is less than

the number of energy groups in the data, particles are terminated when they are down-scattered below the energy of NGROUP (energy cutoff).

- For MVP:

NGROUP is the number of the energy bins to tally particle fluxes, etc. The energy bin structure is also used for multigroup response functions and the energy-dependent parameters for variance reduction. The user can specify the energy structure. The number of energy groups for both neutron (NGROUP.N) and photon (NGROUP.P) must be input for a neutron-photon coupled calculation. The number of total energy groups NGROUP is calculated automatically in the codes.

$$\text{NGROUP} = \text{NGROUP.N} + \text{NGROUP.P}$$

For either neutron- or photon-only problem,

- When NGROUP is input, NGROUP.N (or P) is set to NGROUP.
- When NGROUP.N (or P) is input, NGROUP is set to NGROUP.N (or P).

Either NGROUP or NGROUP.N (or P) must be input.

NGROUP.N[NEUTRON] or NGP1

■ Meaning ■

The number of neutron energy groups. The characters in [] can be omitted. NGP1 can be also specified instead of NGROUP.N.

■ Related input data ■

See the description of NGROUP.

NGROUP.P[PHOTON] or NGP2

■ Meaning ■

The number of photon energy groups. The characters in [] can be omitted. NGP2 can be also specified instead of NGROUP.P.

■ Related input data ■

See the description of NGROUP.

NTIME

■ Meaning ■

The number of time bins for time-dependent tallies (effective for the TIME-DEPENDENT option).

■ Default ■

The default value is unity.

■ Note ■

This value can be specified in the TALLY data block.

NRSINT

■ Meaning ■

Interval of the restart-file output.

■ Description ■

In the first version of MVP/GMVP, a restart file is produced only when the calculation terminates normally. Therefore, once the calculation is terminated abnormally by the computers, no restart file is produced and the result obtained by that time is lost in vain. By setting NRSINT to a non-zero value, a restart file is produced even in the halfway of the calculation by regular interval of batches or histories. Of course, a restart file is also produced at the normal end of calculation.

- NRSINT>0 : A restart file is output every NRSINT batches.
- NRSINT<0 : A restart file is output every NRSINT histories, after all the histories in the corresponding batch finish.
- NRSINT=0 : A restart file is output only at the end of a whole calculation.

■ Default ■

The default value is 0. The restart file is produced at the end of calculation.

NBPINT

■ Meaning ■

Interval of batches to output the monitor information.

■ Description ■

CPU time consumed, the averaged value of source parameters, the result of the eigenvalue problem in progress, etc. are output at the end of every batch by default. If NBPINT is set to larger than one, the results are output every NBPINT batches. This function is used to reduce the printing output for a large calculation with many batches.

■ Default ■

The default value is 1. The monitored information is output every batch.

NTMINT

■ Meaning ■

Interval of batches for monitoring output in the eigenvalue problem.

■ Description ■

MVP/GMVP calculates k_{eff} values by combined estimators (see p. 39). By specifying this data, the user can output the k_{eff} values obtained with the combined estimators every NTMINT batches.

■ Default ■

The default value is 10.

TCPU**■ Meaning ■**

The limit of CPU time. (Unit : minutes)

■ Description ■

If the consumed CPU time exceeds TCPU at the end of a batch, then the calculation is terminated and the result at this time is output. Setting TCPU=0, there is no limitation for the CPU time. In parallel calculations, TCPU is the limit of CPU time for each processing element; namely the calculation is terminated if the CPU time for one of the processing elements exceeds TCPU. Unless the source programs of MVP/GMVP are compiled so that the CPU time is monitored, this specification is not effective.

■ Default ■

The default value is 0. No CPU time is checked.

IRAND**■ Meaning ■**

Seed of an initial random number. It must be an integer (from 0 to $2^{31} - 1$). This value is not used for a restart calculation which is initiated by the random number given in a restart file.

■ Default ■

The default value is 0.

NFBANK**■ Meaning ■**

The number of particles which can be stored in the fission neutron bank for an eigenvalue problem.

■ Description ■

Generated fission neutrons are stored in the array for the source of the next batch. The array to store the data is called 'fission neutron bank'. NFBANK must be equal to or greater than NHIST.

■ Default ■

Set to NHIST.

■ Related input data ■

See the description of WGTF.

NSKIP**■ Meaning ■**

The number of batches skipped before starting to tally particle flux, etc. in an eigenvalue problem. It is desired to start to tally after the fission source distribution converges well. This value depends on the initial guess for the fission source distribution and the convergence characteristics (dominance ratio) of the calculation model.

■ Default ■

The default value is 0, but several batches must be skipped at least for most eigenvalue problems.

■ Note ■

This value must not be changed in a restart calculation. NSKIP is set to 0 internally in the codes for fixed-source problems.

NMEMO**■ Meaning ■**

The size of array to memorize possible ZONES which a particle enters from each ZONE (per ZONE).

■ Description ■

NMEMO is the size per ZONE to memorize ZONES which are found in the ZONE-search process. The processing time can be reduced by searching from the memorized ZONES. It is preferable to be the maximum number of possible ZONES or more.

■ Default ■

The default value is 5.

NMEMOP**■ Meaning ■**

The codes learn the probabilities that a particle enters next ZONES. The order of the ZONES in the memory array is re-ordered in the first \sim NMEMOP'th batches to reduce computation time.

■ Default ■

The default value is 2.

NMEMS**■ Meaning ■**

The number of memorized ZONES where particles are born from each source set.

■ Description ■

The generation points from the source are sampled from each source set given in the SOURCE data block. But it is required to find the ZONES where particles are generated to begin random walk. The processing time can be reduced by memorizing the found ZONES for each source set and searching from the memorized ZONES.

■ Default ■

The default value is 5.

NRSKIP**■ Meaning ■**

The number of skipped random numbers at the beginning of a calculation or a batch.

■ Description ■

For $\text{NRSKIP} > 0$, NRSKIP random numbers are skipped only once before a calculation. For $\text{NRSKIP} < 0$, |NRSKIP| random numbers are skipped at the beginning of each batch.

■ Default ■

The default value is 0.

DEPS**■ Meaning ■**

The minimum distance resolved in the codes (cm).

■ Description ■

When a particle passes over a boundary, MVP/GMVP determines an entering ZONE by searching for a ZONE that includes a position at the DEPS distance away from the boundary along the flight direction. It also means that the accuracy limitation is DEPS for the geometrical data.

■ Default ■

The default value is 10^{-5} cm.

DINF**■ Meaning ■**

The distance that the codes treat as infinity (cm).

■ Description ■

When the codes calculate the distance from the current position of a particle to a ZONE boundary along the flight path, DINF is used to check whether the flight path crosses the BODY boundary or not. If the flight path is larger than DINF, it is treated as infinite.

■ Default ■

The default value is 10^{30} cm.

TCUT**■ Meaning ■**

The limit of particle age tracked in random walks. The random walk for a particle with the age exceeding TCUT is terminated. This option is effective when the TIME-DEPENDENT option is used.

■ Default ■

Set to the last value of the TIMEB data.

MXPGEN**■ Meaning ■**

The limitation of generations for the descendant particles in the fixed-source problem. If MXPGEN is specified, no descendant particles beyond the MXPGEN'th generation are generated. Setting MXPGEN=0, there is no limitation for the descendant particles.

■ Description ■

Neutron histories may not be terminated in a fixed-source problem where fissionable materials exist and the system is super-critical or near critical. MXPGEN can be used to terminate the histories.

■ Default ■

No limitation for the descendant particles (MXPGEN = 0).

WLLIM

■ Meaning ■

Parameter not to generate the particles whose weight are less than WLLIM in the process of variance reduction or generation of secondary particles. This parameter is used to suppress generating more particles than necessary when using the RELATIVE-WEIGHT option etc.

■ Default ■

The default value is 10^{-10} .

ELOOP

■ Meaning ■

Parameter to control particles that may enter in the endless loop during random walk.

■ Description ■

When ELOOP is a positive value, MVP/GMVP terminates particle tracking in each batch by detecting the "endless loop" as follows.

- If the number of survival particles is less than 1% of the source particles or less than 3 in each batch, the code starts detecting particles which might be in the endless loop.
- If the ratio of the number of events after starting detection to the number of all events is ELOOP or larger, the code terminates the survival particles forcibly after it outputs the following message and information of the following events 29 times.

```
XXX    n  PARTICLES ARE PROBABLY IN ENDLESS LOOP.
(MONITOR 30 EVENTS)
```

This process does not always detect the endless loop of random walk correctly. It cannot detect the case where almost all particles in a batch enter the endless loop simultaneously. In another case, important information may be lost by terminating a few particles that continue random walk for a long time. In the latter case, ELOOP should be a larger value or 0.

■ Default ■

The default value is 0.2.

8.2 Control Data Only for MVP

ETOP[.N]

■ Meaning ■

The upper limit of neutron energy calculated in MVP (eV).

■ Description ■

If the energy of a neutron generated by fission is over ETOP.N, it is set to ETOP.N. If ETOP.N is set as $\text{ETOP.N} > \text{ETOPL}$ (upper energy limit of MVP libraries : usually 20 MeV), it is set to ETOPL automatically in the code. “.N” can be omitted.

■ Default ■

The default value is 2×10^7 eV (20 MeV).

EBOT[.N]

■ Meaning ■

The lower limit of neutron energy calculated in MVP (eV).

■ Description ■

If EBOT.N is set as $\text{EBOT.N} < \text{EBOTL}$ (lower energy limit of MVP libraries : 10^{-5} eV), it is set to EBOTL automatically. “.N” can be omitted. This value is set to EBOTL independently of the input value for an eigenvalue problem or a problem with thermal scattering ($\text{ETHMAX} \neq 0$). For these problems, neutron energy is set to EBOT.N if it is less than EBOT.N.

■ Default ■

The default value = 10^{-5} eV.

ETOP.P

■ Meaning ■

The upper limit of photon energy calculated in MVP (eV).

■ Description ■

If the generated-photon energy is over ETOP.P, it is set to this value. If ETOP.P is set as $\text{ETOP.P} > \text{ETOPLP}$ (upper energy limit of photon libraries : 100 MeV), it is set to ETOPLP automatically in the code.

■ Default ■

The default value is 1×10^8 eV (100 MeV).

EBOT.P

■ Meaning ■

The lower limit of photon energy calculated in MVP (eV).

■ Description ■

If EBOT.P is set as $EBOT.P < EBOTLP$ (lower energy limit of photon libraries : 1 keV), it is set to EBOTLP automatically in the code.

■ Default ■

The default value 1×10^3 eV (1 keV).

ETHMAX

■ Meaning ■

The upper energy limit for thermal neutron scattering in MVP (eV).

■ Description ■

If $ETHMAX \neq 0$, thermal scattering is taken into consideration. But if $ETHMAX \leq EBOT.N$, it is automatically set as $ETHMAX = ESABL$ (the highest one of upper energy limits ($E_{S\alpha\beta}$) given in MVP libraries for the thermal inelastic scattering data of nuclides used in the code). In the energy range $E_{S\alpha\beta} < E \leq ETHMAX$, the short-collision time approximation or the free gas model is applied to thermal scattering.

■ Default ■

The default value is 4.5 eV.

AMLIM

■ Meaning ■

The upper limit of nuclide mass below which the free gas model is applied. This value must be input as a relative value to the neutron mass (1.00867 amu).

■ Description ■

The thermal scattering with a nucleus of heavier than AMLIM is treated as isotropic elastic scattering in the center-of-mass system above 0.625 eV. In the energy region below 0.625 eV, the neutron is scattered isotropically in the center-of-mass system without energy loss.

■ Default ■

The default value is 1000.0. The free gas model is applied to all nuclides.

EW CUT

■ Meaning ■

The energy boundary between analog absorption and non-analog absorption in MVP (eV).

■ Description ■

Analog absorption which terminates the history with the absorption probability is applied below the energy of EW CUT and weight reduction (non-analog absorption : the particle weight is reduced by the absorption probability instead of termination) is applied above it.

■ Default ■

The default is 275 eV.

BANKP**■ Meaning ■**

The probability to generate the secondary photons and kill the primary neutrons for a neutron-photon coupled calculation in MVP in a case where the particle bank is full.

■ Description ■

If there is no room in the finite particle bank (NBANK) to store a photon generated by a neutron reaction, either the generated photon or the parent neutron is stored in the bank and the other is killed (terminated) by Russian roulette. BANKP is the survival probability of the generated photon for the Russian roulette process.

■ Default ■

The default value is 0.5.

MB**■ Meaning ■**

The maximum number of materials whose macroscopic cross sections are memorized by each particle in MVP.

■ Description ■

Microscopic cross sections are calculated for all nuclides in MVP whenever the particle energy changes in collisions. Macroscopic cross sections are calculated from microscopic cross sections of constituent nuclides for a given particle energy in MVP. But the macroscopic cross section for a material is required only when the particle is in the zone which contains this material. It is possible that the particle passes through some materials before next collision or leakage, and some materials may be the same along the flight path. Thus, the arrays ('macroscopic cross section bank') to memorize the macroscopic cross sections of MB materials are prepared in MVP to reduce computation time of the cross sections.

The user can save the memory storage to set MB to a smaller value than NMAT(the number of materials) in the case where a system includes multiple materials and a particle does not pass through all the materials during one flight.

■ Default ■

Set to the number of materials for which the compositions are specified (NMAT).

8.3 Input Data for PICTURE Option**NPICT****■ Meaning ■**

The number of cross-sectional views drawn by character patterns from an input geometry. This value is effective when the PICTURE option is specified.

■ Default ■

The default value is 0.

■ Related input data ■

Specify with the PAPER data.

PAPER

■ Format ■

PAPER($V_{x1} V_{y1} V_{z1} A_{x1} A_{y1} A_{z1} B_{x1} B_{y1} B_{z1} N_{x1} N_{y1} CELL-ID_1$
 $V_{x2} V_{y2} V_{z2} A_{x2} A_{y2} A_{z2} B_{x2} B_{y2} B_{z2} N_{x2} N_{y2} CELL-ID_2$
 \dots
 $V_{xN} V_{yN} V_{zN} A_{xN} A_{yN} A_{zN} B_{xN} B_{yN} B_{zN} N_{xN} N_{yN} CELL-ID_N$)

where $N = \text{NPICT}$.

■ Meaning ■

Data to specify the points and ranges of cross-sectional views for the PICTURE option.

- $V_{xi} V_{yi} V_{zi}$
Coordinates of the lower left corner of the i -th cross-sectional view.
- $A_{xi} A_{yi} A_{zi}$
Horizontal vector of the i -th cross-sectional view (x -axis).
- $B_{xi} B_{yi} B_{zi}$
Vertical vector of the i -th cross-sectional view (y -axis).
- N_{xi}
Number of meshes in the x -direction.
- N_{yi}
Number of meshes in the y -direction.
- $CELL-ID_i$
CELL ID to be checked. Specify zero if unnecessary.

Let $\mathbf{V}_i = (V_{xi}, V_{yi}, V_{zi})$, $\mathbf{A}_i = (A_{xi}, A_{yi}, A_{zi})$ and $\mathbf{B}_i = (B_{xi}, B_{yi}, B_{zi})$. The material and REGION numbers are output at \mathbf{X}_i :

$$\mathbf{X}_i = \mathbf{V}_i + \frac{j - \frac{1}{2}}{N_{xi}} \mathbf{A}_i + \frac{k - \frac{1}{2}}{N_{yi}} \mathbf{B}_i. \quad (8.1)$$

Then, the following items are checked for each position.

- Are multiple REGION numbers or material numbers defined in the same ZONE?
- Are there undefined ZONES?

If there are some inconsistencies or errors, the code stops execution after detailed messages are output. The CGVIEW code should be used to draw more accurate cross-sectional views. (See Appendix B.)

■ Related input data ■

Specify with NPICT.

9. Cross Section and Material Data

9.1 Input Data for MVP

Basically the user just specifies constituent nuclides and isotopic number densities of each material for MVP. It is not necessary to prepare a form of macroscopic cross sections because the MVP code uses the continuous-energy cross section data from its own library.

The user can also specify the temperature of each material. Since cross sections depend on temperature, the data must be prepared at each required temperature. MVP automatically generates the cross section data at arbitrary temperatures from the data in a reference library.

Furthermore, the user can produce the user's own temperature-dependent cross section library with the MVP-ART code. When multiple calculations are performed with the same cross section data, the computation time can be saved. (See Appendix C.)

9.1.1 CROSS SECTION or XSEC Data Block

Data to specify compositions of materials must be defined within the data block from a character string “\$CROSS SECTION” or “\$XSEC” to “\$END CROSS SECTION” or “\$END XSEC” in MVP.

```
$CROSS SECTION
  [ TPRECS( temperature-precision ) ]
  [ JVPOP( doppler-broadening-in-vector-mode ) ]
  [ JXPOOL( read-write-doppler-broadened-data-in-pool ) ]
& IDMAT( material-ID )
  [TEMPMT( material-temperature ) ]
  nuclide-ID( number-density [temperature] )
  nuclide-ID( number-density [temperature] )
  ....
& IDMAT( material-ID )
  [TEMPMT( material-temperature ) ]
  nuclide-ID( number-density [temperature] )
  nuclide-ID( number-density [temperature] )
  ....
&
...
$END CROSS SECTION
```

9.1.2 Specification of Compositions (IDMAT and Number Density)

The composition data for each material must be input according to the following rules.

1. The line whose first column is “&” is the beginning of the composition data for each material. The second column must be a blank. Then, the ID number of a material IDMAT must be input in the labeled free format.

& IDMAT (*material-ID*)

“*material-ID*” is any positive integer specified by the user. The ID numbers are not necessary to be sequential, but the different number must be assigned for different material. The ID numbers are used to specify ZONE materials in the geometrical data input.

2. Nuclide ID and atomic number density must be input in the label free format as follows.

nuclide-ID (*number-density*)

This input is repeated by the number of constituent nuclides for the material. The order of the input data is arbitrary.

- *nuclide-ID* :

This is a character string which shows a nuclide name. In the case of a photon calculation, this is not ‘nuclide’ ID but ‘element’ ID. The length of the character string is allowed to be up to 16 characters. The typical pattern for neutron cross sections is

zzmmmcxxx[. . .],

where

- zz : symbol of an element like “PU”, “HE”, etc. (For the symbol of one character, “H0”, “P0”, etc.)
- mmm : atomic mass number expressed with 3 digits. “000” means an element with natural abundance.
- c : index to specify isomers or molecules (See Section G.1.)
- xxx : evaluated nuclear data library (J40 for JENDL-4.0, B70 for ENDF/B-VII.0, F31 for JEFF-3.1 etc.).
- [. . .] : identifier for special library if any (ORT and PAR for spin isomers of hydrogen, U1 for JENDL-4.0 update 1, TZ for nuclear data including 0 K elastic scattering cross section).

The nuclide ID (or element ID) specified here is not the file name of the corresponding cross section but the alias linked to the cross section file. The link between each alias and the cross section file is described in Section 16.3. Therefore, the user can give any alias to the nuclide ID in the index file unlike the alias defined above. But the element ID numbers for photon cross sections are identified from the first two characters of the alias in the neutron-photon coupled problem, so the above rule must be obeyed for the first two characters of the nuclide ID numbers.

- *number-density* : number density (10^{24} atom/cm³)

3. Each material data is terminated by a line which begins with “&” (beginning of the next material composition) or “\$END CROSS SECTION” or “\$END XSEC” (end of input for all materials).

Example:

```
$CROSS SECTION
& IDMAT(100)
  U02350J40(6.189E-4)  U02380J40(2.383E-2)
```

```

      000160J40( <2*(6.189E-4+2.383E-2)> )
& IDMAT(110)
      H0001HJ40(6.6856E-2) 000160J40( <0.5*6.6856E-2> )
$END CROSS SECTION

```

9.1.3 Specification of Nuclide ID in Photon Problems

If the user solves a neutron-photon coupled problem and the photonuclear reactions are neglected (default), the user just specifies neutron nuclide ID's only. The photoatomic cross sections are identified by the first two characters of the nuclide ID (corresponding to the symbol of the element) and read from the photoatomic cross section library. Similarly, the electron cross sections are also identified and read from the electron library when the BREMSSTRAHLUNG option is specified.

If the user solves a photon-only problem and the photonuclear reactions are neglected, both nuclide ID's in the neutron library and element ID's in the photon library are available for specification of elements. In this case, no neutron cross section data are used even if nuclide ID's in the neutron library is specified.

See the following section for the specification of nuclide ID's if the photonuclear reactions are addressed.

9.1.4 Use of Photonuclear Reaction Cross Sections

The user must specify which photonuclear reaction cross sections are used for a problem considering photonuclear reactions. The photonuclear reaction cross sections must be specified in the following format.

$zzmmmcxxx-zzmmmeeee(\textit{number-density})$
--

zzmmmcxxx is a nuclide ID, zzmmmeeee is a photonuclear nuclide ID, and *number-density* is the atomic density (10^{24} atom/cm³). zzmmmeeee is specified as follows.

- zz : Element name of 2 characters (add "0" for 1-character name).
- mmm : Atomic mass number with 3 digits for a neutron library. "000" means an element with natural abundance.
- eeee : Identifier of an evaluated nuclear data library. Currently, the following 5 identifiers are available. JEPN: Evaluation by JENDL (Japan), KAPN: Evaluation by KAERI (Korea), CNPN: Evaluation by CNDC (China), LAPN: Evaluation by LANL (USA), BOPN: Evaluation by BOFOD (Russia).

The photonuclear nuclide ID is used to search the cross section data file in the photonuclear index file. If a natural element is specified, each isotope for the element is treated internally with the abundance. If a photonuclear nuclide ID is specified but its cross section data file is not included in the photonuclear library, the photonuclear reaction is not considered for the nuclide; namely, no photonuclear reactions occur for the nuclide. If the PHOTO-NUCLEAR option is specified and photonuclear nuclide ID's are omitted, MVP presumes the photonuclear nuclide ID's from the first 5 characters of the nuclide ID's.

Example:

```

& IDMAT(5)
PB2060J40-PB206JEPN(7.94293E-03)
PB2070J40-PB207JEPN(7.28376E-03)
PB2080J40-PB208JEPN(1.72701E-02)

```

9.1.5 Use of Arbitrary-temperature Neutron Cross Sections

The index file for the arbitrary-temperature neutron cross section libraries must be input on I/O unit 24. The MVP code reads the reference cross sections from it and produces the cross sections at any temperature by Doppler broadening and interpolation.⁵⁷⁾ Fixed-temperature libraries generated with the MVP-ART code are also available together with the arbitrary-temperature ones but the index file for the former libraries must be input on I/O unit 25. Unless the material or nuclide temperature is specified, the fixed-temperature cross sections are used. If the arbitrary-temperature cross sections are used, the following parameters can be additionally specified.

1. Precision of temperature specification (TPRECS (p. 131)) and the option of vector-processing for Doppler broadening (JVPOP (p. 131)).

These parameters must be specified between “\$XSEC ”(or “\$CROSS SECTION”) and the first “&” line if necessary.

2. Temperature for each material or each nuclide.

If a temperature is specified explicitly for a material with TEMPMT (p. 133) or temperatures are specified for each nuclide (Section 9.2), the reference cross sections for all constituent nuclides of the material or the specified nuclides are read according to the index file on I/O unit 24 and the cross sections at the specified temperature are calculated in MVP.

3. To keep the Doppler broadened cross section data (JXPOOL (p. 132)).

When there are lots of energy points in a cross section data, the Doppler broadening process is time-consuming. To save the processing time, MVP has a function to read the Doppler broadened cross section data saved in the previous calculation.

The following is examples for the specification of arbitrary-temperature cross section data. The same cross sections are generated in both the examples.

Example 1:

```
$CROSS SECTION
& IDMAT( 1 )
  U02350J40( 5.070E-5 1500.0 )
  U02380J40( 1.685E-2 1500.0 )
  O00160J40( 4.150E-2 1500.0 )
& IDMAT( 2 )
  NA0230J40( 2.180E-2 800.0 )
$END CROSS SECTION
```

Example 2:

```
$CROSS SECTION
& IDMAT( 1 )
  TEMPMT( 1500.0 )
  U02350J40( 5.070E-5 )
  U02380J40( 1.685E-2 )
  O00160J40( 4.150E-2 )
& IDMAT( 2 )
```



```

TEMPMT( 800.0 )
NA0230J40( 2.180E-2 )
$END CROSS SECTION

```

9.2 Options for MVP Cross Section Data

TPRECS

Specify the precision of resolved temperature for the arbitrary-temperature cross sections in MVP.

■ Format ■

```

$CROSS SECTION or $XSEC
...
TPRECS( precision )
...
& IDMAT( material-ID )

```

■ Description ■

This parameter must be specified between the \$XSEC (or \$CROSS SECTION) line and the first “&” line for MVP. The unit is degree. Different temperatures within TPRECS are treated as the same in MVP. In a case where a lot of temperature points are specified for a nuclide, the cross sections at different temperatures are taken as the different nuclides and therefore a lot of CPU time to process them and large memory storage are required. A user can avoid them by specifying TPRECS appropriately. For example, setting TPRECS to 1.0, the cross sections at 560.0 K and 560.3 K are treated as the same one at 560 K. The round-off process for temperature is performed as follows:

$$T = \text{int} \left(\frac{T_0}{\Delta T} + \frac{1}{2} \right) \times \Delta T,$$

where T_0 is an input temperature, ΔT is TPRECS and $\text{int}(x)$ is the maximum integer not exceeding x .

■ Default ■

1.0 degree.

JVPOP

This option specifies whether the vectorized routine for Doppler broadening is used to generate the cross sections at any temperature in MVP.

■ Format ■

```

$CROSS SECTION or $XSEC
...
JVPOP( flag )
...
&

```

■ Description ■

The vectorized routine is used if JVPOP \neq 0. In general, the Doppler-broadening process is fast for vector computers with the vectorized routine, but not for scalar computers with it. Therefore, this option is meaningful only for scalar computation.

■ Default ■

The default value depends on computers. (See the FAT tool and the description of the control data in appendix.) JVPOP is set to 1 for vector computers (CRAY MP, FACOM VPP, NEC SX, HITACHI S series) and 0 for scalar computers.

JXPOOL

Flag to save Doppler broadened cross section data as files after an MVP calculation finishes or reuses them for another calculation.

■ Format ■

```
$CROSS SECTION or $XSEC
...
JXPOOL( flag )
...
&
```

■ Description ■

When MVP cross section data for a nuclide are given at a lot of energy points, it takes a large amount of time to Doppler broaden the cross section data. To save the processing time, MVP has a function to keep Doppler broadened cross section data after an MVP calculation finishes or reuse them for another calculation. To use this function, the following requirements must be met.

- The path to a directory where Doppler broadened cross section data are saved must be defined in the environment variable MVP_ARTPOOL. The path must end with a slash(/).
- The value of JXPOOL is not 0 (default).

Even if MVP_ARTPOOL is defined, the above function does not work in the case of JXPOOL=0.

The cross section data for each nuclide is saved as a file created in the directory defined by MVP_ARTPOOL. The file is named a nuclide ID followed by temperature (for example, U02350J401500 for U-235 cross sections of JENDL-4.0 at 1,500 K). To use the saved cross section data, the user just specify MVP_ARTPOOL and JXPOOL \neq 0. Since the MVP code searches and reads the files, the user does not have to prepare an index file for the saved data.

■ Default ■

1

JPNLCP

If photonuclear reactions are treated in an MVP calculation, this option specifies whether or not production cross sections for charged particles are removed from photonuclear reaction cross sections that are read in.

■ Format ■

```

$CROSS SECTION or $XSEC
...
JPNLCP ( flag )
...
&

```

■ Description ■

If JPNLCP = 0, the production cross sections are not removed. If JPNLCP = 1 and the production cross sections exist in the read-in photonuclear reaction cross sections, they are removed. MVP needs very large memories for photonuclear reaction cross section data together with neutron cross section data in a calculation including photonuclear reactions. If the production cross sections for charged particles are included in the photonuclear reaction cross sections, they usually occupy more than half of the photonuclear reaction cross sections. MVP can save the memories required for the photonuclear calculation by removing the production cross sections. This option is effective only if the production particles from photonuclear reactions are limited to neutrons and photons.

■ Default ■

JPNLCP = 0. MVP does not remove the production cross sections for charged particles.

■ Related options ■

This option is effective only if so is the option PHOTO-NUCLEAR (p. 107).

TEMPMT

Temperature of a material at which the cross sections are generated.

■ Format ■

```

$CROSS SECTION or $XSEC
...
& IDMAT ( material-ID )
    TEMPMT ( material-temperature )
    nuclide-ID ( density [temperature] ) ...

```

■ Description ■

If a temperature of a material is specified by TEMPMT (Kelvin), the cross sections of nuclides in the material are generated from the reference cross sections by Doppler broadening and interpolation. If no positive value is specified for TEMPMT, the fixed-temperature cross sections are used.

■ Default ■

Set to 0. Fixed-temperature cross sections are used.

■ Note ■

The user must generate fixed-temperature cross sections and the corresponding index file. The index file for arbitrary-temperature cross sections can be used as the one for fixed-temperature cross sections. In this case, cross sections at the reference temperature (293.15 K for most nuclides) are used.

Specification of Temperature for Each Nuclide

The user can specify the use of arbitrary-temperature cross sections for each nuclide in the material composition data.

■ Format ■

nuclide-ID (density [temperature])

■ Description ■

If a value following a number density is given, it specifies the temperature for the nuclide in the material. In addition, if the value is positive (unit : Kelvin), MVP generates the cross sections at the specified temperature. This temperature is effective even if different TEMPMT is specified for the material. On the other hand, if the value is not positive, then the fixed-temperature cross sections are used even if TEMPMT is specified for the material. This expression enables a user to use only the fixed-temperature cross sections for some nuclides in the material whose temperature is specified by TEMPMT.

■ Default ■

Set to TEMPMT. But if no TEMPMT is specified or TEMPMT is not positive, it is set to 0. Fixed-temperature cross sections are used.

9.3 Input Data for GMVP

9.3.1 Cross Section Form

The GMVP code can read the following forms of multigroup cross section data.

- The form of the ANISN code⁵⁸⁾ or the similar form: the scattering angular distribution is expressed by the Legendre expansion. (P_L cross sections)
- The double differential cross section form (DDX): this form is used in the MORSE-DDX code.²⁸⁾

For the P_L cross sections, GMVP can read the following four formats. The first two are the binary format and the data must be input on an appropriate I/O unit. The third and last are the text-image format and the data is described in the input data.

- (1) ANISN binary format.

This is the standard cross section format of the ANISN code. The binary data must be input.

- (2) ANISN group-ordered binary format.

The same format as the standard format of the ANISN code but the cross sections are written in the different record in the every energy group. (Binary) The MAIL code⁵⁹⁾ in the JACS system outputs this format of multigroup cross sections.

- (3) ANISN free-field FIDO format.⁶⁰⁾ (Text-image)

Though data content itself is the standard ANISN type, the cross sections must be input as text image embedded in the input data. The text image is similar to the FIDO format of the ANISN code.

(4) DTF-IV format.⁶¹⁾ (Text-image)

The ℓ -th order Legendre expansion coefficients are given but do not include $(2\ell + 1)$. Namely, the ANISN-type coefficients are obtained by multiplying the DTF-type ones by $(2\ell + 1)$. This form can be input as text image embedded in input data.

9.3.2 CROSS SECTION Data Block

It is the CROSS SECTION data block to designate the cross sections and material composition used by the GMVP code.

```

$CROSS SECTION
[ TITLE(  ``title-string' '  ) ][ TYPE(  PL or DDX  ) ]
[ PLFORM(  ANISN or ANISN-MAIL or DTF or FIDO  ) ]
[ NGP1(  ngp1 [nds1] ) ][ NGP2(  ngp2 [nds2] ) ]
[ ITBL(  itbl ) ][ IHT(  iht ) ][ ISGG(  isgg ) ]
[ INGP(  ingp ) ]
[ NPL(  npl ) ][ NSCT(  nsct ) ]
[ INCELM(  incelm ) ]
[ PRFLAG(  irdsg istr ifmu imom iprin ipun ) ]
[ INPUT(  ixtape ) ][ OUTPUT(  jxtape ) ]
[ MUS(  mus ) ]
[ XLAST(  id sigt sigpr siga sigf nu-sigf
          id2 sigt2 .....  ) ]
& IDMAT(  material-ID )
  COMPOSITION(  element-ID density element-ID density ... )
  [ FKAI(  fission-spectrum-data ) ]
  ....
#
any-input-data
$END CROSS SECTION

```

The data between the “\$CROSS SECTION” line and the first “&” line specify the common attribution of the input cross-section library and each material. Each material composition must be specified from the line whose first column is “&”.

When the cross section data and etc. are not input from a binary file, the line initiated by “#” must be placed after the input of the material composition. Then, the cross section data and etc. are input between the line and the “\$END CROSS SECTION” line. In this part, labeled data and arithmetic operations, etc. cannot be used.

9.3.3 Input Parameters of Multigroup Cross Sections

In the CROSS SECTION data block of GMVP, the data specified between the “\$CROSS SECTION” line and the line initiated by “&” for the first material are the common data to each material and are the form of input cross sections etc.

(1) Title of Cross Section Data

Title of cross section input data. The blank character string is given when it is omitted.

■ Format ■

TITLE(``title-string'')

(2) TYPE – Type of Cross Section Data

Specify the cross section form: P_L cross sections or DDX.

■ Format ■

TYPE(PL)
or TYPE(DDX)

■ Default ■

P_L cross sections

(3) PLFORM - Form of Cross Section Data

The form of P_L cross sections (or input form of the data).

■ Format ■

PLFORM(ANISN) or
PLFORM(ANISN-MAIL) or
PLFORM(FIDO) or
PLFORM(DTF)

■ Description ■

- PLFORM(ANISN):

The standard cross sections in the ANISN format must be input as binary data. The cross section data is input on the I/O unit specified in INPUT. When INPUT is not specified, it is input on I/O unit 1 (recommended).

- PLFORM(ANISN-MAIL):

The cross section data must be input in the ANISN-type format output by the MAIL code⁵⁹⁾ in the JACS system. The data must be binary data. This form is different from the standard cross section form of the ANISN code in that the scattering matrix is described in each record by every energy group.

- PLFORM(FIDO):

This is also a standard ANISN format. The cross section data must be placed following the line initiated by “#” after the input of the material composition data and be input in the text form similar to the FIDO format. Following data must be given for every material/nuclide (element) and the order of each Legendre coefficient (0 ~ NPL).

1. Character string “***” must be placed in columns 2 and 3.

2. The length (ITBL) of the cross sections must be described by each energy group for all energy groups. The cross section values can be specified by repetition of identical data, etc which follows FIDO format of the ANISN code. In general, numeric data divided by one blank or more may be specified between the first and the 72nd column.
3. In this input form, the line of which the first column is ' (one-fold quotation mark) can be used as a “comment” line because it is disregarded.

In this input, “the element number” is determined in the order of described data. (The same element number is given to the data of the different order Legendre cross sections for the same element.) The user cannot specify the element number arbitrarily.

Example: Input the cross section data for 2 elements of the number of energy groups = 2, ITBL = 5, NPL = 1.

```
$CROSS SECTION
  TYPE(PL) NPL(1) PLFORM(FIDO)
& IDMAT(1)  /***** FUEL
  COMPOSITION( 1 1.0 ) FKAI( 1.0 0.0 )
& IDMAT(2)  /***** CLAD SUS
  COMPOSITION( 2 1.0 )
/**** Text image cross section data follows after ``#``
#
'===== Element 1 =====
' === PL order 0 =====
**
'   --- energy group 1
    0.7 0.1 0.5 0.1 0.0
'   --- energy group 2
    1.0 0.3 0.6 0.05 0.2
' === PL order 1 =====
**
'   --- energy group 1
    0.0 0.0 0.0 -0.1 0.0
'   --- energy group 2
    0.0 0.0 0.0 -0.05 1.0E-1
'
'===== Element 2 =====
' === PL order 0 =====
**
'   --- energy group 1
    0.5 0.0 0.4 5.0E-2 0.0
'   --- energy group 2
    1.5 0.0 0.1 0.9 0.2
' === PL order 1 =====
**
'   --- energy group 1
    0.0 0.0 0.0 -0.01 0.0
'   --- energy group 2
    0.0 0.0 0.0 -0.1 0.3
$END CROSS SECTION
```

- PLFORM(DTF):

This is the DTF-IV format. It is similar to the standard cross section type used in the ANISN code but the Legendre expansion coefficients are not multiplied by $(2\ell + 1)$. The cross sections can be input in the text form after the # line which is placed after the material composition data. The following data must be specified by each material/element and each order of Legendre expansion:

1. Character string up to 48 characters. The string may explain the cross section data and so on.
2. Repeat 6 numerical values of 12-digit per line for all energy groups. The data length of each energy group is ITBL. This format corresponds to “(6E12.0)” in the READ statement in FORTRAN.

■ Default ■

ANISN binary format (PLFORM(ANISN)).

(4) NGP1

Number of energy groups for primary particles in GMVP.

■ Format ■

NGP1 (*NGP1*) or
NGP1 (*NGP1 NDS1*)

■ Description ■

A primary particle is the one that the data is allocated at the section where the energy group number is small in the multigroup cross section table for more than two kinds of particles. Normally 1 ~ NGP1 groups are allocated for neutron as the primary particle and NGP1+1 ~ NGP1+NGP2 groups are for gamma ray as the secondary particle. (“Secondary” means that the gamma ray is generated from the (n,γ) reaction.) NGP2 is the number of energy groups for the secondary particle. NGP1 is used for the neutron or photon cross sections only and just indicates the number of energy groups for neutron or photon.

When NDS1 is specified, it is possible to reduce the number of energy groups for down-scattering used for the calculation from those specified in the library. The number of energy groups for down-scattering including self-scattering must be specified for NDS1.

■ Default ■

NGP1 is set to NGROUP.N (NGP1) (p. 117) defined in the control data. If both NGP1 in the CROSS SECTION data block and NGROUP.N in the control data are defined, their values must be the same. If neither NGP1 nor NGROUP.N is specified, a fatal error occurs. Unless NGROUP.N is specified in the control data, NGP1 must be defined in the CROSS SECTION data block.

(5) NGP2

Number of energy groups for secondary particles in GMVP.

■ Format ■

NGP2 (*NGP2*) or
NGP2 (*NGP2 NDS2*)

■ Description ■

A secondary particle is the one that the data is allocated at the section where the energy group number is large in the multigroup cross section table for more than two kinds of particles. Normally 1 ~ NGP1 groups are allocated for neutron as the primary particle and NGP1+1 ~ NGP1+NGP2 groups are for gamma ray as the secondary particle. (“Secondary” means that the gamma ray is generated from the (n, γ) reaction.) NGP1 is the number of energy groups for the primary particle. In the case of neutron or photon cross sections only, no NGP2 should be specified or NGP2=0.

When NDS2 is specified, it is possible to reduce the number of energy groups for down-scattering used for the calculation from those specified in the library. The number of energy groups for down-scattering including self-scattering must be specified for NDS2.

■ Default ■

- Single-particle problems.

NGP2=0.

- Neutron-photon coupled problems.

NGP2 is set to NGROUP.P (NGP2) (p. 117) defined in the control data. If both NGP2 in the CROSS SECTION data block and NGROUP.P in the control data are defined, their values must be the same. If neither NGP2 nor NGROUP.P is specified, a fatal error occurs. Unless NGROUP.P is specified in the control data, NGP2 must be defined in the CROSS SECTION data block.

(6) ITBL

Data length of the ANISN/DTF-type cross sections for one energy group in GMVP.

■ Format ■

ITBL (*ITBL*)

■ Description ■

The ANISN/DTF-type cross sections for each energy group have a data structure shown in Table 9.1. The value of ITBL is given by input. It must match the input cross section data.

Table 9.1 Data structure of the ANISN/DTF-type cross sections for each energy group

Position	Cross sections for energy group g
1 ... IHT-3	Activation cross section (optional)
IHT-2	σ_a (Absorption cross section)
IHT-1	$\nu\sigma_f$ (Production cross section)
IHT	σ_t (Total cross section)
IHT+1 ... ISGG-1	$\sigma^{g+i \rightarrow g} (i > 0)$ (Up-scattering cross section)
ISGG	$\sigma^{g \rightarrow g}$ (Self-scattering cross section)
ISGG+1 ... ITBL	$\sigma^{g-i \rightarrow g} (i > 0)$ (Down-scattering cross section)

■ Default ■

In the case of P_L cross sections for the ANISN/DTF-type, no default value is defined and thus the appropriate value must be specified. In the case of DDX, it is not required to input this value because it is determined automatically from the number of energy groups.

(7) IHT

Position of the total cross section for an energy group in the ANISN/DTF-type cross sections.

■ Format ■

IHT (IHT)

■ Description ■

The ANISN/DTF-type multigroup cross sections have the data structure shown in Table 9.1. This input data is IHT in this table. IHT=3 for no activation cross section.

■ Default ■

IHT=3. (No optional activation cross section.)

(8) ISGG

Position of the self-scattering cross section for an energy group in the ANISN/DTF-type cross sections.

■ Format ■

ISGG (ISGG)

■ Description ■

The ANISN/DTF-type multigroup cross sections have the data structure shown in Table 9.1. This input data is ISGG in this table. ISGG=IHT+1 for no up-scattering cross section.

■ Default ■

ISGG = ITBL - NDS1 + 1, where NDS1 is the number of energy groups for down-scattering (including self-scattering). NDS1 is the same as NGP1 unless NDS1 is specified.

(9) INGP

Number of energy groups for which multigroup cross sections are to be input.

■ Format ■

INGP (INGP)

■ Note ■

If cross section data for all energy groups stored in a multigroup library are not required for a GMVP calculation, INGP enables the user to specify the number of energy groups to be input from the library.

■ Default ■

INGP=NGP1+NGP2.

(10) NPL

Order of Legendre expansion for GMVP multigroup cross sections.

■ Format ■

NPL (*NPL*)

■ Description ■

The order of Legendre expansion used in the GMVP calculation should be specified. The input multigroup library may include higher order data than the input order.

■ Default ■

NPL=0.

(11) NSCT

Number of scattering angles in the scattering reaction of GMVP.

■ Format ■

NSCT (*NSCT*)

■ Description ■

In the case of DDX cross sections, the number of meshes (bins) for the cosine of the scattering angle must be specified since the scattering probabilities are provided for the meshes.

In the case of P_L cross sections, the particles are scattered into the specified discrete direction in the scattering process when the expansion order is one or more. The scattering directions and its probabilities are determined from the discrete distribution derived from the Legendre coefficients. Normally, NSCT should be $(NPL+1)/2$. This input is ignored when the expansion order is 0 (isotropic scattering).

■ Default ■

20 for DDX, $[(NPL+1)/2]$ for P_L cross sections. $[x]$ denotes the greatest integer which does not exceed x .

(12) INCELM

When the ID number of nuclides or materials depends on the expansion order of the P_L cross sections, the ID number of the higher order is defined by incrementing the ID number of the 0th order by this value.

■ Format ■

INCELM (*INCELM*)

■ Description ■

The ANISN-type P_L cross sections provided by any codes are available in GMVP unlike MVP which requires the special cross section libraries. GMVP uses the ID numbers for materials/nuclides to recognize the cross sections in the input process. The ANISN-type cross sections for GMVP are supposed to be the following structure:

```

< Repeat for all materials >
  < Repeat for the Legendre coefficients >
    Record 0 : N1, N2, N3, N4, Title
    Record X : Cross section data
  < End of repetition >
< End of repetition >

```

GMVP reads the above N4 as a recognition number and uses the value for the 0th order coefficients as the element ID for the following input of material compositions.

There are some cases where the ID number (N4) for the higher order coefficients may be different from that for the 0th order in some platform. For such a cross section data,

$$\langle \text{N4 for the 0th order} \rangle + \text{INCELM} \times \langle \text{expansion order} \rangle$$

should be used as the recognition number. This specification cannot be used in the case where the recognition number is not linear with the expansion order.

■ Default ■

INCELM=0.

(13) PRFLAG

Specification for the standard output to check the input multigroup cross sections.

■ Format ■

PRFLAG(*irdsg istr ifmu imom iprin ipun*)

■ Description ■

The following subjects are output by setting the value to more than 0.

- 1) *irdsg* : Print the cross sections as read.
- 2) *istr* : Print the cross sections stored in the memories after processing.
- 3) *ifmu* : Print the process where the scattering angles are calculated from the P_L cross sections.
- 4) *imom* : Print the moments calculated from the P_L cross sections.
- 5) *iprin* : Print the scattering angles and their probabilities.
- 6) *ipun* : Print results of bad Legendre coefficients.

■ Default ■

0 for all items. No checked information is printed.

(14) INPUT

I/O unit to input the GMVP multigroup cross sections.

■ Format ■

INPUT (*INPUT*)

■ Description ■

The user can specify the FORTRAN I/O unit for DDX cross sections or binary cross sections for PLTYPE(ANISN), PLTYPE(ANISN-MAIL) etc. But it is not desirable for the user to specify it because GMVP uses lots of I/O units. Without the special reason, the default value should be used.

When a negative value is given, GMVP reads the GMVP original form of the cross section file from the I/O unit. The file is output on the I/O unit defined by OUTPUT() in another calculations.

■ Default ■

1 for P_L cross sections, 99 for DDX.

(15) OUTPUT

I/O unit to output the GMVP original form of the multigroup cross sections which is used internally in GMVP.

■ Format ■

OUTPUT (*OUTPUT*)

■ Description ■

This is an option to save the computation time in calculating with the same cross section data. But it is almost meaningless for the recent high-performance computing.

■ Default ■

Not defined.

(16) MUS

Meshes (bins) of the cosine of the scattering angle in the DDX cross sections.

■ Format ■

MUS (*mu_1 mu_2 mu_3 ... mu_NSCT+1*)

■ Description ■

For DDX cross sections, the scattering probabilities are given to the angular meshes (bins) into which the cosine of the scattering angle is divided but the boundary values for the meshes are not given in the library. The (NSCT+1) boundary values must be input in the incremental order from -1.0 to 1.0.

■ Default ■

Boundary values of the range divided equally into (NSCT) bins from -1 to 1 .

(17) XLAST

Replace the cross sections of the lowest energy group in the DDX cross sections.

■ Format ■

```
XLAST ( id sigt sigpr siga sigf nu-sigf
        id2 sigt2 ..... )
```

The total, production, absorption, fission, nu-fission cross sections must be input after ID of the cross section data.

■ Default ■

Not defined.

9.3.4 Specification of Material Compositions

The style of input data for GMVP material composition is similar to that for MVP. Each material composition data must be input in the following rules.

1. The data whose first column is “&” indicates the beginning of each material composition and the second column must be a blank. In this section, the ID number IDMAT should be specified in the labeled free format as follows.

```
IDMAT ( material-ID )
```

material-ID is an arbitrary positive integer. It is unnecessary to be the sequential number for each material but the different number must be used. This ID number is used to specify the material of “ZONE” in the geometry input section.

2. COMPOSITION: Constitutive nuclide ID's and its atomic density.

The pairs of the nuclide or material ID in the multigroup cross sections and its atomic density must be input in the labeled free format as follows.

```
COMPOSITION ( element-ID number-density
              element-ID number-density
              ....
              )
```

element-ID is the recognition number of a nuclide or a material in the cross section data. The recognition number is explained in INCELM (See Section 9.3.3 (12)). It must be the recognition number for the 0th order data in the case of the P_L cross sections.

number-density is the atomic density in units of 10^{24} atoms/cm³.

3. FKA: Fission spectrum.

The fission spectrum for this material can be specified. It must be input in the labeled free format as follows.

FKAI (χ_1 χ_2 ... χ_{NPG1})
--

χ_i is the probability that neutrons of the i -th group are emitted from fission.

FKAI can be specified in the outside of the CROSS SECTION data block for all materials. However, the fission spectrum can be given to each material by specifying FKAI here. Unless FKAI is specified for a material, all values of the fission spectrum data for the material are 0.

4. The input of the composition data for each material ends with a line which includes the next "&" (beginning of the next material composition data) or "\$END CROSS SECTION".

10. Geometry Data

10.1 Structure of Geometry Data

Geometry data must be input in the GEOMETRY data block between the “\$GEOMETRY” line and the “\$END GEOMETRY” line. The input format is as follows.

```

$GEOMETRY
Specification of LATTICE geometry (for use of LATTICE geometry)
END (for use of LATTICE geometry)

BODY input (geometrical object to define ZONES)

END

Specification of ZONES, their REGION names and materials

#CELL
Definition of ZONES belonging to a unit cell of a LATTICE (for use of
LATTICE geometry)
#END CELL

#TALLY REGION
Definition of combinatorial regions to tally
$END GEOMETRY

```

The specification of LATTICE geometry and the definition of ZONES involved in a unit cell of a LATTICE are required only when the LATTICE option is effective. The CELLS must be defined in the data sub-block between “#CELL” and “#END CELL”. Tally regions are not always necessary to be defined for tallying (after “#TALLY REGION”). The user can specify these data if necessary.

10.2 Definition of BODIes

BODIes must be specified in the labeled free format as follows:

```
BODY-type ( BODY-ID data data ... )
```

where *BODY-type* is a character string to show a BODY geometry, *BODY-ID* a BODY ID number to be referred in the ZONE specification, and *data data ...* to define a BODY size and location. In principle, the unit for the geometry data is cm. Table 10.1 shows the BODIes available in MVP/GMVP and the data to be specified.

For example,

RPP(20000 <-W/2> <W/2> <-W/2> <W/2> <D1> <D1+H>)

shows the BODY ID of 20000 and a right parallelepiped with the faces perpendicular to the coordinate axes. The ranges in the x -, y - and z -directions are from $\langle -W/2 \rangle$ to $\langle W/2 \rangle$, from $\langle -W/2 \rangle$ to $\langle W/2 \rangle$ and from $\langle D1 \rangle$ to $\langle D1+H \rangle$, respectively.

As the other example,

CYL(1202 < A> 0 <D1> <HD> <R>)

shows the BODY ID of 1202 and a cylinder with the central axis parallel to the z -axis. The central coordinate of the base is $x = \langle A \rangle$, $y = 0$, $z = \langle D1 \rangle$, the height is $\langle HD \rangle$ and the radius is $\langle R \rangle$.

Table 10.1 Available BODies in MVP/GMVP (1/2)

BODY	Geometry	Data (excluding BODY ID)
RPP	Right parallelepiped	X_{\min}, X_{\max} : range of x -axis Y_{\min}, Y_{\max} : range of y -axis Z_{\min}, Z_{\max} : range of z -axis
CYL	Cylinder along the z -coordinate	V_x, V_y, V_z : coordinates at the center of one base h : height (a negative value is allowed) R : radius
RCC	Right circular cylinder	V_x, V_y, V_z : coordinates at the center of one base H_x, H_y, H_z : height vector R : radius
SPH	Sphere	V_x, V_y, V_z : coordinates at the center R : radius
BOX	Parallelepiped	V_x, V_y, V_z : coordinates at one of the vertices $A1_x, A1_y, A1_z$: leg vector 1 $A2_x, A2_y, A2_z$: leg vector 2 $A3_x, A3_y, A3_z$: leg vector 3
WED or RAW	Wedge or Right angle wedge	V_x, V_y, V_z : coordinates at one of the vertices $A1_x, A1_y, A1_z$: one of the leg vectors (1) $A2_x, A2_y, A2_z$: one of the leg vectors (2) $A3_x, A3_y, A3_z$: one of the leg vectors (3) (length, width and height vectors)
ARB	Arbitrary polyhedron with 4, 5 or 6 faces	$V1_x, V1_y, V1_z$: coordinates at one of the vertices (1) $V2_x, V2_y, V2_z$: coordinates at one of the vertices (2) $V3_x, V3_y, V3_z$: coordinates at one of the vertices (3) $V4_x, V4_y, V4_z$: coordinates at one of the vertices (4) $V5_x, V5_y, V5_z$: coordinates at one of the vertices (5) $V6_x, V6_y, V6_z$: coordinates at one of the vertices (6) $V7_x, V7_y, V7_z$: coordinates at one of the vertices (7) $V8_x, V8_y, V8_z$: coordinates at one of the vertices (8) $S1, S2, S3, S4, S5, S6$: 4-digit integers to describe the 6 faces
HAF	Half space with planer surface	N_x, N_y, N_z : an outward vector normal to surface A : distance from the surface to the origin ($A < 0$ if the origin is inside the half space)

(Cont'd)

Table 10.1 Available BODIes in MVP/GMVP (2/2)

BODY	Geometry	Data (excluding BODY ID)
RHP	Right hexagonal prism along z -axis	V_x, V_y, V_z : coordinates at the center of one base h : height W : width of the base
HEX	Right hexagonal prism	V_x, V_y, V_z : coordinates at the center of one base H_x, H_y, H_z : height vector W : width of the base A_x, A_y, A_z : normal vector to one of side faces
RCL	Right cylinder with specified radial direction	V_x, V_y, V_z : coordinates at the center of one base H_x, H_y, H_z : height vector R : radius A_x, A_y, A_z : vector to specify the base direction
TRC	Truncated right cone	V_x, V_y, V_z : coordinates at the center of one base H_x, H_y, H_z : height vector $R1, R2$: radii of the lower and upper bases
TEC	Truncated elliptical cone	V_x, V_y, V_z : coordinates at the center of one base H_x, H_y, H_z : height vector A_x, A_y, A_z : radius vector of the major (or minor) axis B_x, B_y, B_z : radius vector of the minor (or major) axis r : ratio of one base to the other
ELL	Spheroid	$V1_x, V1_y, V1_z$: coordinates of a focus $V2_x, V2_y, V2_z$: coordinates of the other focus D : length of the major axis
GEL	General ellipsoid	V_x, V_y, V_z : coordinates at the center $A1_x, A1_y, A1_z$: axis vector (1) $A2_x, A2_y, A2_z$: axis vector (2) $R1, R2, R3$: lengths of the three half axes
ELT	Elliptic tours	V_x, V_y, V_z : center of the hole N_x, N_y, N_z : vector defining the axis of revolution R_m : distance to the center of the ellipse $R1$: major (or minor) radii of the ellipse $R2$: minor (or major) radii of the ellipse τ : angle of the $R1$ direction from the plane normal to the revolution axis (degree)
GQS	General quadratic curved surface	S_{xx}, S_{yy}, S_{zz} : coefficients of the second order terms S_{xy}, S_{yz}, S_{zx} : coefficients of the second order cross terms ($x * y, y * z, z * x$) S_x, S_y, S_z : coefficients of the first order terms C : constant term
BBC	BODY-by-BODY combination	BODY1 : ID of BODY 1 to be combined BODY2 : ID of BODY 2 to be combined BODY3 : ID of BODY 3 to be combined : :

10.2.1 BODY RPP (Right Parallelepiped)

Right parallelepiped with surfaces perpendicular to the coordinate axes. See Fig. 10.1 for the definition of an RPP.

■ Format ■

RPP(ID d1 d2 d3 d4 d5 d6)

■ Parameter ■

ID : BODY ID

d1, d2 : $\min(d1, d2) = X_{\min} \leq x \leq X_{\max} = \max(d1, d2)$

d3, d4 : $\min(d3, d4) = Y_{\min} \leq y \leq Y_{\max} = \max(d3, d4)$

d5, d6 : $\min(d5, d6) = Z_{\min} \leq z \leq Z_{\max} = \max(d5, d6)$

■ Description ■

If $d1 = d2$ or $|d1 - d2| \geq \text{DINF}$ (distance treated as infinite in the codes), the right parallelepiped has the infinite side along the x -axis. The similar treatment is done for the y , z -axes. If only one coordinate of the right parallelepiped is bounded, the BODY shows the region between a pair of parallel planes.

If the BODY is used to define FRAME ZONES (ZONES where a LATTICE is arranged), X_{ℓ} -, Y_{ℓ} - and Z_{ℓ} -axes in the LATTICE-coordinate system are set to be parallel to the x -, y - and z -axes, respectively.

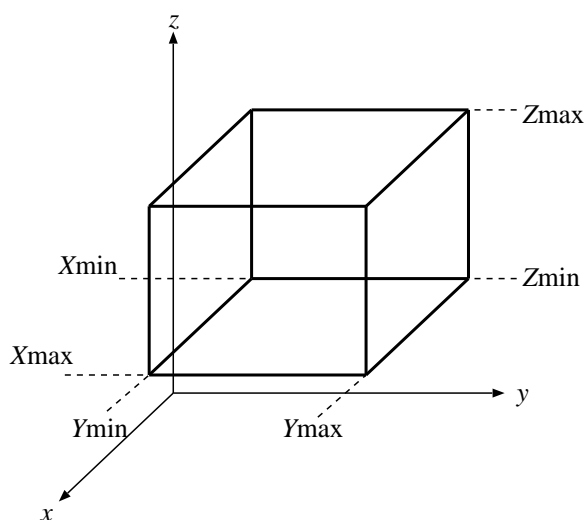


Fig. 10.1 BODY RPP

10.2.2 BODY CYL (Cylinder in z -direction)

Right circular cylinder with the central axis parallel to the z -axis. See Fig. 10.2 for the definition of a CYL.

■ Format ■

CYL(*ID d1 d2 d3 d4 d5*)

■ Parameter ■

ID : BODY ID

d1, d2, d3 : coordinates $\mathbf{V}(V_x, V_y, V_z)$ at the center of one base

d4 : height h

d5 : radius R

■ Description ■

If the height is set to $d4 \geq \text{DINF}$ (distance treated as infinite in the codes) or $d4 = 0.0$, then the cylinder has the infinite height.

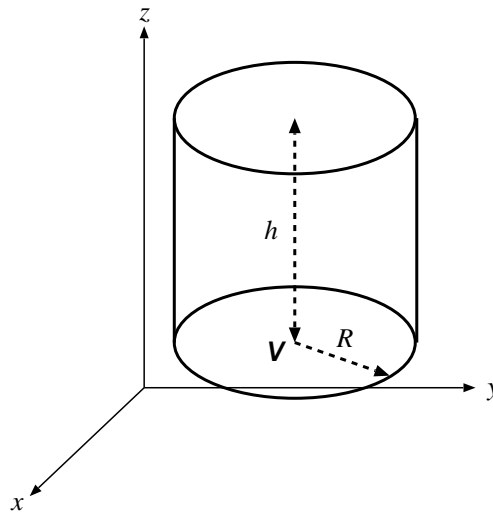


Fig. 10.2 BODY CYL

10.2.3 BODY RCC (Right Circular Cylinder)

General right circular cylinder. See Fig. 10.3 for the definition of an RCC.

■ Format ■

RCC(*ID d1 d2 d3 d4 d5 d6 d7*)

■ Parameter ■

ID : BODY ID

d1, d2, d3 : coordinates $\mathbf{V}(V_x, V_y, V_z)$ at the center of one base

d4, d5, d6 : height vector $\mathbf{H}(H_x, H_y, H_z)$

d7 : radius R

■ Description ■

The coordinates $(d1 + d4, d2 + d5, d3 + d6)$ correspond to the center of the upper base. If the length of the vector \mathbf{H} ($\sqrt{d4^2 + d5^2 + d6^2}$) is greater than DINF (distance treated as infinite in the codes), the cylinder has the infinite height and no bases.

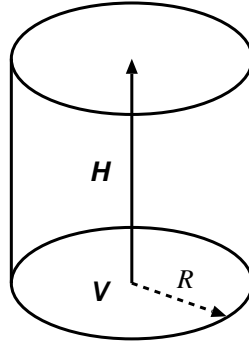


Fig. 10.3 BODY RCC

10.2.4 BODY SPH (Sphere)

Sphere. See Fig. 10.4 for the definition of a SPH.

■ Format ■

SPH(ID d1 d2 d3 d4)

■ Parameter ■

ID : BODY ID

d1, d2, d3 : coordinates $\mathbf{V}(V_x, V_y, V_z)$ at the center

d4 : radius R

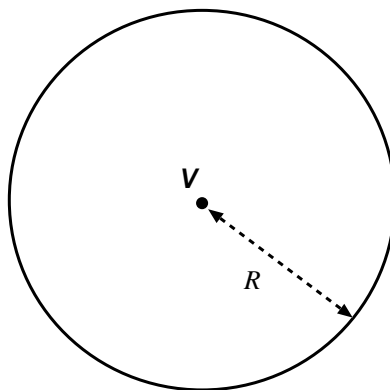


Fig. 10.4 BODY SPH

10.2.5 BODY BOX (Parallelepiped)

General parallelepiped. See Fig. 10.5 for the definition of a BOX.

■ Format ■

BOX(ID d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12)

■ Parameter ■

ID : BODY ID

d1, d2, d3 : coordinates $V(V_x, V_y, V_z)$ at one of the vertices

d4, d5, d6 : leg vector $A1(A1_x, A1_y, A1_z)$

d7, d8, d9 : leg vector $A2(A2_x, A2_y, A2_z)$

d10, d11, d12 : leg vector $A3(A3_x, A3_y, A3_z)$

■ Description ■

The three leg vectors are not necessary to be perpendicular to one another.* But they must be perpendicular when it is used for the definition of FRAME ZONES. In addition, the direction of vector $A3$ must be that of $A1 \times A2$; vectors $A1$, $A2$ and $A3$ construct the right-handed Cartesian coordinate system. The X_ℓ -, Y_ℓ - and Z_ℓ -axes of the LATTICE-coordinate system are set to be parallel to the $A1$, $A2$ and $A3$ directions, respectively.

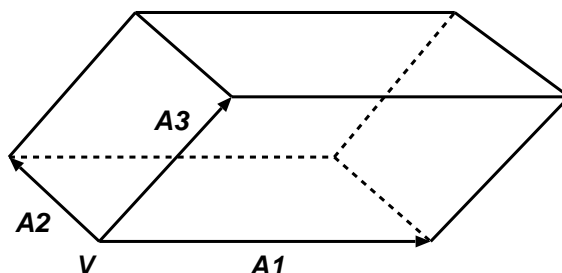


Fig. 10.5 BODY BOX

10.2.6 BODY WED or RAW (Wedge)

Wedge. See Fig. 10.6 for the definition of a WED.

■ Format ■

WED(ID d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12)

RAW(ID d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12)

■ Parameter ■

*BOX indicates a rectangular parallelepiped in the MORSE-CG code, etc.

ID : BODY ID

d1, d2, d3 : coordinates $V(V_x, V_y, V_z)$ at one of the vertices

d4, d5, d6 : one of the leg vectors $A1(A1_x, A1_y, A1_z)$

d7, d8, d9 : one of the leg vectors $A2(A2_x, A2_y, A2_z)$

d10, d11, d12 : one of the leg vectors $A3(A3_x, A3_y, A3_z)$

■ Description ■

The wedge consists of two leg vectors $A1, A2$ of the base and one height vector $A3$. The three vectors need not to be perpendicular to one another.

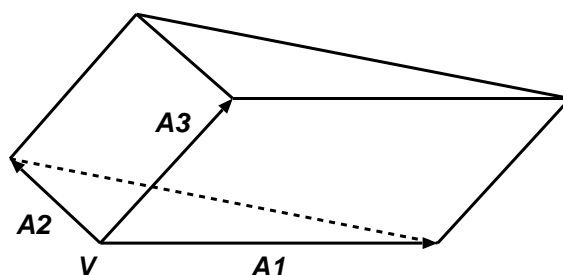


Fig. 10.6 BODY WED (or RAW)

10.2.7 BODY ARB (Arbitrary Polyhedron with 4,5 or 6 Faces)

Arbitrary polyhedron with 4, 5 or 6 faces. See Fig. 10.7 for the definition of an ARB.

■ Format ■

```
ARB( ID d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12
      d13 d14 d15 d16 d17 d18 d19 d20 d21 d22 d23 d24
      d25 d26 d27 d28 d29 d30 )
```

■ Parameter ■

ID : BODY ID

d1, d2, d3 : coordinates $V1(V1_x, V1_y, V1_z)$ at one of the vertices

d4, d5, d6 : coordinates $V2(V2_x, V2_y, V2_z)$ at one of the vertices

d7, d8, d9 : coordinates $V3(V3_x, V3_y, V3_z)$ at one of the vertices

d10, d11, d12 : coordinates $V4(V4_x, V4_y, V4_z)$ at one of the vertices

d13, d14, d15 : coordinates $V5(V5_x, V5_y, V5_z)$ at one of the vertices

d16, d17, d18 : coordinates $V6(V6_x, V6_y, V6_z)$ at one of the vertices

d19, d20, d21 : coordinates $V7(V7_x, V7_y, V7_z)$ at one of the vertices

d22, d23, d24 : coordinates $V8(V8_x, V8_y, V8_z)$ at one of the vertices

d25, d26, d27, d28, d29, d30 : 4-digit integers to describe the 6 faces

■ Description ■

d25, d26, d27, d28, d29, d30 are 4-digit integers giving four vertices of a face. Each digit indicates a vertex. The order of the four vertex points is arbitrary. Though three vertices of the four can define a

face, it is checked whether the remaining vertex is on the face or not. If the user wants to define a face with three vertices, the remaining vertex must be either 0 or a repeat of one of the other vertices. If the number of faces is less than six, the remaining face description must be zero. Any set of the coordinates can be specified for the unused vertex.

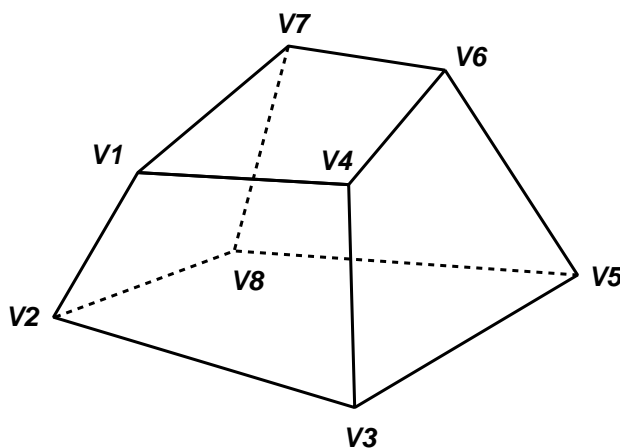


Fig. 10.7 BODY ARB

The following input is an example for BODY ARB. The coordinates of vertices **V1**, **V2**, ... , **V8** are (2, 2, -1), (-2, 2, -1), ... , (1, -1, 1), respectively. The planes which consist of the vertices are specified by 4-digit integers 1234, 5678, ... , 4158. The integer of 1234 denotes the plane which consists of vertices **V1**, **V2**, **V3** and **V4** and other integers denote in the same way.

```
ARB(  1   2   2  -1   -2   2  -1   -2  -2  -1   2  -2  -1
      1   1   1   -1   1   1   -1  -1   1   1  -1   1
      1234 5678 1265 2376 3487 4158 )
```

10.2.8 BODY HAF (Half Space with a Planer Surface)

Half space divided by a planer surface. See Fig. 10.8 for the definition of a HAF.

■ Format ■

```
HAF( ID d1 d2 d3 d4 )
```

■ Parameter ■

ID : BODY ID

d1, *d2*, *d3* : outward vector $\mathbf{N}(N_x, N_y, N_z)$ normal to the boundary surface (arbitrary length)

d4 : distance *A* from the surface to the origin in direction \mathbf{N} (*d4* can be a negative value.)

■ Description ■

The internal points satisfy the following inequality:

$$\frac{d1 \cdot x + d2 \cdot y + d3 \cdot z}{\sqrt{d1^2 + d2^2 + d3^2}} + d4 < 0.$$

If $d4 < 0$, the origin is in the inside of the half space. Otherwise, it is in the outside.

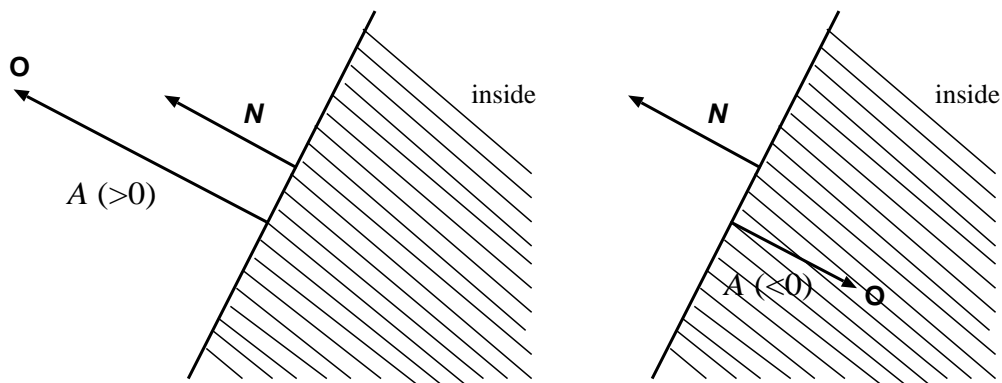


Fig. 10.8 BODY HAF

10.2.9 BODY RHP (Right Hexagonal Prism)

Right hexagonal prism with the central axis parallel to the z -axis and sides perpendicular to the x -axis. See Fig. 10.9 for the definition of an RHP.

■ Format ■

RHP (ID d1 d2 d3 d4 d5)

■ Parameter ■

ID : BODY ID

d1, d2, d3 : coordinates $V(V_x, V_y, V_z)$ at the center of one base

d4 : height h

d5 : width W of the base

■ Description ■

If $d4 \geq \text{DINF}$ (distance treated as infinite in the codes) or $d4 = 0.0$, then the right hexagonal prism has the infinite height. BODY HEX must be used for a right hexagonal prism whose side is not perpendicular to the x -axis even if the central axis is parallel to the z -axis. If this BODY is used to define FRAME ZONES for a hexagonal LATTICE, X_ℓ -, Y_ℓ -, and Z_ℓ -axes in the LATTICE-coordinate system are set to be parallel to the x -, y - and z -axes, respectively.

10.2.10 BODY HEX (Hexagonal Prism)

General hexagonal prism. See Fig. 10.10 for the definition of a HEX.

■ Format ■

HEX (ID d1 d2 d3 d4 d5 d6 d7 d8 d9 d10)

■ Parameter ■

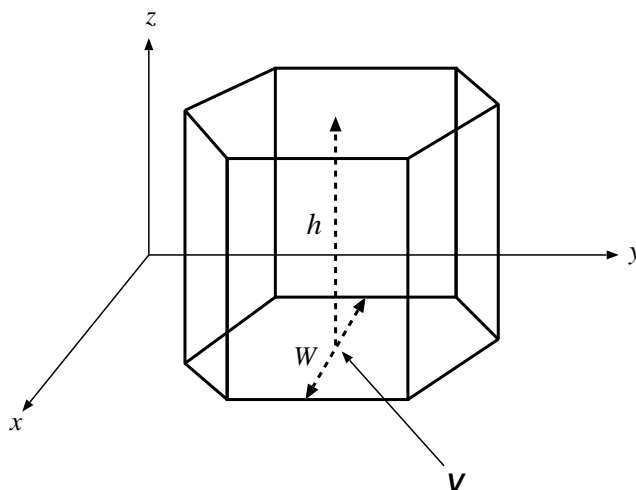


Fig. 10.9 BODY RHP

ID : BODY ID

d1, d2, d3 : coordinates $\mathbf{V}(V_x, V_y, V_z)$ at the center of one base

d4, d5, d6 : height vector $\mathbf{H}(H_x, H_y, H_z)$

d7 : width W of the hexagonal base

d8, d9, d10 : normal vector $\mathbf{A}(A_x, A_y, A_z)$ to one of the side faces (arbitrary length)

■ Description ■

The coordinates $(d1 + d4, d2 + d5, d3 + d6)$ correspond to the center of the upper base. If the length of vector \mathbf{H} ($\sqrt{d4^2 + d5^2 + d6^2}$) is greater than DINF (distance treated as infinite in the codes), the hexagonal prism has the infinite height and no bases. Two vectors \mathbf{A} and \mathbf{H} must be perpendicular with each other. If this BODY is used to define FRAME ZONES for a hexagonal LATTICE, the X_ℓ -, Y_ℓ - and Z_ℓ -axes in the LATTICE-coordinate system are set to be parallel to \mathbf{A} , $\mathbf{H} \times \mathbf{A}$, \mathbf{H} , respectively.

10.2.11 BODY RCL (Right Cylinder with Specified Radial Direction)

Right cylinder with specified radial direction. See Fig. 10.11 for the definition of an RCL.

■ Format ■

RCL(*ID d1 d2 d3 d4 d5 d6 d7 d8 d9 d10*)

■ Parameter ■

ID : BODY ID

d1, d2, d3 : coordinates $\mathbf{V}(V_x, V_y, V_z)$ at the center of one base

d4, d5, d6 : height vector $\mathbf{H}(H_x, H_y, H_z)$

d7 : radius R

d8, d9, d10 : vector $\mathbf{A}(A_x, A_y, A_z)$ to specify the base direction (arbitrary length)

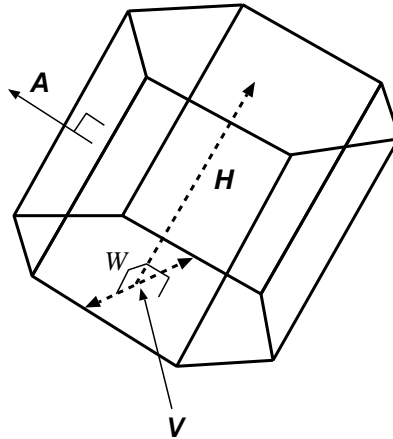


Fig. 10.10 BODY HEX

■ Description ■

This BODY has the same shape as RCC (right circular cylinder) and is used to define a FRAME ZONE for a hexagonal LATTICE. \mathbf{A} is an arbitrary-length vector denoting the X_ℓ -direction in the LATTICE coordinates of a hexagonal LATTICE. The X_ℓ -, Y_ℓ - and Z_ℓ -axes in the LATTICE coordinates are set to be parallel to the \mathbf{A} , $\mathbf{H} \times \mathbf{A}$, \mathbf{H} direction, respectively.

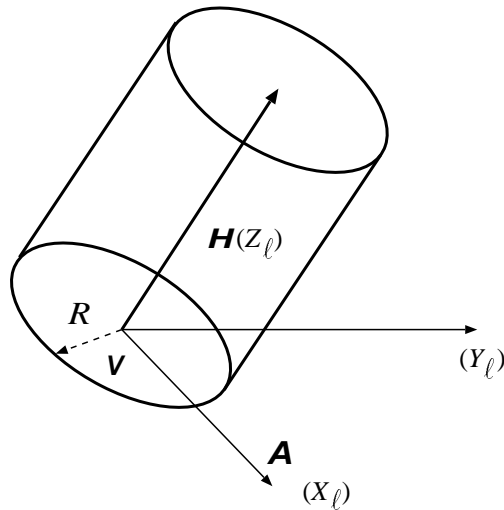


Fig. 10.11 BODY RCL

10.2.12 BODY TRC (Truncated Right Cone)

Truncated right cone. See Fig. 10.12 for the definition of a TRC.

■ Format ■

TRC(ID d1 d2 d3 d4 d5 d6 d7 d8)

■ Parameter ■

ID : BODY ID

d1, d2, d3 : coordinates $\mathbf{V}(V_x, V_y, V_z)$ at the center of one base

d4, d5, d6 : height vector $\mathbf{H}(H_x, H_y, H_z)$

d7 : radius $R1$ of the lower base

d8 : radius $R2$ of the upper base

■ Description ■

The lower and upper bases are perpendicular to the central axis.

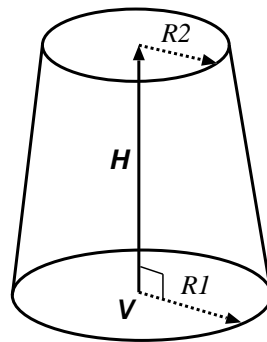


Fig. 10.12 BODY TRC

10.2.13 BODY TEC (Truncated Elliptic Cone)

Truncated elliptic cone See Fig. 10.13 for the definition of a TEC.

■ Format ■

TEC(ID d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12 d13)

■ Parameter ■

ID : BODY ID

d1, d2, d3 : coordinates $\mathbf{V}(V_x, V_y, V_z)$ at the center of one base

d4, d5, d6 : height vector $\mathbf{H}(H_x, H_y, H_z)$

d7, d8, d9 : radius vector $\mathbf{A}(A_x, A_y, A_z)$ of the major (or minor) axis

d10, d11, d12 : radius vector $\mathbf{B}(B_x, B_y, B_z)$ of the minor (or major) axis

d13 : ratio r of one base to the other

■ Description ■

Vectors \mathbf{A} and \mathbf{B} must be perpendicular to each other. A cross section parallel to the base is a similar ellipse and the semi-axes are parallel to those of the base.

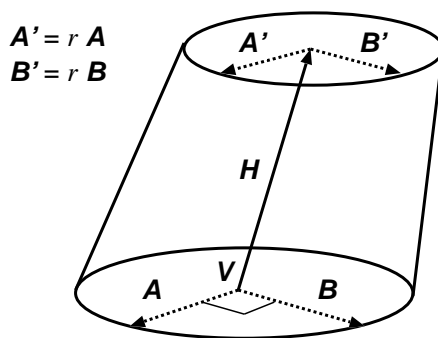


Fig. 10.13 BODY TEC

10.2.14 BODY ELL (Spheroid)

Spheroid (ellipsoid of revolution). See Fig. 10.14 for the definition of an ELL.

■ Format ■

ELL(ID d1 d2 d3 d4 d5 d6 d7)

■ Parameter ■

ID : BODY ID

d1, d2, d3 : coordinates $V1(V1_x, V1_y, V1_z)$ of a focus

d4, d5, d6 : coordinates $V2(V2_x, V2_y, V2_z)$ of the other focus

d7 : length D of the major axis (not major radius)

■ Description ■

The axis of revolution is a line that passes through the two foci. The length of the minor axis is calculated as follows:

$$r = \sqrt{d7^2 - \{(d1 - d4)^2 + (d2 - d5)^2 + (d3 - d6)^2\}}.$$

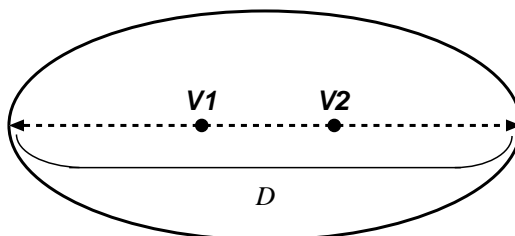


Fig. 10.14 BODY ELL

10.2.15 BODY GEL (General Ellipsoid)

General ellipsoid. See Fig. 10.15 for the definition of a GEL.

■ Format ■

GEL(ID d1 d2 d3 d4 d5 d6 d7 d8 d9 d10 d11 d12)

■ Parameter ■

ID : BODY ID

d1, d2, d3 : coordinates $\mathbf{V}(V_x, V_y, V_z)$ of the center

d4, d5, d6 : axis vector $\mathbf{A1}(A1_x, A1_y, A1_z)$ (arbitrary length)

d7, d8, d9 : axis vector $\mathbf{A2}(A2_x, A2_y, A2_z)$ (arbitrary length)

d10, d11, d12 : lengths $R1, R2, R3$ of the three half axes

■ Description ■

Axis vectors $\mathbf{A1}$ and $\mathbf{A2}$ must be perpendicular to each other. The length is arbitrary. The third axis vector is calculated internally in the codes.

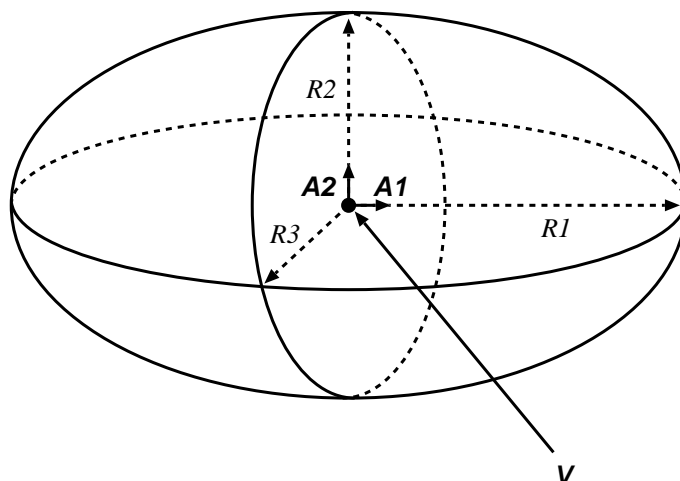


Fig. 10.15 BODY GEL

10.2.16 BODY ELT (Elliptical Torus)

Elliptical torus. See Fig. 10.16 for the definition of an ELT.

■ Format ■

ELT(ID d1 d2 d3 d4 d5 d6 d7 d8 d9 d10)

■ Parameter ■

ID : BODY ID

d1, d2, d3 : coordinates $\mathbf{V}(V_x, V_y, V_z)$ of the center of the hole

d4, d5, d6 : vector $\mathbf{N}(N_x, N_y, N_z)$ defining the rotation axis (arbitrary length)

d7 : distance R_m to the center of the ellipse

d8 : major (or minor) radius $R1$ of the ellipse

d9 : minor (or major) radius $R2$ of the ellipse

d10 : angle τ of the $R2$ direction from the plane perpendicular to the rotation axis (degree)

■ Description ■

τ is the angle of the $R2$ direction from the plane perpendicular to the rotation axis.

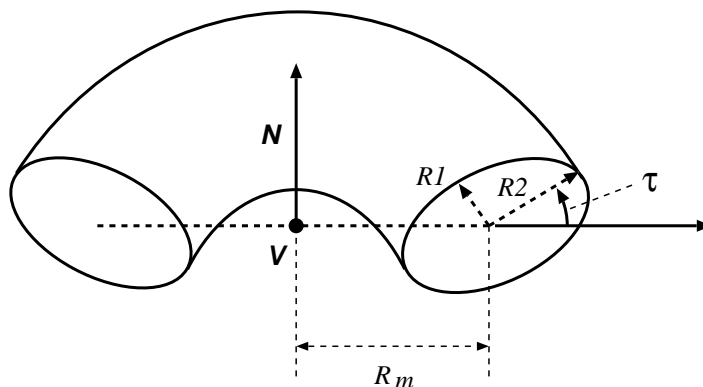


Fig. 10.16 BODY ELT

10.2.17 BODY GQS (General Quadratic Curved Surface)

General quadratic curved surface. This general input form is used when a quadratic curved surface cannot be defined by the normal BODIEs such as SPH, RCC, TRC, TEC, GEL, etc.

■ Format ■

GQS(*ID d1 d2 d3 d4 d5 d6 d7 d8 d9 d10*)

■ Parameter ■

ID : BODY ID

d1, d2, d3 : coefficients S_{xx}, S_{yy}, S_{zz} of the second order terms

d4, d5, d6 : coefficients S_{xy}, S_{yz}, S_{zx} of the second order cross terms

d7, d8, d9 : coefficients S_x, S_y, S_z of the first order terms

d10 : constant C

■ Description ■

The internal part of the BODY is defined by the following inequality:

$$S_{xx}x^2 + S_{yy}y^2 + S_{zz}z^2 + S_{xy}xy + S_{yz}yz + S_{zx}zx + S_x x + S_y y + S_z z + C \leq 0.$$

10.3.1 ZONE Name

ZONE-name is a ZONE name given for each ZONE. The length of the *ZONE-name* must be not more than 12 characters. The first character of *ZONE-name* must be located in the 1st to 5th column. *ZONE-name* is delimited by the first “:” and should not include a blank. If the blank is inserted, the characters after the blank are neglected. The same ZONE name can be used for different ZONES. The ZONE names are used for rejection sampling by the ZONE in source sampling.

10.3.2 REGION Name

REGION-name is a REGION name given to a ZONE. The REGION name must be alphabetical letters, numbers or underscores (.) and it is case sensitive. The length of the REGION name must be not more than 12 characters.

The REGION name is used to define the subspaces for tallies and the variance reduction parameters etc. by the REGION. It can be specified independently of the ZONE material. The same name can be used for more than one ZONE but each ZONE must belong to only one REGION. If *REGION-name* is not specified for a ZONE, the ZONE belongs to the REGION without a REGION name. It is meaningless to define the REGION name for some ZONES such as a ZONE filled with a fictitious material (*material-ID* ≤ -999).

10.3.3 Material ID

material-ID is an integer to define a material filling a ZONE. The material ID's have the following meanings. Material ID's for fictitious materials are also used to define the boundary conditions shown in Table 10.2.

Table 10.2 Material ID's and materials

Material ID	Material
1 or more	Real material
0	Void (“inner void”)
−1 to −998	Include LATTICE geometry
−999	Fictitious material to define outside of a CELL
−1000	Perfect absorber (black absorber, “outer void”)
−2000	Mirror reflector
−3000	White reflector
−4000	Fictitious material to define periodic boundaries

In general, the negative material ID numbers are assigned to special materials and these special materials enable us to describe the boundary conditions or hierarchical geometry (‘LATTICE’ in the following section).

The whole space for random walk of particles must be surrounded with some perfect absorbers, (vacuum), some reflective materials and fictitious materials used for the periodic boundary condition. The fictitious materials for boundary conditions must be used even if a calculation is performed for an infinite geometry such as a 1-dimensional infinite cylinder or a 1-dimensional infinite slab. The user can define a BODY which has an infinite length in a direction but they are not allowed to represent the above infinite geometry only with the BODY.

The following specification in place of Material ID is also available to define the ZONE which includes a LATTICE:

LAT=*LATTICE-ID*

or

L=*LATTICE-ID*

where *LATTICE-ID* is a positive integer to identify the LATTICE. The material ID's from -1 to -998 are used to specify the LATTICES as shown in Table 10.2, where the absolute values are the LATTICE ID's. Thus, the LATTICE ID's must be from 1 to 998. If the LATTICE ID is specified as "LAT=" or "L=", the arbitrary positive integer can be used as the LATTICE ID.

10.3.4 Combinatorial Geometry

Combinatorial geometry (CG) is employed in MVP/GMVP. A calculation geometry is defined by combination of BODIEs as follows.

- (1) +{BODY ID number} or -{BODY ID number}

The positive and negative signs represent the inside and outside of a BODY, respectively. The positive sign "+" can be omitted.

- (2) "OR"

This is a logical operator to represent the union of BODIEs. Just a series of BODY ID numbers represents the intersection of BODIEs.

If more than one line are required to specify the combination of BODIEs, the 1st to 5th columns must be blanks in the following lines for data continuation.

10.3.5 'INPUT-ZONE' and 'ZONE'

The ZONEs defined in the geometry input are treated as 'INPUT-ZONE' in the codes. If the "OR" operator is used, the 'INPUT-ZONE' is separated into more than one ZONE that may have overlap. Practically, the ZONE definition with "OR" is equivalent to the multiple ZONE definition without "OR" as described below.

```
ZONE1 : REGION1 : <MAT1> : 1 -2 OR 10 11 -20 OR 12000 -90000
```

and the other ZONE definition:

```
ZONE1 : REGION1 : <MAT1> : 1 -2
ZONE1 : REGION1 : <MAT1> : 10 11 -20
ZONE1 : REGION1 : <MAT1> : 12000 -90000
```

are the same. In both cases, the multiple ZONEs may be doubly defined in space. However, such definition is allowed because those ZONEs have the same REGION name and contain the same material.

10.3.6 ZONE Specification and Calculation Time

Particles usually travel across some ZONEs in the random walk process. Then the following processes are performed in the codes.

- (a) Calculate the distance from the current position to a ZONE boundary along the flight path.

- (b) Search for a next ZONE where a particle enters when the particle reaches the ZONE boundary.

The distances to the surfaces of all BODIEs defining the ZONE must be calculated in the first process and all possible contiguous ZONES must be searched in the second process. This is why the following geometry data takes a lot of calculation time.

- (1) Too many components (BODIEs) defining a ZONE.
- (2) Too many contiguous ZONES.

MVP/GMVP memorizes information of the ZONES contiguous to each ZONE through the process of random walk and then the memorized ZONES are first searched in the next-ZONE search task. Of course, all ZONES are investigated at the beginning of the calculation. The available memory size for the ZONE search is limited and it is input by NMEMO. If no ZONES can be found in the memorized data, the codes investigate all possible ZONES except for the memorized ZONES. If this situation occurs frequently, the user should specify the larger value of NMEMO. The default value is set to 5 but the larger value is recommended in many cases.

However, the larger value of NMEMO cannot always solve problems (1) and (2) mentioned above. The following technique is recommended to solve them thoroughly.

- (a) Splitting of a space with the same material and REGION name to multiple parts as shown in Fig. 10.17. From another point of view, it has the advantage that it is easy to understand the geometry input data.

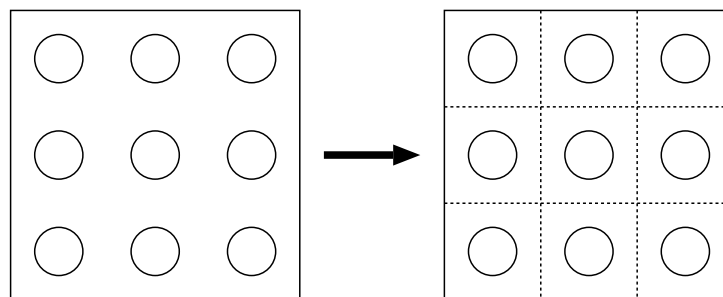


Fig. 10.17 Splitting calculation geometry

- (b) Use of 'LATTICE geometry' in the case of repetition of the same CELLS.

10.3.7 Periodic Boundary Condition

The periodic boundary condition is available for geometries with parallel planes in MVP/GMVP. Currently, it cannot be used for those with rotational symmetry.

The periodic boundary condition must be defined by using the fictitious material of ID "−4000". Figure 10.18 shows the schematic view for the periodic boundary condition. The pair of two planes that define the periodic boundaries must be parallel and included in the same BODY. If the planes are parallel but each plane belongs to the different BODY, they cannot be the periodic boundaries.

The following planes can be used as the periodic boundaries.

- Rectangle : a pair of parallel planes of RPP or BOX.
- Right hexagonal prism : a pair of parallel planes of RHP or HEX
- Cylinder : a pair of parallel planes of CYL or RCC.

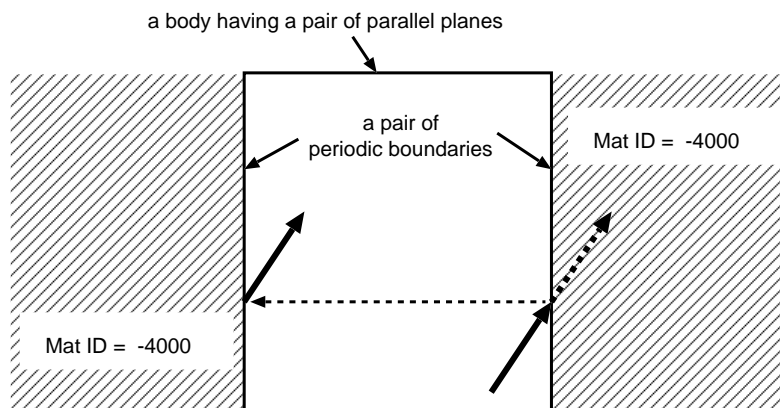


Fig. 10.18 Specification of periodic boundary condition and tracking of a particle

10.4 Input Data for LATTICE Geometry

This section describes input data for LATTICE geometry. The concept of the geometry and the definition of some terms specific to MVP/GMVP are described in Section 5.5. The user unfamiliar with them should see Section 5.5 first.

10.4.1 Specification of LATTICE Structure

In order to use the LATTICE geometry, the LATTICE option must be specified. With this option, the LATTICE parameters (size, the number of CELLS, arrangement of CELLS, etc.) must be defined at the beginning of the GEOMETRY data block for a rectangular or hexagonal lattice. The LATTICE parameters for the statistical geometry model are also defined at the beginning but their meaning is different. The details are described in Section 10.8.

```
$GEOMETRY
  IDLAT( LATTICE-ID-1 )
  LTPP( LATTICE-type )
  NVLAT( nx ny nz )
  SZLAT( s1 s2 s3 )
  SZHEX( h1 h2 h3 h4 h5 ) /* (only for hexagonal LATTICE)
  RCELL( ix iy ) /* (only for hexagonal LATTICE)
  KLATT( CELL CELL CELL ... )
  KSLAT( dir1 dir2 dir3 ... )

  IDLAT( LATTICE-ID-2 )
  ....
  ....
  END
... BODY Data
.....
```

In the above specification, the following style can be used for KLATT and KSLAT.

```

CELLIDS( CELL CELL CELL ... )
CELLSYMS( CELL-symbol CELL-symbol ... )
CKLATT( CELL-pattern-string ... )
CELLDIRS( dir1 dir2 ... )
DIRSYMS( direction-symbol ... )
CKSLAT( direction-pattern-string ... )

```

The following data must be specified for each LATTICE. The line including only "END" indicates the end of the input for the LATTICE data.

(1) IDLAT

LATTICE ID number (usually, $1 \leq \text{IDLAT} \leq 998$). This must be input first for the specification of each LATTICE. The LATTICE is placed in the FRAME ZONEs which are the ZONEs with the negative-signed LATTICE ID number. A positive integer more than 998 can be used as a LATTICE ID if the form of $\text{LAT}=\text{LATTICE-ID}$ ($\text{L}=\text{LATTICE-ID}$) is used for the specification of the material ID.

(2) LTYP

LATTICE type.

- LTYP=1 : Rectangular LATTICE. The LATTICE consists of rectangular-parallelepiped SUBFRAMEs of the same size.
- LTYP=2 : Hexagonal LATTICE with a hexagonal-prism frame. The LATTICE consists of hexagonal prism SUBFRAMEs of the same size.
- LTYP=3 : Hexagonal LATTICE with a cylinder frame. The LATTICE consists of hexagonal prism SUBFRAMEs of the same size.
- LTYP=4 : Hexagonal LATTICE with a rectangular-prism frame. The LATTICE consists of hexagonal prism SUBFRAMEs of the same size.
- LTYP=10 : Statistical geometry model (treated as a kind of LATTICEs in the code). The inside of the LATTICE is filled probabilistically with spherical BODIEs at the specified packing fraction.

(3) NVLAT(*nx ny nz*)

The number of SUBFRAMEs along each axis in the LATTICE coordinates.

(4) SZLAT(*s1 s2 s3*)

Parameters to define the size of a SUBFRAME and the relation between the CELL-array coordinates and the LATTICE coordinates of a hexagonal LATTICE. The meaning of the 3 parameters is shown in Table 10.3.

The angle θ between the X_a -axis in the CELL-array coordinates (Fig. 5.9) and the X_ℓ -axis in the LATTICE coordinates (Fig. 5.10) plus 30 degrees must be input for *s2* in the hexagonal LATTICE.

(5) SZHEX(*h1 h2 h3 h4 h5*)

This parameter list is used only for the hexagonal LATTICE. The parameters indicate the size of the outer frame of a hexagonal LATTICE (LATTICE FRAME) and the relation between the

CELL-array coordinates and the LATTICE coordinates. The meaning of each parameter is shown in Table 10.4.

The parameters $h1$, $h2$ define the CELL arrangement in the FRAME. They are the X_ℓ - and Y_ℓ -components of the vector from the center of the LATTICE FRAME to that of the reference SUBFRAME in the X_ℓ - Y_ℓ plane of the LATTICE coordinates.

The reference SUBFRAME is defined by RCELL(i j) described below. The center position is at the center of the hexagonal SUBFRAME.

The parameters $h3$, $h4$ and $h5$ depend on $LTYP = 2, 3, 4$.

(6) RCELL(ix iy)

Position index of the reference SUBFRAME in the CELL-array coordinates for a hexagonal LATTICE. As shown in Fig 10.19, the parameters must be specified as “RCELL(3 4)” since the indices in the CELL-array coordinates are $I = 3$, $J = 4$. See also Fig. 5.9.

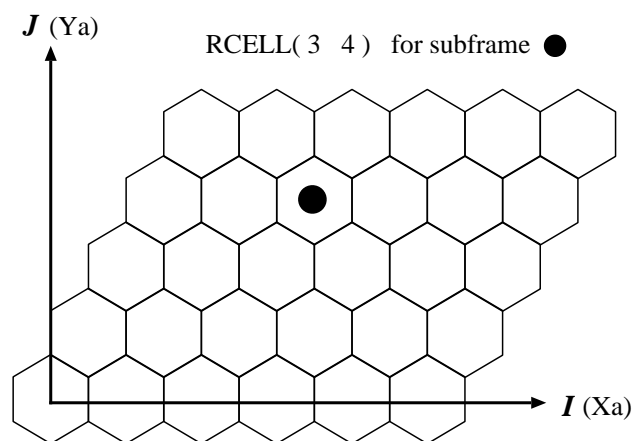


Fig. 10.19 RCELL to specify the reference SUBFRAME in a hexagonal LATTICE

(7) KLATT(i, j, k) ($i=1, NVLAT(1)$, $j=1, NVLAT(2)$, $k=1, NVLAT(3)$)

CELL ID number located at SUBFRAME(i, j, k) in a LATTICE. “ i, j, k ” are the indices of X_a -, Y_a - and Z_a -axes in the CELL-array coordinates in Fig 5.9, respectively. The KLATT data are stored in memories in the same way as a 3-dimensional array in FORTRAN. But the input order for index i is from 1 to $NVLAT(1)$ and that for indices j and k is from $NVLAT(2)$ to 1 and from $NVLAT(3)$ to 1, respectively. This input order means that the arrangement for (i, j) is just the same as it is seen from the positive viewpoint along the z -axis. For Fig 5.9, CELL ID numbers must be input in the order of (1,3), (2,3), (3,3), (4,3), (1,2), (2,2), (3,2), (4,2), (1,1), (2,1), (3,1), (4,1).

In the case of $KLATT(i, j, k) = 0$, the SUBFRAME is not included in the LATTICE and the corresponding subspace must be defined as a different ZONE or a LATTICE region. In this case, the subspace must be excluded in the definition of the FRAME ZONE where the LATTICE is placed or on the outside of the FRAME of the hexagonal LATTICE.

(8) KSLAT(i, j, k) ($i=1, NVLAT(1)$, $j=1, NVLAT(2)$, $k=1, NVLAT(3)$)

KSLAT defines the direction of CELLS placed in SUBFRAMEs. For $KSLAT = 0$, the CELL is placed along the direction defined in the CELL data. Namely, it is placed so that the Cartesian

CELLIDS, CELLSYMS and CKLATT must be input in this order as follows.

- CELLIDS

List of the CELL ID's to be located in a LATTICE. The CELL ID can be multiply defined.

- CELLSYMS

List of characters which are delimited by one or more blanks. The characters cannot be multiply defined. The number of parameters in CELLSYMS must be the same as that in CELLIDS. Each character in CELLSYMS represents uniquely the CELL ID of the corresponding position in CELLIDS. The available characters are capital/small alphabets, numbers and symbols except for parentheses, “/” and “*”.

- CKLATT

Specify the CELL arrangement by one-character symbol defined in CELLSYMS. The rule for the input order of the CELLS is the same as KLATT. The characters except for blanks, return (enter) codes, comment lines and comment character strings are effective as the input data.

(10) CELLDIRS, DIRSYMS and CKSLAT

These are another input format for KSLAT which defines the CELL directions. They are used as follows.

```
CELLDIRS( 0 0 100 010 110 )
DIRSYMS( 0 A X Y M )
CKSLAT( XXXXX0AAAAA
        XXXXX0AAAAA
        XXXXX0AAAAA
        00000000000
        MMMMM0YYYYY
        MMMMM0YYYYY
        MMMMM0YYYYY )
```

CELLDIRS, DIRSYMS and CKSLAT must be input in this order as follows.

- CELLDIRS

List of values which define the directions of CELLS to be placed in a LATTICE. The values can be multiply defined. The meaning of specified values is explained in the KSLAT section.

- DIRSYMS

List of characters which are delimited by one or more blanks. The characters cannot be multiply defined. The number of parameters in DIRSYMS must be the same as that in CELLDIRS. Each character in DIRSYMS represents uniquely the CELL direction of the corresponding position in CELLDIRS. The available characters are capital/small alphabets, numbers and symbols except for parentheses, “/” and “*”.

- CKSLAT

Specify the CELL direction by one-character symbol defined in DIRSYMS. The rule for the input order of the CELLS is the same as KSLAT. The characters except for blanks, return (enter) code, comment lines and comment character strings are effective.

Table 10.3 Parameters in SZLAT to define CELLS

SZLAT	Rectangular LATTICE LTYP=1	Hexagonal LATTICE LTYP=2,3,4
1	width of a CELL in the X_a -axis	pitch of CELLS (distance between parallel planes)
2	width of a CELL in the Y_a -axis	$\theta+30$ (degree). θ is the angle of the X_a -axis in the CELL-array coordinate system from the X_ℓ -axis in the LATTICE coordinate system.
3	width of a CELL in the Z_a -axis	height of a CELL

Table 10.4 Parameters in SZHEX to define the frame size of a hexagonal LATTICE and the relation between the CELL-array coordinates and the LATTICE coordinates

SZHEX	LTYP=2 hexagonal frame	LTYP=3 cylindrical frame	LTYP=4 rectangular frame
1	X_ℓ -component of a vector from the center of a LATTICE FRAME to the center of a reference SUBFRAME		
2	Y_ℓ -component of a vector from the center of a LATTICE FRAME to the center of a reference SUBFRAME		
3	width of a hexagon	radius	width along the X_ℓ -direction
4	height	height	width along the Y_ℓ -direction
5	(dummy data)	(dummy data)	width along the Z_ℓ -direction

10.4.2 Examples of Input Data for LATTICE Geometry

The LATTICE ID's of 10 and 100 are defined for 2 rectangular LATTICES (LTYP = 1) in Example 1.

Example 1:

```

$GEOMETRY
% PA = 4.0, HT = 366.0, P = 1.26
***** DATA FOR LATTICE ID = 10 *****
IDLAT( 10 )
  LTYP( 1 ) NVLAT( 2 2 1 ) SZLAT( <PA> <PA> <HT> )
  KLATT( 1 2
        2 1 )
  KSLAT( 0 0
        110 110 )
***** DATA FOR LATTICE ID = 100 *****
IDLAT( 100 )
  LTYP( 1 ) NVLAT( 3 3 1 ) SZLAT( <P> <P> <HT> )
  KLATT( 10 10 10
        10 20 10
        10 10 10 )
  KSLAT( < 3*3 >(0) )
END

```

- LATTICE ID 10:

“LTYPE(1)” stands for the rectangular LATTICE. “NVLAT(2 2 1)” stands for the CELL arrangement of two CELLS in the X_a - and Y_a -directions, one CELL in the Z_a -direction in the CELL-array coordinates. See Fig. 10.20 for the CELL arrangement of LATTICE ID 10. The 3 parameters in SZLAT give the size of the CELLS along the X_a -, Y_a -, and Z_a -directions. Note that these parameters do not represent the size of the LATTICE.

KLATT represents the CELL ID's to be placed in each SUBFRAME in the LATTICE. Here CELL 1 is placed at the upper left and lower right SUBFRAMEs and CELL 2 is placed at the upper right and lower left SUBFRAMEs.

KSLAT represents the direction of CELLS placed in each SUBFRAME. KSLAT is set to 0 for CELL 1 at upper left and CELL 2 at upper right, and it means the CELLS are placed such that the x -, y - and z -directions in the CELL coordinates $((X_c, Y_c, Z_c))$ are the same as those in the CELL-array coordinates $((X_a, Y_a, Z_a))$. On the other hand, KSLAT is set to 110 for CELL 2 at lower left and CELL 1 at lower right. Thus the CELLS are reversed in the x - and y -directions since both the digits of 100 and 10 are 1.

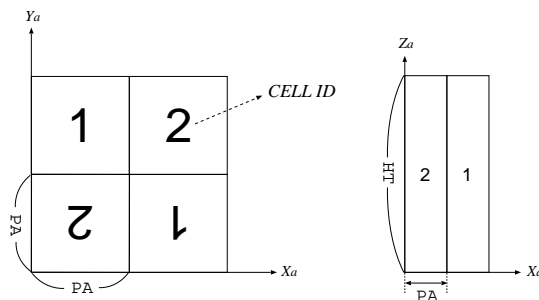


Fig. 10.20 Arrangement of CELLS for LATTICE ID 10

- LATTICE ID 100

This LATTICE is a rectangular LATTICE of “LTYPE(1)” as well as LATTICE 10 but the CELL arrangement consists of three CELLS in the X_a - and Y_a -directions, one CELL in the Z_a -direction. CELLS 10 and 20 are used for the CELL arrangement defined in KLATT. In this case, CELL ID 10 is the same number as the LATTICE ID 10 but the former indicates a CELL and the latter does a LATTICE. It should be noted that they should be distinguished. The parameters in KSLAT are all 0 and thus the CELLS are not reversed.

The following is an example for the hexagonal LATTICE. This is a typical example for fuel assemblies of a FBR.

Example 2:

```
$GEOMETRY
% P = 0.787, HT = 93.0 /** CELL pitch and height.
% A = P/SQRT(3) /** Length of a side of CELL hexagon.
% NL=4, DH = 11.0*A /** Number of pin layers and frame width.
IDLAT(100)
  LTYPE(2) NVLAT( 9 9 1 )
  SZLAT( <P> 0.0 <HT> )
```

```

RCELL( 5 5 )
SZHEX( 0.0 0.0
      <DH> <HT> 0.0 )
KLATT(
      2 2 2 2 2 0 0 0 0
      2 1 1 1 1 2 0 0 0
      2 1 1 1 1 1 2 0 0
      2 1 1 1 1 1 1 2 0
      2 1 1 1 1 1 1 1 2
      0 2 1 1 1 1 1 1 2
      0 0 2 1 1 1 1 1 2
      0 0 0 2 1 1 1 1 2
      0 0 0 0 2 2 2 2 2
)
KSLAT( <9*9>(0) )

```

This example defines a hexagonal LATTICE consisting of a 9×9 hexagonal CELL array and a hexagonal LATTICE FRAME as shown in Fig. 10.21.

- LTYP(2) defines the hexagonal LATTICE whose frame is a right hexagonal prism.
- NVLAT(9 9 1) stands for the CELL arrangement of 9 CELLS in the X_a - and Y_a -directions, one CELL in the Z_a -direction in the CELL-array coordinates.
- The first value of SZLAT is the pitch of the CELLS. It is the distance between the centers of the adjacent CELLS.
- The second value of SZLAT is the counter-clockwise rotation angle from the X_ℓ -axis in the LATTICE coordinates (X_ℓ, Y_ℓ, Z_ℓ in Fig. 10.21) to the X_a -axis in the CELL-array coordinates (X_a, Y_a, Z_a in Fig. 10.21). The direction of the Z_a -axis must coincide with that of the Z_ℓ -axis. In this figure, the angle is -30° but 0 must be input for this value because the input value is automatically reduced by 30 in the codes.
- The third value of SZLAT is the length of the CELLS in the Z_a -direction. Since NVLAT(3) is 1 in this data, SZLAT(3) is the length of the whole LATTICE in the Z_ℓ -direction.
- For a hexagonal LATTICE, the position relation between the CELL-array and the LATTICE FRAME must be defined. The relative direction of the CELL-array coordinates to the LATTICE FRAME is defined by SZLAT(2). In addition, the position of the LATTICE FRAME must be defined in the CELL array. It is defined by the relative position of the center of the reference SUBFRAME specified by RCELL to the center of the LATTICE FRAME. The indices in X_a - and Y_a -directions must be input for RCELL. In this example, it is defined as RCELL(5 5), namely the SUBFRAME at the center of the CELL arrangement.
- The first two values of SZHEX indicate the relative position of the center of the reference SUBFRAME specified by RCELL to the center of the LATTICE FRAME. They are the components of the 2-dimensional vector in the LATTICE coordinates (X_ℓ, Y_ℓ). In this example, the center of the reference SUBFRAME is the same location as that of the LATTICE FRAME and thus both SZHEX(1) and SZHEX(2) are 0. However, they are not necessary to be the same. The following examples show such cases. All these inputs are equivalent to Example 2.

– RCELL(4 4) : SZHEX(1)= <-3*A> , SZHEX(2)= 0

- RCELL(1 1) : SZHEX(1)= <-12*A> , SZHEX(2)= 0
 - RCELL(5 6) : SZHEX(1)= <1.5*A> , SZHEX(2)= <P/2>
 - RCELL(6 7) : SZHEX(1)= <4.5*A> , SZHEX(2)= <P/2>
 - RCELL(9 1) : SZHEX(1)= 0 , SZHEX(2)= <-4*P>
- The other values of SZHEX indicate the size of hexagonal LATTICE FRAME and are defined for each LATTICE FRAME as shown in Table 10.4. Since LTYP is 2 in this example, the last three values of SZHEX are the width, height of the hexagonal LATTICE FRAME and dummy data, respectively. The value which indicates the height of the LATTICE FRAME is the same as that of CELLS in this example.
 - KLATT shows that the LATTICE consists of CELLS 1 and 2. Some SUBFRAMEs specified as 0 for CELL ID's are located outside of the LATTICE FRAME as found in Fig. 10.21. Like this case, CELLS outside of the LATTICE FRAME should be specified as 0 or any CELL of which size is the same as that of the SUBFRAME. The LATTICE FRAME is located across the SUBSPACEs with CELL 2 and only the CELL geometry on the inside of the LATTICE FRAME is meaningful in defining the LATTICE.
 - Since all values of KSLAT are 0, neither rotation nor reverse is performed in placing the CELLS in the SUBFRAMEs.

10.4.3 Arrangement of LATTICEs

There are two ways to define the FRAME ZONEs where LATTICEs are arranged.

- The material ID number is specified as $-1 \times \text{IDLAT}$. IDLAT is the LATTICE ID number described in Section 10.4.1. In this case, the LATTICE ID's are limited between 1 and 998.
- “LAT=LATTICE ID” or “L=LATTICE ID” is specified in the location where the material ID number must be input in the ZONE specification. In this case, the LATTICE ID must be an arbitrary positive integer.

The same LATTICE can be placed in more than one FRAME ZONE. The restriction on the shape of the FRAME ZONE depends on the FREE-LATTICE-FRAME option. The user must note that this option affects all LATTICEs in the input data.

■ Case where the FREE-LATTICE-FRAME option is not used.

- The “OR” operator cannot be used in the combination of BODYs to define the FRAME ZONE.
- The first BODY in the definition must be the same size and shape as the LATTICE FRAME. Therefore, for each LTYPE, usable BODYs are limited as follows.
 - LTYPE=1 : RPP or BOX (BOX must be a rectangular parallelepiped.)
 - LTYPE=2 : RHP or HEX
 - LTYPE=3 : RCL (CYL, RCC are not available.)
 - LTYPE=4 : RPP or BOX (BOX must be a rectangular parallelepiped.)

In the case where a LATTICE is arranged in a FRAME ZONE, the direction is determined as follows.

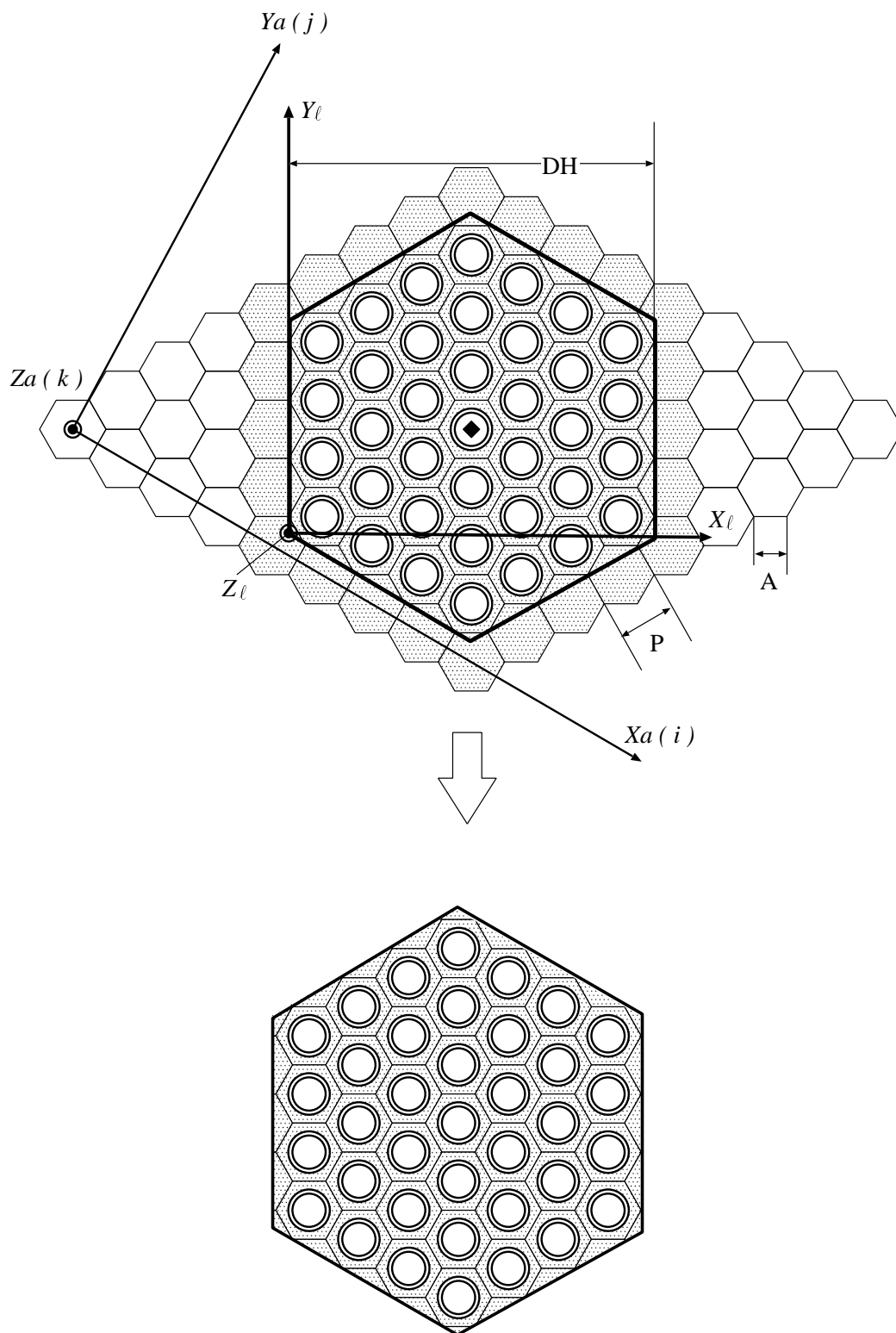


Fig. 10.21 Example of a hexagonal LATTICE with a hexagonal frame

- Case of the RPP frame : The X_ℓ , Y_ℓ , Z_ℓ -directions of a LATTICE correspond to the x,y,z-direction of RPP, respectively.
- Case of the BOX frame : The X_ℓ , Y_ℓ , Z_ℓ -directions of a LATTICE correspond to the first, second and third side-vectors of BOX, respectively. The side-vectors must construct the right-handed coordinates system.
- Case of the RHP frame : The X_ℓ , Y_ℓ , Z_ℓ -directions of a LATTICE correspond to the x,y,z-direction of RHP, respectively.
- Case of the HEX frame : The height vector of a hexagonal prism (4th to 6th BODY data) corresponds to the Z_ℓ -direction of a LATTICE and a normal vector to one of the sides (8th ~ 10th BODY data) corresponds to the X_ℓ -direction of it.
- Case of the RCL frame : The height vector of a cylinder (4th to 6th BODY data) corresponds to the Z_ℓ -direction of a LATTICE and a radial vector (8th to 10th BODY data) corresponds to the X_ℓ -direction of it.

■ Case where the FREE-LATTICE-FRAME option is used.

If the statistical geometry model (LTYP=10) is used, this option is automatically set to be effective.

- The shape of the FRAME ZONE is arbitrary. But the OR operator cannot be used in the combination of BODIes to define the FRAME ZONE.
- The FRAME ZONE must be filled with CELLS for LATTICE type 1 to 4 (rectangular/hexagonal LATTICE).
- For LATTICE type 1 to 4 (rectangular/hexagonal LATTICE), a BODY that has the same size and shape of the LATTICE FRAME must be included in the BODIes to define the FRAME ZONE as in the case where the FREE-LATTICE-FRAME option is not used. This BODY is used to determine the position of the LATTICE in the FRAME ZONE. In addition, the BODY must satisfy either of the following conditions.

- (1) The BODY must be specified first in the list of BODIes to define the FRAME ZONE.
- (2) The BODY has a prefix character string "FRAME=" or "F=" in prior to the BODY ID.

The former case is the same as the case where the FREE-LATTICE-FRAME option is not used. If there is a BODY with "FRAME=" or "F=", the BODY is not necessary to be specified first and the BODY is used to place a LATTICE only. In this case, the BODY is not used for defining the FRAME ZONE but for positioning the LATTICE in the FRAME ZONE. This explicit specification of the LATTICE FRAME is effective when the shape of a FRAME ZONE is arbitrary. It is because there may be a case where the BODY is not necessary to define the FRAME ZONE. In such a case, this explicit specification may enable the code to process the geometry calculation efficiently in random walk.

10.5 Definition of CELLS

The CELL data must be input in the ZONE data after the definition of all ZONEs that are not included in the CELLS. The data for each CELL must be defined between the line of #CELL and the line of #END CELL. The line of #END CELL can be omitted.

```
#CELL ID( CELL-ID ) TYPE( CELL-type )
definition of ZONES in a CELL
#END CELL
```

- *CELL-ID*:

CELL ID number. These numbers are used to specify KLATT in a LATTICE data.

- *CELL-type*:

Character string to denote the CELL types.

- BOX : Type 1 (rectangular-parallelepiped CELL)
- HEXA : Type 2 (hexagonal-prism CELL)
- STGP : Type 3 (spherical CELL used for a particle in the statistical geometry model)
- MBASE : Type 5 (fictitious CELL used for a matrix in the statistical geometry model)

10.5.1 Definition of ZONES in a CELL

ZONES in a CELL are defined as the ordinary ZONES described in the above section. But the following rules must be kept.

1. The ZONE defined first in the CELL must be a fictitious ZONE to define the CELL boundary. In addition, it must be defined by the following BODies.
 - Rectangular-parallelepiped CELL (CELL type BOX) : “RPP” (“BOX” cannot be used.)
 - Hexagonal-prism CELL (CELL type HEXA) : “RHP” (“HEX” cannot be used.)
 - Spherical CELL (CELL type STGP) : SPH
 - Fictitious CELL (CELL type MBASE) : Arbitrary

All the other ZONES belonging to the CELL must be defined in one of the above BODies. The geometrical data is meaningless and is ignored for the CELLS of CELL type MBASE used for the definition of the statistical geometry model.

2. The material ID number of the fictitious material surrounding the CELL must be “-999”.

10.6 Examples for LATTICE Geometry

Some complicated examples for LATTICE geometry are shown in this section. The following examples are not based on realistic geometry to understand the input data easily. Some material data must be defined in the CROSS SECTION data block before these input data.

Example 1 is the input data for a LATTICE geometry which includes another LATTICE with a different pitch as shown in Fig 10.22.

Example 1:

\$GEOMETRY

```

* ----- GEOMETRY PARAMETERS -----
*   P1 : LATTICE PITICH, HT : ASSEMBLY HEIGHT,
*   P2 : LATTICE PITICH,
*   K1 : CELL (PIN) NUMBER IN X & Y DIRECTIONS
*   K2 : CELL (PIN) NUMBER IN X & Y DIRECTIONS
*   C1,D1 : INNER & OUTER RADIUS OF CLADS FOR FUEL PINS.
*   C2,D2 : INNER & OUTER RADIUS OF CLADS FOR FUEL PINS.
*
% K1 = 11, P1 = 1.829, WH1 = K1*P1
% L1 = 5,          WL1 = L1*P1
% K2 = 7, P2 = 1.220, WH2 = K2*P2
% HT = 44.46
% C1 = 0.625 , C2 = 0.48 , D1 = 0.7085, D2 = 0.5865
% WB = WH1/2+10.0
* ----- LATTICE DEFINITION -----
IDLAT(100)
  LTP(1) NVLAT(<K1> <K1> 1) SZLAT( <P1> <P1> <HT> )
  KLATT( 1 1 1 1 1 1 1 1 1 1 1
         1 1 1 1 1 1 1 1 1 1 1
         1 1 1 1 1 1 1 1 1 1 1
         1 1 1 0 0 0 0 0 1 1 1
         1 1 1 0 0 0 0 0 1 1 1
         1 1 1 0 0 0 0 0 1 1 1
         1 1 1 0 0 0 0 0 1 1 1
         1 1 1 0 0 0 0 0 1 1 1
         1 1 1 1 1 1 1 1 1 1 1
         1 1 1 1 1 1 1 1 1 1 1
         1 1 1 1 1 1 1 1 1 1 1 )
  KSLAT( <K1*K1>(000) )
IDLAT(200)
  LTP(1) NVLAT(<K2> <K2> 1) SZLAT( <P2> <P2> <HT> )
  KLATT( 2 2 2 2 2 2 2
         2 2 2 2 2 2 2
         2 2 2 2 2 2 2
         2 2 2 2 2 2 2
         2 2 2 2 2 2 2
         2 2 2 2 2 2 2
         2 2 2 2 2 2 2 )
  KSLAT( <K2*K2>(000) )
END
* ----- BODY DEFINITION -----
RPP( 10000 <-WH1/2> <WH1/2> <-WH1/2> <WH1/2> 0.0 <HT> )
RPP( 20000 <-WL1/2> <WL1/2> <-WL1/2> <WL1/2> 0.0 <HT> )
RPP( 30000 <-WH2/2> <WH2/2> <-WH2/2> <WH2/2> 0.0 <HT> )
RPP( 40000 <-WB> <WB> <-WB> <WB> 0.0 <HT> )
RPP( 1111 <-P1/2> <P1/2> <-P1/2> <P1/2> 0.0 <HT> )
RPP( 2222 <-P2/2> <P2/2> <-P2/2> <P2/2> 0.0 <HT> )
RCC( 1 0.0 0.0 0.0 0.0 0.0 <HT> <C1> )
RCC( 11 0.0 0.0 0.0 0.0 0.0 <HT> <D1> )
RCC( 2 0.0 0.0 0.0 0.0 0.0 <HT> <C2> )
RCC( 22 0.0 0.0 0.0 0.0 0.0 <HT> <D2> )
END
* ----- ZONE DEFINITION -----

```



```

ETV : : -1000 : -40000
REF : REFLECTOR : 6 : 40000 -10000
LAT1 : LATTICE1 : -100 : 10000 -20000
WG : WATERGAP : 3 : 20000 -30000
LAT2 : LATTICE2 : -200 : 30000
* ----- CELL DEFINITION -----
#CELL ID(1) TYPE(BOX) /* CELL
C11 : : -999 : -1111
C12 : MODERATOR : 6 : 1111 -11
C13 : CLAD1 : 5 : 11 -1
C14 : FUEL1 : 4 : 1
#END CELL
#CELL ID(2) TYPE(BOX) /* CELL
C21 : : -999 : -2222
C22 : MODERATOR : 3 : 2222 -22
C23 : CLAD2 : 2 : 22 -2
C24 : FUEL2 : 1 : 2
#END CELL
$END GEOMETRY

```

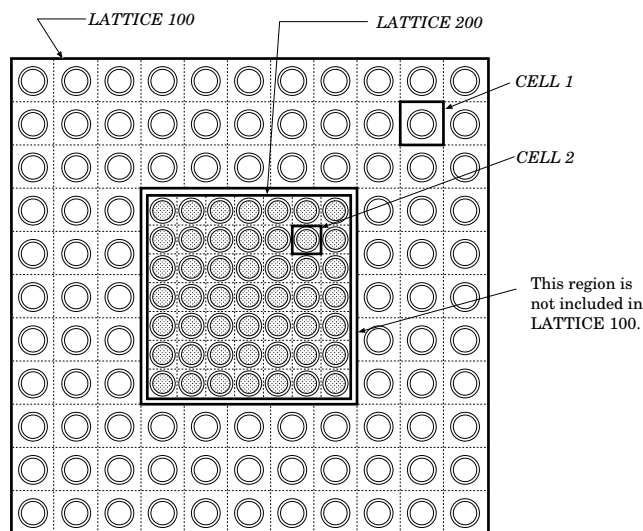


Fig. 10.22 A right-parallelepiped LATTICE geometry which includes another right-parallelepiped LATTICE with a different pitch

- No FREE-LATTICE-FRAME option is required in this case but the LATTICE option must be specified.
- LATTICE 100 is defined between the line IDLAT(100) and the line KSLAT(<K1*K1>(000)). It is a right parallelepiped LATTICE but the central 5×5 SUBFRAMEs are not included in the LATTICE because 25 zero values are specified in the KLATT data.
- LATTICE 100 is placed at the position of BODY 10000 as specified by the line
LAT1 : LATTICE1 : -100 : 10000 -20000.
Note that the central region of 5×5 SUBFRAMEs are excluded by BODY 20000.

- The excluded region (central 5×5 SUBFRAMEs in LATTICE 100) is defined by the following lines.

```
WG : WATERGAP : 3 : 20000 -30000
LAT2 : LATTICE2 : -200 : 30000
```

In this region, another LATTICE (LATTICE 200) is placed at the position of BODY 30000 and the region outside of LATTICE 200 is filled with material 3.

Example 2 is the input data for a right-parallelepiped LATTICE with the FREE-LATTICE-FRAME option as shown in Fig 10.23. A relatively large rod (water rod) is located at the center of the regular arrangement of fuel pins in this example. If this geometry is defined without the FREE-LATTICE-FRAME option as in example 1, the input data is complicated. But it is simplified by the FREE-LATTICE-FRAME option.

Example 2:

```
$GEOMETRY
* ----- LATTICE DEFINITION -----
% NC = 4
% PCH = 1.26206 /* PCH : PITCH OF UNIT PIN CELL
% HT = 1000 /* HT: HEIGHT OF FUEL ASSEMBLY
IDLAT(100)
  LTYP(1) NVLAT(<NC> <NC> 1) SZLAT(<PCH> <PCH> <HT>)
  KLATT( 1 1 1 1
         1 1 1 1
         1 1 1 1
         1 1 1 1 )
  KSLAT(<NC*NC>(000))
END
* ----- BODY DEFINITION -----
% RF0 = 0.39306 /* RF0 : OUTER RADIUS OF FUEL PELLET
% RF1 = 0.45802 /* RF1 : OUTER RADIUS OF Zr clad
% WAH = PCH*NC/2
RPP( 1000 <-WAH> <WAH> <-WAH> <WAH> 0.0 <HT> )
RPP( 100 <-PCH/2> <PCH/2> <-PCH/2> <PCH/2> 0.0 <HT> )
RCC( 200 0.0 0.0 0.0 0.0 0.0 <HT> <RF0> )
RCC( 300 0.0 0.0 0.0 0.0 0.0 <HT> <RF1> )
RCC( 10 0.0 0.0 0.0 0.0 0.0 <HT> 1.30 )
RCC( 20 0.0 0.0 0.0 0.0 0.0 <HT> 1.45 )
END
* ----- ZONE DEFINITION -----
OUTW : : -2000 : -1000
LAT : ASSEMBLY : -100 : 1000 -20
WRODC : WRODCCLAD : 11 : 20 -10
WROD : WROD : 12 : 10
* ----- CELL DEFINITION -----
#CELL ID(1) TYPE(BOX)
C10 : : -999 : -100
C11 : MODERATOR : 12 : 100 -300
C12 : CLADDING : 11 : 300 -200
C13 : FUEL : 3 : 200
#END CELL
$END GEOMETRY
```

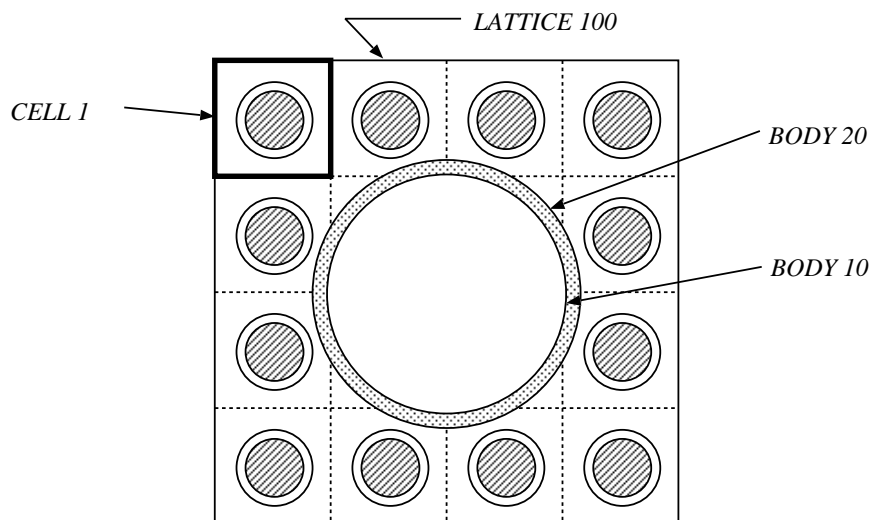


Fig. 10.23 A right-parallellepiped LATTICE with the FREE-LATTICE-FRAME option

- The FREE-LATTICE-FRAME option must be specified.
- LATTICE 100 is defined between the line IDLAT (100) and the line KSLAT (<NC*NC> (000)). It includes 4×4 SUBFRAMEs that are all filled with CELL 1.
- LATTICE 100 is placed at the position of BODY 1000 by the line
 LAT : ASSEMBLY : -100 : 1000 -20
 but the central-rod region is clipped off by BODY 20.
- The central-rod region is defined by the following lines.
 WRODC : WRODCLAD : 11 : 20 -10
 WROD : WROD : 12 : 10

Example 3 is the input data for a hexagonal-prism LATTICE which includes another hexagonal-prism LATTICE as shown in Fig. 10.24. If this geometry is defined without the FREE-LATTICE-FRAME option as in example 1, the input data is very complicated. But it is simplified by the FREE-LATTICE-FRAME option.

Example 3:

```
$GEOMETRY
% HT = 100.0
% P1 = 0.787, DH1 = 17.0*P1/SQRT(3)
% C1 = 0.27, D1 = 0.33124
% P2 = 0.4, DH2 = 4*P1
% C2 = 0.14, D2 = 0.18
* ----- LATTICE DEFINITION -----
IDLAT(100)
  LTP(2) NVLAT( 13 13 1 )
  SZLAT( <P1> 0.0 <HT> )
  SZHEX( 0.0 0.0 <DH1> <HT> 0.0 )
```

```

RCELL( 7 7 )
KLATT(
      2 2 2 2 2 2 2 0 0 0 0 0 0
      2 1 1 1 1 1 1 2 0 0 0 0 0
      2 1 1 1 1 1 1 1 2 0 0 0 0
      2 1 1 1 1 1 1 1 1 2 0 0 0
      2 1 1 1 2 2 2 1 1 1 2 0 0
      2 1 1 1 2 2 2 2 1 1 1 2 0
      2 1 1 1 2 2 2 2 2 1 1 1 2
      0 2 1 1 1 2 2 2 2 1 1 1 2
      0 0 2 1 1 1 2 2 2 1 1 1 2
      0 0 0 2 1 1 1 1 1 1 1 1 2
      0 0 0 0 2 1 1 1 1 1 1 1 2
      0 0 0 0 0 2 1 1 1 1 1 1 2
      0 0 0 0 0 0 2 2 2 2 2 2 2 )
KSLAT( <13*13>(0) )
IDLAT(200)
LTP(2) NVLAT( 11 11 1 )
SZLAT( <P2> 0.0 <HT> )
SZHEX( 0.0 0.0 <DH2> <HT> 0.0 )
RCELL( 6 6 )
KLATT(
      4 4 4 4 4 4 0 0 0 0 0
      4 3 3 3 3 3 4 0 0 0 0
      4 3 3 3 3 3 3 4 0 0 0
      4 3 3 3 3 3 3 3 4 0 0
      4 3 3 3 3 3 3 3 3 4 0
      4 3 3 3 3 3 3 3 3 3 4
      0 4 3 3 3 3 3 3 3 3 4
      0 0 4 3 3 3 3 3 3 3 4
      0 0 0 4 3 3 3 3 3 3 4
      0 0 0 0 4 3 3 3 3 3 4
      0 0 0 0 0 4 4 4 4 4 4 )
KSLAT( <11*11>(0) )
END
* ----- BODY DEFINITION -----
* ... OUTER LATTICE FRAME
RHP( 10000 0.0 0.0 0.0 <HT> <DH1> )
* ... INNER LATTICE FRAME
RHP( 5000 0.0 0.0 0.0 <HT> <DH2> )
* ... PINS FOR LATTICE 100
RHP( 100 0.0 0.0 0.0 <HT> <P1> )
RCC( 110 0.0 0.0 0.0 0.0 0.0 <HT> <C1> )
RCC( 120 0.0 0.0 0.0 0.0 0.0 <HT> <D1> )
* ... PINS FOR LATTICE 200
RHP( 200 0.0 0.0 0.0 <HT> <P2> )
RCC( 210 0.0 0.0 0.0 0.0 0.0 <HT> <C2> )
RCC( 220 0.0 0.0 0.0 0.0 0.0 <HT> <D2> )
END
* ----- ZONE DEFINITION -----
REF : : -2000 : -10000
LAT1 : LATTICE1 : -100 : 10000 -5000
LAT2 : LATTICE2 : -200 : 5000
* ----- CELL DEFINITION -----

```

```
#CELL ID(1) TYPE(HEXA) /* FUEL
  C11 :           : -999 : -100
  C12 : COOLANT   :      3 : 100 -120
  C13 : SUS       :      2 : 120 -110
  C14 : FUEL      :      1 : 110
#END CELL
#CELL ID(2) TYPE(HEXA) /* SODIUM
  C21 :           : -999 : -100
  C22 : COOLANT   :      3 : 100
#END CELL
#CELL ID(3) TYPE(HEXA) /* FUEL
  C31 :           : -999 : -200
  C32 : COOLANT   :      3 : 200 -220
  C33 : SUS       :      2 : 220 -210
  C34 : FUEL      :      1 : 210
#END CELL
#CELL ID(4) TYPE(HEXA) /* SODIUM
  C41 :           : -999 : -200
  C42 : COOLANT   :      3 : 200
#END CELL
$END GEOMETRY
```

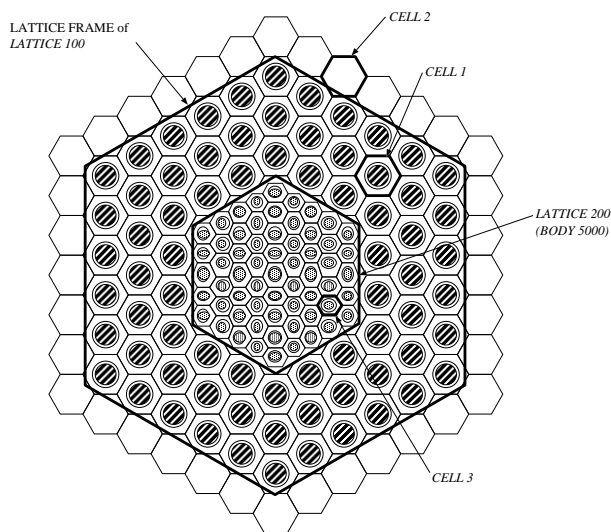


Fig. 10.24 A hexagonal-prism LATTICE geometry which includes another hexagonal-prism LATTICE with a different pitch

- The FREE-LATTICE-FRAME option must be specified.
- LATTICE 100 is defined between the line IDLAT(100) and the line KSLAT(<13*13> (000)). There are 13×13 SUBFRAMEs in the LATTICE. The 3 layers from the center and the outermost 7th layer are filled with CELL 1, and the SUBFRAMEs between them are filled with CELL 2. The other SUBFRAMEs are not included in this LATTICE. The LATTICE FRAME is defined by SZHEX. The width and height of the LATTICE are DH1(=0.33124) and HT(=100.0), respectively.

- LATTICE 100 is placed at the position of BODY 10000 by the line

```
LAT1 : LATTICE1 : -100 : 10000 -5000
```

but the central hexagonal-prism region is clipped off by BODY 5000. Note that the shape of the LATTICE FRAME fits that of BODY 10000.

- The central hexagonal-prism region is defined by the following line.

```
LAT2 : LATTICE2 : -200 : 5000
```

LATTICE 200 is placed at the position of BODY 5000.

- LATTICE 200 is defined between the line IDLAT(200) and the line KSLAT(<11*11> (000)). There are 6 layers (central part of 11×11 SUBFRAMEs) in the LATTICE. The 5 layers from the center and the outermost layer are filled with CELLS 3 and 4, respectively. The outermost layer is used to fill out the peripheral region inside LATTICE 200.

10.7 Tallying in the Same CELLS

The repeated geometry can be represented easily by using a LATTICE and CELLS. However, the tallied data is calculated as the average over all ZONES with the same REGION name even when SUBFRAMEs are different. By using the TALLY-LATTICE or FRAME-DEPENDENT-TALLY option, the user can tally independently for each SUBFRAME even though it contains the same CELL. This option is useful, for example, to tally the pin power distribution in a fuel assembly that consists of the same fuel pin cells.

10.7.1 Labeling FRAME ZONES

The REGION name of a FRAME ZONE means the FRAME ZONE name when the TALLY-LATTICE (or FRAME-DEPENDENT-TALLY) option is used. But the name has no meaning without the option.

10.7.2 Labeling SUBFRAMEs

If the TALLY-LATTICE (or FRAME-DEPENDENT-TALLY) option is specified, the labeling data for SUBFRAMEs must follow the ZONE definition data for each FRAME ZONE like the following example.

```
#SUBFRAME
  NAMES( S1 S2 S3 M4 )
  SPACE( 1 1 1 1 4 2 2 2 2
         1 1 1 1 4 2 3 3 2
         1 1 1 1 4 2 3 3 2
         1 1 1 1 4 2 2 2 2
         4 4 4 4 4 4 4 4 4
         2 2 2 2 4 1 1 1 1
         2 3 3 2 4 1 1 1 1
         2 3 3 2 4 1 1 1 1
         2 2 2 2 4 1 1 1 1 )
#END SUBFRAME
```

The following two types of data must be input between “#SUBFRAME” and “#END SUBFRAME”.

- NAMES

List of SUBFRAME names. The SUBFRAMEs are labeled by the corresponding ordinal number in this list. Each SUBFRAME name must be a character string up to 12 alphabetic or numerical characters. If this input is omitted, values of the following SPACE data are interpreted as the SUBFRAME names.

- SPACE

Each SUBFRAME name is specified by the ordinal number in the NAMES list. The input order is the same as KLATT in the LATTICE data. The value of SPACE must be 0 for SUBFRAMEs which are not included in a LATTICE (KLATT=0).

Using this labeling option, the REGION name of ZONEs in the specified space is defined with the FRAME ZONE names and the SUBFRAME names as follows.

!FRAME ZONE name:SUBFRAME name[!FRAME ZONE name:SUBFRAME name!...]!REGION name in the ZONE input

The names in the [] are used for more than one hierarchical LATTICE geometry (LATTICE in LATTICE).

The following specification similar to CELLIDS, CELLSYMS, CKLATT in LATTICE data is also available for the input of SUBFRAME names.

```
#SUBFRAME
  NAMES( S1 S2 S3 M4 )
  CNAMES ( A B C M )
  CSPACE( AAAA M BBBB
           AAAA M BCCB
           AAAA M BCCB
           AAAA M BBBB

           MMMM M MMMM

           BBBB M AAAA
           BCCB M AAAA
           BCCB M AAAA
           BBBB M AAAA )
#END SUBFRAME
```

- CNAMES

List of characters delimited by one or more blanks. The characters cannot be multiply defined. The number of components in CNAMES must be the same as that of NAMES. Each character corresponds to the SUBFRAME names in NAMES uniquely. The available characters are capital/small alphabetical letters, numbers and symbols (except for parentheses, apostrophe, “/”, “*”).

- CSPACE

Specification of the SUBFRAME names by using the one-character symbols defined in CNAMES. The rule of the input order is the same as SPACE. Blanks, enter code and comment lines/comment character strings are meaningless and ignored.

10.8 Input Data for Statistical Geometry Model

The spatial region which includes the statistical geometry model is called the “STGM region” and spherical particles probabilistically placed in the STGM region are called “STG particles”. To use the model, the following must be specified.

- Use a LATTICE geometry (LATTICE option).
- The STGM region is defined as a LATTICE and the LATTICE type must be 10 (LTYP=10). In this LATTICE, STG particle region and of the STGM region exclusive of the particles (matrix region) are treated as CELLS.
- The STG particles must be defined as CELLS of type “STGP”.
- The matrix medium in the STGM region must be defined as a CELL of type “MBASE”.

10.8.1 Definition of the STGM Region

The LATTICE geometry is defined by the LATTICE ID, the LATTICE type, the size of the LATTICE and CELLS in the LATTICE before the input of BODY data in the GEOMETRY data block. The STGM region is also defined as one of the LATTICE geometries but only the LATTICE ID (IDLAT) and the LATTICE type (LTYP) are the same as the other LATTICE geometries. The other input data are specific to the statistical geometry model. Any number of the STGM regions can be defined as LATTICES. Multiple CELLS of type STGP for different STG particles (not more than 8) can be arranged in a STGM region if the size of the particles are the same.

```

IDLAT ( LATTICE-ID )          /* LATTICE ID
LTYP ( 10 )                    /* LATTICE type = 10
PF ( packing-fraction )      /* Packing fraction
MBASE ( CELL-ID-of-matrix )   /* CELL ID of the MATRIX region
CELL ( CELL-ID-of-STG-particle-1 CELL-ID-of-STG-particle-2 ... )
                                /* CELL ID's of the STG particles
PCELL ( fraction-of-STG-particle-1 CELL-ID-of-STG-particle-2 ... )
                                /* Fraction of STG particles
[ WNND1 ( NND1-mesh-relative-to-STG-diameter )
                                /* Mesh width to define NND1 ( relative value to the diameter of STG )
FNND1 ( NND1 ... )
                                /* Cumulative distribution for NND1
WNND2 ( NND2-mesh-relative-to-STG-diameter )
                                /* Mesh width to define NND2 ( relative value to the diameter of STG )
FNND2 ( NND2 ... )
                                /* Cumulative distribution for NND2
WNND3 ( NND3-mesh-relative-to-STG-diameter )
                                /* Mesh width to define NND3 ( relative value to the diameter of STG )
FNND3 ( NND3 ... )
                                /* Cumulative distribution for NND3 ]
[ COPYNND ( STGM-region-ID-whose-NND-is-copied )
                                /* STGM region ID whose NND is copied ]

```

(1) IDLAT

LATTICE ID of the STGM region. It must be an arbitrary positive integer.

(2) LTYP

LATTICE type of the STGM region. It must be 10.

(3) PF

Packing fraction of the STG particles in the STGM region. $0 < PF < 1$.

(4) MBASE

ID number of the CELL to specify the matrix medium where the STG particles are distributed. A CELL of this type must be TYPE(MBASE) and includes two ZONES. The material ID of the ZONE defined first in the CELL must be -999 as well as that of normal CELLS and the second material is the matrix medium in the STGM region. The shape of the ZONE is arbitrary and is ignored in particle tracking.

(5) CELL

CELL ID of the STG particles. Multiple CELL ID's (not more than 8) can be specified but the size (radius) of the STG particles must be the same.

(6) PCELL

Relative fractions of multiple kinds of STG particles (partial packing fraction). The sum of the fractions is normalized to unity internally. No PCELL data is required for one kind of STG particles.

(7) WNND1, FNND1, WNND2, FNND2, WNND3, FNND3

Nearest Neighbor Distributions (NNDs). The following 3 distributions output from the MCRDF code⁵³⁾ are available.

- NND1 : NND for particles which enter the MATRIX region from the STG particle regions.
- NND2 : NND for particles scattered in the MATRIX region.
- NND3 : NND for particles which enter the STGM region from the boundary.

The NNDs are defined in the table form of cumulative probabilities to the distance to the nearest STG particles. The distance is given by the relative value to the diameter of the STG particles and the probability is given by an equal distance in the table. WNND1, WNND2, WNND3 are the unit of the distance for each NND. FNND1, FNND2, FNND3 are the cumulative probability distributions for each NND.

If no data is given to some NNDs, the analytical nearest neighbor function is employed to define the NNDs. The function $F(r)$ is given by

$$\frac{dF(r)}{dr} = \frac{3}{2} \cdot \frac{f_p}{1 - f_p} \exp \left(-\frac{3}{2} \cdot \frac{f_p}{1 - f_p} r \right),$$

where r is the distance relative to the diameter of the STG particle and f_p is the packing fraction of the STG particles.

(8) COPYNND

FNND1, FNND2 and FNND3 are copied from those of the LATTICE ID defined in this variable. The LATTICE (STGM region) to be copied must be defined in prior to the STGM region specifying COPYNND.

10.8.2 Allocation of STGM Regions

If a rectangular or hexagonal LATTICE is allocated in a ZONE, the ZONE must have the material ID which indicates the LATTICE ID ($-1 \times \text{LATTICE ID}$ or “LAT=LATTICE ID” or “L=LATTICE ID”). Likewise, when the STGM region is allocated in a ZONE, the ZONE must have the material ID which is a negative sign of LATTICE ID or “LAT=LATTICE ID” or “L=LATTICE ID”. The different points from the other LATTICES are as follows.

- (1) The shape of ZONES where the STGM region is allocated is arbitrary.
- (2) The STGM region with the same LATTICE ID can be allocated in different ZONES whose shapes may be different.
- (3) If the TALLY-LATTICE option is specified, the FRAME ZONE including the STGM region (STGM ZONE) must have the #SUBFRAME data sub-block as well as the ordinary FRAME ZONES. The number of SUBFRAMEs must be $N + 1$; N is the number of CELLS of type STGP for STG particles and the rest is a CELL of type MBASE for a matrix region. The SUBFRAME names for STGP CELLS must be specified in the order of the CELL ID's specified in the CELL parameter. The last SUBFRAME name is for the matrix region.

Example of definition of the STGM region:

```

/** LATTICE definition section
  *** STG region as LATTICE 200
  IDLAT(200) LTYP(10) PF(0.3) MBASE(1027) CELL(123)

  ....

  /** ZONE definition section
  STG1: STG1 : -200 : 1 -2 -3 4 /* FRMAE ZONE(1) for
                                /* STG region with -200

  #SUBFRAME
  /** Specify SUBFRAME names in TALLY-LATTICE mode
  NAMES( STG MATRIX ) SPACE( 1 2 )
  #END SUBFRAME

  STG2: STG2 : -200 : 1234 -345 -678 /* FRAME ZONE(2) for
                                /* STG region with -200

  #SUBFRAME
  NAMES( STG MATRIX ) SPACE( 1 2 )
  #END SUBFRAME

```

10.8.3 Definition of STG CELLS

The STG particles are defined as a kind of the CELLS (STGP CELLS) in the data sub-block between #CELL and #END CELL of the ZONE input section. The type of CELLS must be “STGP” and the BODY of SPH (sphere) must be used for the firstly defined ZONE whose material ID is -999 (outside of the CELL).

Example of a STGP CELL:

```

SPH( 111  0 0 0 <R> ) /*** a spherical BODY
....

#CELL ID(123) TYPE( STGP )
      :      : -999 : -111
LAYER1: L1 : <MAT1> : 111 -112
LAYER2: L2 : <MAT2> : 112 -113
LAYER3: L3 : <MAT3> : 113
#END CELL

```

10.8.4 Dummy CELL for MATRIX

A dummy CELL must be defined to specify the material and REGION name of the matrix medium where the STG particles are distributed. The CELL type (TYPE) must be “MBASE” and the CELL includes two ZONES. The material ID of the firstly defined ZONE in the CELL must be -999 as well as normal CELLS and that of the ZONE defined next is the material of the medium. The REGION name of the second ZONE is that of the medium. The shape of the ZONES included in this type of CELLS is arbitrary and is ignored in tracking particles.

Example of the CELL which defines a matrix medium:

```

RPP( 1000  0 1 0 1 0 1 ) /*** arbitrary BODY for dummy CELL
....

#CELL ID(1027) TYPE( MBASE )
      :      :      : -999 : -1000
MATRIX : MTX : <MATRIX> : 1000
#END CELL

```

10.8.5 Example for Multiple Kinds of STG Particles

An example for multiple kinds of STG particles with the statistical geometry model is shown in this section. The example is the input data for a pebble-bed reactor core as shown in Fig 10.25. The core consists of 2 kinds of pebbles (fuel and moderator pebbles) and one of the pebbles (fuel pebble) includes a number of coated fuel particles.

Example for a pebble-bed reactor core:

```

$GEOMETRY
% HT = 1000.0
% RRR = 90.0
***** LATTICE DATA *****
IDLAT(1) /* CFP region in a pebble bed
      LTYP(10) PF( 0.05 ) MBASE( 1 ) CELL( 2 )
IDLAT(10) /* mixture (fuel & dummy pebble) region
      LTYP(10) PF( 0.61 ) MBASE( 10 ) CELL( 11 12 ) PCELL( 0.57 0.43 )
END
***** BODY DATA *****

```

```

CYL(   100   0. 0. 0.  <HT>  <RRR> )
SPH(   102   0. 0. 0.  2.5 ) /* pebble bed (fuel ZONE)
SPH(   101   0. 0. 0.  3.0 ) /* pebble bed
SPH(    15   0. 0. 0.  0.0455 )
SPH(    14   0. 0. 0.  0.0415 )
SPH(    13   0. 0. 0.  0.038 )
SPH(    12   0. 0. 0.  0.034 )
SPH(    11   0. 0. 0.  0.025 )
END
***** ZONE definition *****
  OUTV      :           : -3000 : -100
  CORE      : MIXTURE   :    -10 :  100
#SUBFRAME
  NAMES( FUEL DUMMY MATRIX ) SPACE( 1 2 3 )
#END SUBFRAME
#CELL ID(1) TYPE(MBASE) /* matrix
  C11      :           : -999 : -102
  MATRIX : MATRIX      :    306 :  102
#END CELL
#CELL ID(2) TYPE(STGP) /* CFP
  C11      :           : -999 : -15
  C12      : OPYC       :    305 :  15 -14
  C13      : SIC        :    304 :  14 -13
  C14      : IPYC       :    303 :  13 -12
  C15      : BUFFER     :    302 :  12 -11
  C16      : FUEL       :    301 :  11
#END CELL
#CELL ID(10) TYPE(MBASE) /* helium
  C11      :           : -999 : -100
  MATRIX : HELIUM      :    203 :  100
#END CELL
#CELL ID(11) TYPE(STGP) /* fuel pebble bed
  C11      :           : -999 : -101
  SHELL    : SHELL     :    306 :  101 -102
  CFP      : CFP       :     -1 :  102
#SUBFRAME
  NAMES( FUEL MATRIX ) SPACE( 1 2 )
#END SUBFRAME
#END CELL
#CELL ID(12) TYPE(STGP) /* dummy pebble bed
  C11      :           : -999 : -101
  C12      : DUMMY     :    202 :  101
#END CELL
$END GEOMETRY

```

- LATTICE 1 is defined for fuel pebbles including coated fuel particles (CFPs). It stands for the statistical geometry model because LTYPE is 10. The STG particle (CFP) is defined in CELL 2 and the matrix region is defined in CELL 1. The packing fraction of the STG particles is 5%. The analytical nearest neighbor distribution is used for NND1 through NND3.
- LATTICE 10 is defined for a core including 2 kinds of pebbles (fuel and moderator pebbles). It is also a LATTICE of the statistical geometry model where 57% of STG particles of CELL 11 and 43% of those of CELL 12 are randomly distributed in a matrix medium of CELL 10. The packing

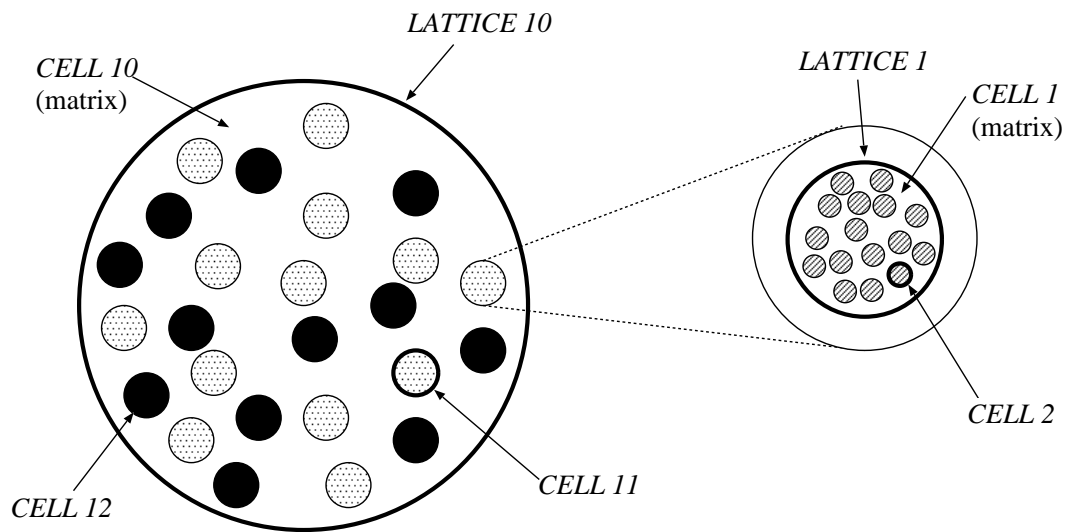


Fig. 10.25 An example for multiple kinds of STG particles

fraction of the STG particles (mixture of CELL 11 and CELL 12) is 61%. The analytical nearest neighbor distribution is used for NND1 through NND3.

- The ZONE for the core is defined by the following line.

```
CORE : MIXTURE : -10 : 100
```

LATTICE 10 is placed at the position of BODY 100. The SUBFRAME names are defined for this LATTICE. The SUBFRAME names for CELL 11, CELL 12 and CELL 10 are FUEL, DUMMY and MATRIX, respectively.

- CELLS 1 and 2 are defined for the matrix medium and CFPs in the fuel pebbles, respectively.
- CELL 10 is defined for the matrix medium in the core. CELLS 11 and 12 are defined for fuel and moderator pebbles, respectively. CELL 11 includes LATTICE 1 represented by the statistical geometry model.

10.9 Definition of TALLY REGIONS

‘TALLY REGION’ is a spatial REGION to tally and can be defined by one REGION or combination of more than one REGION. The statistical results by REGION and the combined results of them can be obtained simultaneously by this definition. If this data sub-block is omitted, each REGION corresponds to a TALLY REGION. The TALLY REGION data must be specified in the data sub-block which begins with “#TALLY REGION” at the end of the GEOMETRY data block.

```
#TALLY REGION
definition-type @TALLY-REGION-Name ( list-of-REGIONS )
...
...
(repeat as necessity requires)
$END GEOMETRY
```

- (1) The TALLY-REGION data sub-block must begin with “#TALLY REGION” at the end of the GEOMETRY data block.
- (2) There are two types to define the TALLY REGIONS.
 - ADD
A new TALLY REGION is defined as the sum of REGIONS (p.49) specified in the list of REGION names (p.163) in addition to each REGION. Namely, the codes output tallied data in each REGION and the sum of the REGIONS.
 - DEFINE
This definition type defines a new TALLY REGION as the sum of REGIONS specified in the list of REGION names. The codes output tallied data in the sum of the REGIONS but does not output tallied data in each REGION. This type is useful to re-define minimum and necessary TALLY REGIONS.
- (3) If the definition type is omitted, the previously defined type is effective. If no type is specified for all TALLY REGION data, the definition type is interpreted as ADD.
- (4) The TALLY REGION name must be a character string which begins with “@” and is up to 12 alphabetic or numeric characters excluding special characters such as “!”, “.”.
- (5) The TALLY REGIONS whose names begin with “@” are distinguished from the normal REGIONS and the results in the TALLY REGIONS are output in prior to those in the normal REGIONS.
- (6) A single REGION can belong to different TALLY REGIONS.
- (7) The list of REGION NAMES must be described by the hierarchical expression of REGION names (p.70).

Example:

```
$GEOMETRY
* ----- LATTICE DEFINITION -----
% NC = 3
% PCH = 1.26206 /* PCH : PITCH OF UNIT PIN CELL
% HT = 1000 /* HT: HEIGHT OF FUEL ASSEMBLY
IDLAT(100)
  LTP(1) NVLAT(<NC> <NC> 1) SZLAT( <PCH> <PCH> <HT> )
  KLATT( 1 1 1
        1 1 1
        1 1 1 )
  KSLAT( <NC*NC>(000) )
END
* ----- BODY DEFINITION -----
% RF0 = 0.39306 /* RF0 : OUTER RADIUS OF FUEL PELLET
% RF1 = 0.45802 /* RF1 : OUTER RADIUS OF Zr clad
% WAH = PCH*NC/2
RPP( 1000 <-WAH> <WAH> <-WAH> <WAH> 0.0 <HT> )
RPP( 100 <-PCH/2> <PCH/2> <-PCH/2> <PCH/2> 0.0 <HT> )
RCC( 200 0.0 0.0 0.0 0.0 0.0 <HT> <RF0> )
RCC( 300 0.0 0.0 0.0 0.0 0.0 <HT> <RF1> )
END
```

```

* ----- ZONE DEFINITION -----
OUTW  :           : -2000 : -1000
LAT   : ASSEMBLY  :   -100 :  1000
#SUBFRAME
NAMES( PIN01 PIN02 PIN03
       PIN04 PIN05 PIN06
       PIN07 PIN08 PIN09 )
SPACE(  1  2  3
       4  5  6
       7  8  9 )
#END SUBFRAME
* ----- CELL DEFINITION -----
#CELL ID(1) TYPE(BOX)
C10   :           :   -999 :  -100
C11   : MODERATOR :    12 :   100 -300
C12   : CLADDING  :    11 :   300 -200
C13   : FUEL      :     3 :   200
#END CELL
#TALLY REGION
DEFINE @PIN01( !ASSEMBLY:PIN01!FUEL ) /* fuel region in PIN01
DEFINE @PIN02( !ASSEMBLY:PIN02!FUEL ) /* fuel region in PIN02
DEFINE @PIN03( !ASSEMBLY:PIN03!FUEL ) /* fuel region in PIN03
DEFINE @PIN04( !ASSEMBLY:PIN04!FUEL ) /* fuel region in PIN04
DEFINE @PIN05( !ASSEMBLY:PIN05!FUEL ) /* fuel region in PIN05
DEFINE @PIN06( !ASSEMBLY:PIN06!FUEL ) /* fuel region in PIN06
DEFINE @PIN07( !ASSEMBLY:PIN07!FUEL ) /* fuel region in PIN07
DEFINE @PIN08( !ASSEMBLY:PIN08!FUEL ) /* fuel region in PIN08
DEFINE @PIN09( !ASSEMBLY:PIN09!FUEL ) /* fuel region in PIN09
DEFINE @ALLCELL( !ASSEMBLY:*!* )
$END GEOMETRY

```

11. Source Data

11.1 General Rules for Source Input Data

11.1.1 SOURCE Data Block

The source parameters must be input in the following data block.

```

$SOURCE
options or parameters common to all sources
&
the first source information set
&
the second source information set
&
...
$END SOURCE

```

- “\$” and “&” must be located at the first column of an input line and a source information set is terminated by the next “&” or “\$END SOURCE”.
- It is recommended that this data block is located after the GEOMETRY and CROSS SECTION data blocks because symbolic parameters defined in these blocks can be used in the SOURCE data block.
- There are only 2 parameters for common data to all source information sets. Normally the default values are available for the parameters.

$NWORK(n)$: The size of the work area for processing each source information set. The user must specify it in the word unit (n word). The default value is 4,096 ($NWORK = 4,096$).

$MXREJ(n)$: The number of maximum times for rejection with ACCEPT and END-ACCEPT (n times). The default value is 300 ($MXREJ = 300$).

11.1.2 Source Information Set

Each source information set requires the following data.


```

& [NEUTRON or PHOTON]
  [ID ( source-set-ID )]
  RATIO ( source-strength-ratio )

@parameter-name = sampling-information ;

@ ( parameter-name [parameter-name ...] ) = sampling-information ;

ACCEPT [ ; ]
  @parameter-name = sampling-information ;
  WHEN condition-to-accept-sample ;
END-ACCEPT [ ; ]

```

(1) NEUTRON or PHOTON

Particle type to be generated. It can be omitted for a neutron- or photon-only problem.

(2) ID

This is defined when the ID number (positive integer) is specified for the source information set. Without specifying ID's, the sets are numbered in the order of input data in the SOURCE data block. The same ID number can be used to specify the multiple source information sets. The ID numbers are used to tally separately particles from different sources.

(3) RATIO

Generation ratio of each source set in the case of multiple sets. (not necessary to be normalized.)

(4) @*parameter-name* = *sampling-information* ;

This line specifies how to sample source parameters (source position, energy, etc.). This line is called “sampling specification line” hereafter.

The source parameters to be sampled are listed in Table 11.1. The unit of the sampled parameters is “cm” for the spatial variable (X, Y, Z), “eV” for energy, “s” for time and dimensionless for the others. The default value is available for some parameters. The default values are listed in Table 11.2.

The user can make use of any variable name as a work variable which begins with an alphabet and are up to 8 characters excluding the parameters in Table 11.1. But the variable names which begin with WHEN or SAVE are unavailable because they are used internally in the codes.

If only one parameter is sampled by sampling information, the parameter must be specified as

```
@parameter name = sampling information;
```

If some parameters are sampled simultaneously, the parameters must be specified as

```
@(parameter name [parameter name ...] ) = sampling information;
```

The combination of parameters is arbitrary.

Table 11.1 Source parameters

Parameter name	Description	Unit
X	Source position in the x -direction	cm
Y	Source position in the y -direction	cm
Z	Source position in the z -direction	cm
A	Source direction (direction cosine along the x -direction)	
B	Source direction (direction cosine along the y -direction)	
C	Source direction (direction cosine along the z -direction)	
E or G	Energy or Energy group	eV
T	Time	s
W	Weight	

Table 11.2 Default values for source parameters

Parameter	Default
X, Y, Z	No default value. Always required.
A, B, C	Isotropic source. But if only one or two parameters among A, B, C are specified, a fatal error occurs.
E or G	No default value. Always required.
T	0 s
W	1.0

11.2 Representation of Sampling Information

The “sampling information” for each parameter consists of the following data.

- How to sample the parameters

```
Value;
Mathematical expression;
Sampling or data-transform function;
```

- To manipulate the sampled parameters

```
Data-transform function;
```

- To select the sampled parameters (rejection sampling by rejection method)

11.3 Values and Mathematical Expression

The “value” is a numerical constant and the mathematical expression includes numerical constants, symbolic parameters, basic functions and operators which are available in the MVP/GMVP input. In addition, source-parameter names or work-variable names which begin with “@” are also available.

If the symbolic parameters are referred to in the SOURCE data block, “< >” is not always required as in the other data block. The source parameters or the variables must be defined before they are referred to.

Example :

```
@E = 1.4E+7;

% X0 = 10.0

@X = X0;
@Y = <X0/2>;
@Z = SQRT(3*X0 - @X**2 - @Y**2);
```

11.4 Sampling with Functions

“Sampling functions” are used to sample one or more than one value from some distribution and ‘data-transform functions’ to transform the sampled value to other values in the case where it is difficult to represent the sample distribution and so on. The format is as follows.

```
#Function-name( data );
```

or

```
#Function-name data-name( data ) [ data-name( data )... ];
```

The function name must begin with “#”. (Without “#”, it is possibly interpreted as an ordinary mathematical expression or a numerical function and then a fatal error may occur.)

Let us consider a cylindrical source as an example of sampling functions. When the central axis of the cylindrical source is parallel to the z -axis, the pair of (X,Y) and Z of the source point can be sampled independently. Then

- Sample (X, Y) from a uniform distribution in a circle.
- Sample Z from a uniform distribution in the bounded range of the height.

For example, suppose the central axis passes through (0,0), and source particles are generated from the circular ring of $R1 < r < R2$ and the bounded height of $Z1 < z < Z2$, the source can be specified as follows.

```
$SOURCE
& NEUTRON      RATIO(1.0)
  @( X Y ) = #DISC( <R1> <R2> );
  @Z = #UNIFORM( <Z1> <Z2> );
  @E = #MAXWELL(1.4727E6);
$END SOURCE
```

The “#DISC” function is used to sample the pair of (X,Y) coordinates from the uniform distribution in the ring.

#DISC : Uniform distribution in a circle (or a ring)

<input form>

```
@( parameter1 parameter2 ) = #DISC( r1 r2 [angle1 angle2] );
```

This function samples a pair of coordinates from a uniform distribution in a ring. (radius range $r1 < r < r2$) If *angle1* and *angle2* are specified, the pair is sampled between the central angle satisfying ($angle1 < \theta < angle2$), where the *x*-axis is the base line of the angle θ . The unit of the angle is degree and it is defined counter-clockwise.

The #UNIFORM function samples a value from a uniform distribution between two values.

#UNIFORM : Uniform distribution between two values

<input form>

```
@parameter = #UNIFORM( bound1 bound2 );
```

A value is sampled from a uniform distribution between *bound1* and *bound2*. The order of *bound1* and *bound2* is arbitrary.

When the energy value is sampled by using the #MAXWELL function, the following is an example of the specification.

#MAXWELL : Maxwellian distribution

<input form>

```
@parameter = #MAXWELL( T );
```

The square of the speed or the kinetic energy is sampled from the Maxwellian distribution ($T > 0$).

$$P\{x|0 < x < \infty\} \propto \sqrt{x}e^{-\frac{x}{T}}$$

The direction cosines (A,B,C) are not specified and thus are sampled from an isotropic distribution.

The above example is available with R1=R2. But in this case, the pair (X Y) can be also specified as follows.

```
@( X Y ) = #CIRCLE( <R1> );
```

In the other case where the central axis of the cylinder passes through (X0,Y0), not (0,0), the sampled values can be transferred in parallel.

```
$SOURCE
& NEUTRON      RATIO(1.0)
  @( X Y ) = #DISC( <R1> <R2> );
  @X = @X + <X0>;
  @Y = @Y + <Y0>;
  @Z = #UNIFORM( <Z1> <Z2> );
  @E = #MAXWELL(1.4727E6);
$END SOURCE
```

The available sampling and data-transform functions are listed in Tables 11.3 and 11.4, respectively.

Table 11.3 Sampling function

Name	Function	Number of parameters
#TABLE	Sampling from a set of table data	1
#TABLEXY	Conditional sampling from a set of table data	2
#UNIFORM	Uniform distribution between two values	1
#COSINE	Cosine distribution	1
#WATT	Watt's fission spectrum	1
#GAUSS	Gaussian	1
#MAXWELL	Maxwellian energy	1
#EVAPORATION	Evaporation	1
#POWER	Power distribution	1
#FISSION	Fission spectrum	2
#DISC	Uniform distribution in a circular disk	2
#CIRCLE	Uniform distribution on a circle	2
#SPHERE	Uniform distribution in a sphere	3
#ISOTROPIC	3-dimensional isotropic vector	3
#FISSIONFILE	Neutron source file from other calculation	7 or 8
#FEYNMAN	Sampling a time for a spontaneous fission source	4
#TYPE _n	Old forms of source specification	6 or 7
#USER _n	User-defined sampling function	arbitrary

Table 11.4 Data-transform function

Name	Function	Number of parameters
#SWAP	Exchange two parameters	2
#ROTATE2D	Rotation of a 2-dimensional vector	2
#ROTATE3D	Rotation of a 3-dimensional vector	3
#RESIZE	Adjust the length of a 3-dimensional vector	3
#NEWAXIS	Transform the coordinate system for a 3-dimensional vector	3
#NORMALIZE	Normalize the sum of sampled parameters	1

11.5 Sampling Functions

#TABLE

■ Function ■

A value is sampled from a set of table data. The table data are pairs of a data point and probability at the point.

■ Format ■

@parameter = #TABLE X(data-points) PDX or PX(probabilities) [INT(interpolation-method)] ;

■ Description ■

- PDX (or PX) is the probability density for X. No normalization is required for PDX (and PX). If the interpolation method is given with STEP or LINEAR, the user inputs a probability density per unit X to PDX. If it is given with LOG-LINEAR or LOG-STEP, the user inputs a probability density per log X (natural logarithm of X) to PDX. If the interpolation method is DISCRETE, there is no difference between PDX and PX. PX is the probability itself. Since there is an ambiguity in using LINEAR or LOG-LINEAR, the user should use PDX in general. X must be input in either ascending or descending order (the same X values can be input). In addition, the input order of X, PDX (or PX) and INT is arbitrary.
- INT specifies the interpolation method for the probability PDX (or PX). It must be specified by a character string of the following five ones.

STEP DISCRETE LINEAR LOG-STEP LOG-LINEAR

(a) STEP:

Supposing the sampled value is x , the relative probability that x satisfies $X(i) < x < X(i+1)$ is $PDX(i) \times |X(i+1) - X(i)|$ (or $PX(i)$) for each i and x is equally sampled between $X(i)$ and $X(i+1)$.

The equation $\langle \text{the number of PDX (or PX)} \rangle = \langle (\text{the number of X}) - 1 \rangle$ must be satisfied.

If no INT(. . .) is specified and the equation $\langle \text{the number of PDX (or PX)} \rangle = \langle (\text{the number of X}) - 1 \rangle$ is satisfied, INT is set to STEP.

(b) DISCRETE:

The relative probability of $x = X(i)$ is $PDX(i)$ (or $PX(i)$). The equation $\langle \text{the number of PDX (or PX)} \rangle = \langle \text{the number of X} \rangle$ must be satisfied. If no INT(. . .) is specified and the equation $\langle \text{the number of PDX (or PX)} \rangle = \langle \text{the number of X} \rangle$ is satisfied, INT is set to DISCRETE.

(c) LINEAR:

Supposing the sampled value is x , the relative probability that x satisfies $X(i) < x < X(i+1)$ is $(PDX(i) + PDX(i+1)) \times |X(i+1) - X(i)|$ (or $PX(i) + PX(i+1)$) for each i . The distribution of possible x values between $X(i)$ and $X(i+1)$ is proportional to the one that interpolates linearly between $PDX(i)$ and $PDX(i+1)$ (or $PX(i)$ and $PX(i+1)$). Namely, PX is treated as the probability density for sampling $X(i) < x < X(i+1)$. It should, however, be noted that PX is not the probability density except for the case where the values of $X(i)$ are equally spaced. It is recommended to use $PDX(i)$ in general.

$$P\{x|X(i) < x < X(i+1)\} \propto PDX(i) + \frac{x - X(i)}{X(i+1) - X(i)}(PDX(i+1) - PDX(i))$$

(For PX, PDX is replaced with PX in the equation above.)

The equation $\langle \text{the number of PDX (or PX)} \rangle = \langle \text{the number of X} \rangle$ must be satisfied.

(d) LOG-STEP:

$\log(x)$ is sampled in the same way as STEP and then x is obtained from $e^{\log(x)}$. The relative probability such that $X(i) < x < X(i+1)$ is $PDX(i) \times |\log(X(i+1)) - \log(X(i))|$ (or $PX(i)$) for each i .

$\langle \text{the number of PDX (or PX)} \rangle = \langle \text{the number of X} - 1 \rangle$ must be satisfied.

(e) LOG-LINEAR:

$\log(x)$ is sampled in the same way as LINEAR and then x is obtained from $e^{\log(x)}$. The relative probability such that $X(i) < x < X(i+1)$ is $(PDX(i) + PDX(i+1)) \times |\log(X(i+1)) - \log(X(i))|$ (or $PX(i) + PX(i+1)$) for each i .

$\langle \text{the number of PDX (or PX)} \rangle = \langle \text{the number of X} \rangle$ must be satisfied.

Example:

```
@X = #TABLE  X(0.0 1.0)  PDX(1.0);
@T = #TABLE  INT(DISCRETE) X(0.0 1.0E-3)  PDX(2 1);
@E = #TABLE  X(1.0E-5 10.0)  PDX(1.0 0.0)  INT(LOG-LINEAR);
```

#TABLEXY**■ Function ■**

Table function of 2 parameters which depend on each other. Two variables are sampled simultaneously from the table of 2-dimensional sampling points and the probabilities.

$$P(x, y) = P(x)P(y|x)$$

The first parameter x is sampled ($P(x) \rightarrow x$) and then the second parameter y is sampled ($P(y|x) \rightarrow y$).

■ Format ■

```
@(parameter1 parameter2 ) = #TABLEXY
    [INTX( x-interpolation-method )]
    X( x-data-points ) PDX( x-probabilities )
    [ INTY( y-interpolation-method ) ]
    Y( y-data-points ) PDY( y-probabilities )
    [ INTY( y-interpolation-method ) ]
    Y( y-data-points ) PDY( y-probabilities )
    ...
;
```

At first, data X for x must be specified in the same way as the #TABLE function. The interpolation method must be specified by INTX, not by INT. Next, the sampling information for y must be given by some sets of Y, PDY and INTY (if necessary) for each X or each pair of adjacent 2 X's in the case of INTX(DISCRETE) or except for INTX(DISCRETE), respectively. (It should be noted that the user can specify PY instead of PDY but the user cannot specify both PDY and PY for the same X. PDY is

specified for one X, and PY can be specified for the other X.) The interpolation method for each X or Y is specified in the same way as #TABLE. Once the interpolation method INTY for Y is specified, it is effective until the next INTY is specified.

■ Example ■

Sampling of the direction vector from the table of the direction cosines and the rotation angles. Since there are four pairs of (X, PX), corresponding four pairs of (Y, PDY or PY) are given.

```
@(A PHAI) = #TABLEXY INTX(LINEAR)
X(-1 -0.5 0.0 0.6 1.0)
PDX(0.1 0.2 0.1 0.8 0.5)
Y(0.0 1.0)      PY(1.0)
Y(0.0 0.5 1.0) PDY(0.2 0.8)
Y(0.0 1.0)      PDY(1.0)
Y(0.0 0.6 1.0) PY(1.0 2.0);

% PAI2=2*ACOS(-1)
@C = SQRT(1.0-@A**2);
@B = @C*COS(PAI2*@PHAI);
@C = @C*SIN(PAI2*@PHAI);
```

In this example, @PHAI is a work variable. Since the interpolation method is not specified for Y and the number of data for PDY or PY is the one for Y minus 1 for all the intervals of X, then the interpolation method is treated as STEP.

#UNIFORM

■ Function ■

Function to sample from a uniform distribution between 2 values.

■ Format ■

```
@parameter = #UNIFORM( bound1 bound2 );
```

A value is sampled from the uniform distribution defined between *bound1* and *bound2* with the uniform probabilities. *bound1* can be larger than *bound2*.

#COSINE

■ Function ■

Function to sample from the cosine distribution.

■ Format ■

```
@parameter = #COSINE( bound1 bound2 );
```

A value is sampled from the following cosine distribution defined between *bound1* and *bound2*. *bound1* may be larger than *bound2*.

■ Probability distribution ■

$$P\{x|a < x < b\} \propto \cos\left(\pi \frac{x-a}{b-a} - \frac{\pi}{2}\right),$$

where $a = \min(\text{bound1}, \text{bound2})$, $b = \max(\text{bound1}, \text{bound2})$.

#WATT

■ Function ■

Function to sample from Watt's fission spectrum.

■ Format ■

@parameter = #WATT(a b) ;

A value is sampled from the Watt's fission spectrum distribution defined by 2 positive constants of a and b .

■ Probability distribution ■

$$P\{x|0 < x < \infty\} \propto \exp(-x/a) \sinh(\sqrt{bx})$$

#GAUSS

■ Function ■

Function to sample from the Gauss distribution.

■ Format ■

@parameter = #GAUSS(x0 sigma) ;

Sampling from the Gauss distribution (normal distribution) with the mean value $\mu = x0$ and the standard deviation $\sigma = \text{sigma}$.

■ Probability distribution ■

$$P\{x|-\infty < x < +\infty\} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

#MAXWELL

■ Function ■

Function to sample from the Maxwell distribution. The square of velocity or the kinetic energy is sampled from this function. ($T > 0$)

■ Format ■

@parameter = #MAXWELL(T) ;

■ Probability distribution ■

$$P\{x|0 < x < +\infty\} \propto \sqrt{x} \exp\left(-\frac{x}{T}\right)$$

■ Example ■

Cf-252 neutron source.

```
@E = #MAXWELL( 1.4727E6 ) ;
```

#EVAPORATION

■ Function ■

Function to sample from the evaporation spectrum distribution. ($T > 0$)

■ Format ■

```
@parameter = #EVAPORATION( T ) ;
```

■ Probability distribution ■

$$P\{x|0 < x < +\infty\} \propto x \cdot \exp\left(-\frac{x}{T}\right)$$

#POWER

■ Function ■

Function to sample from the power distribution.

■ Format ■

```
@parameter = #POWER( p bound1 bound2 ) ;
```

A value is sampled from the power distribution of the p -th power defined between $bound1$ and $bound2$ ($bound1 > 0, bound2 > 0$). $bound1$ may be larger than $bound2$.

■ Probability distribution ■

$$P\{x|a < x < b\} = \frac{(p+1)x^p}{b^{p+1} - a^{p+1}} \quad (p \neq -1)$$

$$P\{x|a < x < b\} = \frac{1}{\log(b/a)} \frac{1}{x} \quad (p = -1)$$

where $a = \min(bound1, bound2)$, $b = \max(bound1, bound2)$.

#FISSION

■ Function ■

Function to sample from the fission spectrum distribution of a specified nuclide (MVP) or a material (GMVP). The value of energy and the energy group is sampled in the case of MVP and GMVP, respectively. This function can be used to define the initial source for eigenvalue problems.

■ Format ■

Case of MVP: `@parameter = #FISSION(nuclide-ID-pattern [E_in]) ;`

Case of GMVP: `@parameter = #FISSION(material-ID) ;`

In the case of MVP, *nuclide-ID-pattern* is the nuclide ID of a fissionable nuclide specified in the CROSS SECTION data block. The wild card characters (“*”, “?”) can be used to specify the fissionable nuclide. In this case, the nuclide which matches *nuclide-ID-pattern* first in the input order of the CROSS SECTION data block is adopted.

When the arbitrary-temperature neutron cross section library is employed (See p. 130), the internal nuclide ID in MVP is the character string combined with the original nuclide ID and a temperature. This internal nuclide ID must be specified for #FISSION. For example, when the nuclide ID of “U02350J40” is used at the temperature of 625 K, the internal nuclide ID is “U02350J40625”. So, if the fission spectrum at this temperature is used strictly, #FISSION must be specified like “#FISSION(U02350J40625 0.0253)”. But there are many cases where the fission spectrum is independent of the temperature. In this case, #FISSION can be specified like “#FISSION(U02350J40* 0.0253)” or “#FISSION(U02350* 0.0253)” so that the fission spectrum which matches first in the list of the nuclide ID’s is used.

E_in is the neutron energy (eV) to cause fission and can be omitted.

■ Default ■

E_in = 0.0253 eV.

#DISC

■ Function ■

Function to sample from the uniform distribution in a disk. A point is sampled uniformly from the 2-dimensional disk (radius $r1 < r < r2$).

■ Format ■

`@(parameter1 parameter2) = #DISC(r1 r2 [angle1 angle2]) ;`

If *angle1* and *angle2* are specified, the angle θ from the direction associated with *parameter1* is limited in $angle1 < \theta < angle2$. The unit of the angle is degree and the angle is defined counter-clockwise.

■ Example ■

```
/* Uniform distribution in the disk of radius = 10cm,
/* height = 100cm
@Z = #TABLE X(0.0 100.0) PX(1) ;
@(X Y)= #DISC( 0 10.0 ) ;
```

#CIRCLE**■ Function ■**

Function to sample from the uniform distribution on a circle.

■ Format ■

```
@( parameter1 parameter2 ) = #CIRCLE( r [ angle1 angle2 ] );
```

A point is sampled uniformly from the circle of radius of r in the 2-dimensional space. When *angle1* and *angle2* are specified, the angle θ from the direction associated with *parameter1* is limited in *angle1* $< \theta < \textit{angle2}$. The unit of the angle is degree and the angle is defined counter-clockwise.

■ Example ■

```
/* 2-dimensional isotropic angular distribution
/* on a circle with radius = 1cm
@(A B)= CIRCLE(1);
```

#SPHERE**■ Function ■**

Function to sample from the uniform distribution in a sphere.

■ Format ■

```
@( parameter1 parameter2 parameter3 ) = #SPHERE( r1 r2 [ mu1 mu2 ] );
```

A point or a vector is sampled from the uniform distribution in the 3-dimensional sphere (radius $r1 < r < r2$). When *mu1* and *mu2* are specified, the cosine of the angle θ from the direction associated with *parameter1* is limited in *mu1* $< \cos \theta < \textit{mu2}$.

■ Example ■

```
/* isotropic angular distribution in the cone of  $0.8 > \mu > 0.5$ 
/* in the z-direction
@(C A B)= #SPHERE( 1 1 0.5 0.8 );
```

The #ISOTROPIC function described below is equivalent to the #SPHERE function of $r1 = r2 = 1$. The following example is the same as the above one.

```
@(C A B)= #ISOTROPIC( 0.5 0.8 );
```

#ISOTROPIC**■ Function ■**

Function to sample an isotropic 3-dimensional unit vector.

■ Format ■

```
@( parameter1 parameter2 parameter3 ) = #ISOTROPIC( [ mu1 mu2 ] );
```

A unit vector is sampled uniformly in the 3-dimensional space. (It is the same as the #SPHERE function of $r1 = r2 = 1.0$.) When $mu1$ and $mu2$ are specified, the cosine of the angle θ from the direction associated with $parameter1$ is limited in $mu1 < \cos \theta < mu2$. Even if $mu1$ and $mu2$ are omitted, the parentheses are required.

■ Example ■

```
@(C A B)= #ISOTROPIC( );
```

#FISSIONFILE

■ Function ■

Function to sample source data on a file. The fission source file must be output in other runs with the SOURCE-OUTPUT option or written in such a form. See Section 16.8 about the file form and the contents.

■ Format ■

```
@( X Y Z A B C W [ E ] ) = #FISSIONFILE (
                                [ FILE( "filename" ) ]
                                [ IUNIT( iunit ) ]
                                [ NUSE( nuse ) ]
                                ) ;
```

■ Description ■

- FILE("filename"):

The fission source file is specified on I/O unit 8 by default. This input can be used when the user wants to specify the other file.

- IUNIT(iunit):

I/O unit on which the fission source file is allocated. The default value is 8. Use of IUNIT is not recommended because other I/O units may be used in MVP/GMVP.

- NUSE(nuse):

NUSE specifies the number of particles whose information is stored in the fission source file is used. The all information stored in the file is used by default.

There are following features and attentions to be paid for this sampling function.

- In sampling, it is not necessary that the parameters such as the batch size (NHIST) are the same as those used in the calculation where the fission source file is output because source particles are randomly chosen. The file which GMVP outputs can be used in MVP and conversely the file which MVP outputs can be used in GMVP. But the following restriction is applied.

- GMVP cannot sample a value of energy in the form of “@ (X Y Z A B C W E) = #FISSION FILE (...) ;”. Instead, the form of “@ (X Y Z A B C W) = #FISSIONFILE (...) ;” that does not sample a value of energy must be used. MVP can use the former form, but the file must be output by MVP, not GMVP. In addition, there is a possibility to cause an error when the fission nuclides used in the calculation where MVP outputs the fission source file are not used in the current calculation (calculation where the file is used).
- The fission sites are stored and sampled in the absolute coordinates even if LATTICEs are used. Thus the fission source file can be used for calculations where the calculation geometry is different.
- The direction of the generated particles is sampled isotropically.
- The particle weight W must be input in the sampling information because the sampling of the fission sites only may bias the distribution in the calculation where an input data WGTF is not constant in the whole regions and whole energy groups. The particle weights are normalized so that the average value is unity.
- Since only the spatial positions are required for sampling, the fission source data output by other programs can be used by converting the data forms.

#FEYNMAN

■ Function ■

Function to sample time when multiple fission neutrons are generated from a spontaneous fission source.

■ Format ■

@T = #FEYNMAN (t_1 t_2 $\bar{\nu}$ σ) ;

The number of neutrons per spontaneous fission is determined from Terrell's distribution of Eq. (4.48) with a mean value of $\bar{\nu}$ and a standard deviation of σ . The time when the fission neutrons are generated is sampled from the uniform distribution between t_1 and t_2 and the time is the same for all the neutrons from one spontaneous fission.

■ Example ■

Sampling a time between 0.0 and 0.01 s for a spontaneous fission source of Cf-252.

@T = #FEYNMAN (0.0 0.01 3.86 1.21) ;

#TYPEn

■ Function ■

These functions (n=1,2,3,4,5,6,9) provide sources sets in the old form.³⁾ There are 7 types of sources in the old form, where the spatial distribution, energy distribution, etc. must have been input as the separated data for each type. (Data : KSOUR (type of sources), PSPAC (spatial and angular distribution etc.), PENRG (energy distribution)). This sampling function provides the spatial and angular distributions given by PSPAC.

These sampling functions are used to convert the old form of the source specification into the new form in the MVP/GMVP code. The old form of source specification is supported for backward compatibility and therefore it is not recommended to input source data in the old form.

■ Format ■

```

/***** point source, isotropic angular distribution
@(X Y Z A B C) = #TYPE1( x0 y0 z0 );
/***** point source, mono-directional angular distribution
@(X Y Z A B C) = #TYPE2( x0 y0 z0 mu eta xi );
/***** uniform distribution in a rectangular parallelepiped, isotropic angular distribution
@(X Y Z A B C) = #TYPE3( xmin xmax ymin ymax zmin zmax );
/***** uniform distribution in a sphere, isotropic angular distribution
@(X Y Z A B C) = #TYPE4( x0 y0 z0 r );
/***** uniform distribution in a disk, angular distribution perpendicular to the disk
@(X Y Z A B C) = #TYPE5( x0 y0 z0 r mu eta xi );
/***** uniform distribution in a rectangle, angular distribution perpendicular to the rectangle
@(X Y Z A B C) = #TYPE6( x0 y0 z0 ax ay az b mu eta xi );
/***** (d,t) neutron source
@(X Y Z A B C E) =
#TYPE9( x0 y0 z0 xr xdirec th1 fa1 pu1 );

```

In principle, the lists of the parameters in the left hand side must be the above ones in the specification of the sampling functions. That is, the first 3 parameters indicate the spatial coordinates, the next 3 parameters direction cosines and the 7th parameter of #TYPE9 energy. (GMVP also samples energy.) After this sampling, temporary working variables can be used to transform the sampled data further or the order of the spatial parameters can be interchanged.

The sampling of energy or the energy group cannot be specified with #TYPE1, #TYPE2, #TYPE3, #TYPE4, #TYPE5 or #TYPE6. But the sampling of energy or the energy group is automatically added from PENRG, IFISM etc. when MVP/GMVP converts the old form of the source input data (NSOUR, SOUR, KSOUR ...) into the new form.

#USERn

■ Function ■

Sampling function defined by the user. This function is used when values cannot be sampled by using the above sampling functions and the data-transform functions mentioned below. The user can use the function by writing a subroutine that samples the values in the format defined in the codes. The specification of the input parameters in the left hand side and the definition of the input data in the right hand side are user-dependent. In the input, the codes just convert the data in the parentheses of #USERn into the arbitrary number of the double-precision floating point data.

■ Format ■

```
@( parameter1 parameter2 ... ) = #USERn( ... );
```

n of #USERn indicates a number which distinguishes the user-defined sampling function ($n > 0$). The format of the subroutine written by the user is as follows.

```

      subroutine usrsrc( type, ns, irand,
&                      data, ndata, rwk, iwk,
&                      x, y, z, a, b, c, e, t, w,
&                      ierr )
c==<MVP/GMVP>=====
c purpose: generation of source particle from user supplied source.
c
c << this is a skeleton of user supplied source routine >>
c
c Users can add sampling statements to this routine.
c Examples are shown in subroutine SYSSRC.
c
c-----
c
c << arguments >>
c
c      (usage symbol: i=input, o=output, io=input & output, w=work )
c
c      i   type   : user supplied source type (character string as 'USER1')
c      i   ns     : source particles to be generated
c      io  irand  : seed of random number (INTEGER)
c      i   data   : input data to specify users source property (REAL*8)
c      i   ndata  : number of effective data elements stored in 'data' array
c      w   rwk    : working array of type REAL dimensioned as rwk(ns,4)
c      w   iwk    : working array of type INTEGER dimensioned as iwk(ns,4)
c                   Rwk and iwk occupies identical memory area , so users
c                   should take care in using them.
c
c Users must give values for all or part of the following parameters.
c
c Beware that users cannot specify kind of generated particles as
c neutron, photon etc., and cannot give energy group #'s directly in
c multigroup calculation.
c
c      o   x,y,z   : source position (unit: cm)
c      o   a,b,c   : direction cosines (a**2 + b**2 + c**2 = 1.0)
c      o   e       : energy (unit: eV)
c      o   t       : time (unit: second)
c      o   w       : particle weight (ordinary = 1.0 without biasing)
c
c Users must set and return an error code as follows.
c
c      o   ierr    : .eq. 0   if sampling succeeded.
c                   .ne. 0   if any sampling errors occurred.
c
c << Random number generation >>
c
c Use of 'RANU2' routine is recommended to keep consistency with
c other parts of the code.
c
c      call ranu2( irand, r, n, icode )
c
c      irand : seed of random number (renewed after each call)
c      r     : array of type REAL to save random number(s)
c      n     : number of random numbers generated
c      icode : error code
c
c=====
c
c      character*(*) type
c
c ... user supplied data as input ....
c
c      real*8 data(ndata)

```


#SWAP

Function to swap 2 parameters.

$$@(\textit{parameter1} \textit{parameter2}) = \# \text{SWAP}(\quad) ;$$

No parameter is specified in the parentheses of #SWAP.

■ Function ■

Function to rotate a 2-dimensional vector.

■ Format ■

@(*parameter1 parameter2*) = #ROTATE2D(*theta*) ;

This function rotates the vector of (*parameter1, parameter2*) in the (*x, y*) coordinates by *theta* degree counter-clockwise.

■ Transform ■

$$x' = x \cos \theta - y \sin \theta$$

$$y' = x \sin \theta + y \cos \theta$$

#ROTATE3D

■ Function ■

Function to rotate a 3-dimensional vector.

■ Format ■

@(*parameter1 parameter2 parameter3*) = #ROTATE3D(*theta ax ay az*) ;

Suppose that $\mathbf{X} = (parameter1, parameter2, parameter3)$ and $\mathbf{A} = (ax, ay, az)$ are the vectors in the (*x, y, z*) coordinates. \mathbf{X} is rotated about \mathbf{A} by $\theta = theta$ degree.

■ Transform ■

$$\mathbf{P} = \mathbf{X} - (\mathbf{AX})\mathbf{A}$$

$$\mathbf{P}' = \mathbf{P} \cos \theta + \mathbf{A} \times \mathbf{P} \sin \theta$$

$$\mathbf{X}' = (\mathbf{AX})\mathbf{A} + \mathbf{P}'$$

(\times is a vector product.)

Vector \mathbf{A} is normalized to the unit vector in the codes.

#RESIZE

■ Function ■

Function to normalize a 3-dimensional vector.

■ Format ■

@(*parameter1 parameter2 parameter3*) = #RESIZE(*a*) ;

The length (Euclidean norm) of the 3-dimensional vector (*x, y, z*) = (*parameter1, parameter2, parameter3*) is normalized to *a*.

■ Transform ■

$$\begin{aligned}x' &= ax/\sqrt{x^2 + y^2 + z^2} \\y' &= ay/\sqrt{x^2 + y^2 + z^2} \\z' &= az/\sqrt{x^2 + y^2 + z^2}\end{aligned}$$

#NEWAXIS**■ Function ■**

Coordinate transformation of a 3-dimensional vector.

■ Format ■

@(*parameter1 parameter2 parameter3*) = #NEWAXIS(*ax ay az bx by bz*) ;

Suppose that the 3-dimensional vector of $(x, y, z) = (parameter1, parameter2, parameter3)$ is defined in the coordinates comprised of 3 vectors $\mathbf{A} = (ax, ay, az)$, $\mathbf{B} = (bx, by, bz)$ and \mathbf{C} , where $\mathbf{C} = \mathbf{A} \times \mathbf{B} / |\mathbf{A} \times \mathbf{B}|$. The vector (x, y, z) is then transformed to the vector in the standard Cartesian coordinates with the following equation.

■ Transform ■

$$(x', y', z') = x\mathbf{A} + y\mathbf{B} + z\mathbf{C},$$

where \mathbf{C} is the unit vector perpendicular to \mathbf{A} and \mathbf{B} and is defined as

$$\mathbf{C} = \frac{\mathbf{A} \times \mathbf{B}}{|\mathbf{A} \times \mathbf{B}|}.$$

The transformation is carried out with a warning message even when \mathbf{A} or \mathbf{B} is not a unit vector or the vectors are not perpendicular to each other.

■ Example ■

```
/* Uniform angular distribution in the cone of 45-degree
/* angle about the direction cosine (1/2, -1/2, sqrt(1/2))
@(A B C) = #ISOTROPIC(<COSD(45)> 1.0);
@(A B C) = #NEWAXIS(0.5 -0.5 <SQRT(0.5)>
                  <SQRT(0.5)> <SQRT(0.5)> 0);
```

$(\sqrt{1/2}, \sqrt{1/2}, 0)$ is a unit vector perpendicular to $(1/2, -1/2, \sqrt{1/2})$.

#NORMALIZE**■ Function ■**

Function to normalize the sum of sampled parameters.

■ Format ■

@parameter = #NORMALIZE([*a*]) ;

The sum of sampled values is normalized to a . When a is set to 0 or is omitted, the sum is normalized to the number of sampled particles. This default definition is useful when the source weights are not unity and are normalized to the number of sampled particles.

■ Example ■

```

**** Correction for biased sampling
@X = #TABLE INT(LINEAR) X(<X0=0.0> <X1=1.0>)
    PDX(<P0=1.0> <P1=0.1>);

    /** Setting the weights to 1/P(@X),
    /** the distribution of @X is biased.
@W = 1/(P0 + (@X-X0)/(X1-X0)*(P1-P0)) ;

**** The sum of the weights is normalized to the number
**** of sampled particles
@W = #NORMALIZE();

```

11.7 Selection of Sampled Values

The sampled parameters (samples) can be selected by using the operation described in this section. They can be accepted as the parameters of a source particle only when they satisfy specified conditions. This process is repeated until the number of accepted samples amounts to the specified one (usually the number of particles to be sampled from a source set).

11.7.1 ACCEPT and END-ACCEPT

The selection of the samples is specified as follows.

```

ACCEPT [ ; ]
    Sampling specification line - 1;
    Sampling specification line - 2;
    ...
    WHEN Acceptance condition - 1;
    WHEN Acceptance condition - 2;
    ...
END-ACCEPT [ ; ]

```

“Sampling specification line - 1”, “Sampling specification line - 2”, ... are sampling operations that are repeated until the number of accepted samples amounts to the specified one. They are specified by using the sampling functions and data-transform functions described in the previous section. “WHEN Acceptance condition - 1”, “WHEN Acceptance condition - 2”, ... are conditions which the samples must satisfy to be accepted. The acceptance condition is the expression comprised of the sampled parameters or the conditional expression of acceptance/rejection by the specified space.

An ACCEPT ... END-ACCEPT block can be specified in another blocks. There can be not more than 4 blocks, one existing inside another.

```

ACCEPT
  Sampling specification line - 1;
  Sampling specification line - 2;
  ...
ACCEPT
  Sampling specification line - 2.1;
  Sampling specification line - 2.2;
  ...
  WHEN Acceptance condition - 2.1;
  WHEN Acceptance condition - 2.2;
  ...
END-ACCEPT
  WHEN Acceptance condition - 1;
  WHEN Acceptance condition - 2;
  ...
END-ACCEPT

```

11.7.2 Logical and Relational Functions

The logical and relational functions evaluate a condition defined by some expressions. Unless the evaluated result is 0, then the condition is satisfied. If the evaluated result is 0, then the condition is not satisfied. The available logical and relational functions are listed in Table 11.5.

Example:

```

/* accept only the energy values in the energy range
/* of the MVP library
ACCEPT
  @E = #WATT( 0.965E6 <1/2.29E6> );
  WHEN LT(@E, <%ETOPL>);
END-ACCEPT

```

Table 11.5 Logical and relational functions for conditional sampling

Logical and relational functions	Returned value (Exp1, ... are evaluated values)
AND(Exp1, Exp2 [, Exp3...])	1 when none of Exp1, Exp2 ... is 0. Otherwise, 0.
OR(Exp1, Exp2 [, Exp3...])	1 when any of Exp1, Exp2 ... is not 0. Otherwise, 0.
NOT(Exp)	1 when Exp is 0. 0 when Exp is not 0.
EQ(Exp1, Exp2)	1 when Exp1 = Exp2. Otherwise 0.
NE(Exp1, Exp2)	1 when Exp1 ≠ Exp2. Otherwise 0.
GT(Exp1, Exp2 [, Exp3...])	1 when Exp1 > Exp2 > Exp3 Otherwise, 0.
GE(Exp1, Exp2 [, Exp3...])	1 when Exp1 ≥ Exp2 ≥ Exp3 ≥ Otherwise, 0.
LT(Exp1, Exp2 [, Exp3...])	1 when Exp1 < Exp2 < Exp3 Otherwise, 0.
LE(Exp1, Exp2 [, Exp3...])	1 when Exp1 ≤ Exp2 ≤ Exp3 ≤ Otherwise, 0.
IF(Exp1, Exp2, Exp3)	Exp2 when Exp1 is not 0. Exp3 when Exp1 is 0.

Since all expressions in the parentheses are evaluated for the functions of IF, AND and OR, an error may occur when there is an expression that cannot be evaluated in some cases like division by 0. The following example may cause an error.

```
IF (EQ(@X,@Y), @Z, @Z/(@Y-@X))
```

11.7.3 Acceptance/Rejection by Specified Space

- The following statement checks whether sampled positions are included in the specified body or not.

```
WHEN [NOT-]IN-BODY( body-ID ) ;
```

- The following statement checks whether sampled positions are included in the specified ZONE or not.

```
WHEN [NOT-]IN-ZONE( input-zone-name1 input-zone-name2 ... ) ;
```

If multiple ZONES are specified, the sampled position is accepted only when it is in the inside of all the ZONES.

Example:

```
/* sample in the rectangular parallelepiped region and accept
/* only when the sampled position is in Body 1000
```

```
ACCEPT
  @X = #UNIFORM(0.0 10.0);
  @Y = #UNIFORM(0.0 20.0);
  @Z = #UNIFORM(0.0 50.0);
  WHEN IN-BODY(1000);
END-ACCEPT
```

11.8 Biased Sampling

The weight of source particles is 1.0 by default. The sampling of the weight can be biased in each source set by using the parameter W. Suppose that the following source data is specified.

```
$SOURCE
&   /** source set 1
%   RATIO=2
    RATIO(<RATIO>)
    @W = <1/RATIO> ;
...
&   /** source set 2
    RATIO(1)
...
    @W = 1.0 /** default weight
$END SOURCE
```

In this case, the ratio of the number of generated particles in source set 1, $\text{RATIO}(2)$ is twice as large as that in source set 2, $\text{RATIO}(1)$. But the source weight is $1/2$ by specifying “@W = <1/RATIO>”. Thus, the contribution of both source sets is the same in the viewpoint of probability. This is an example of the biased sampling.

In another case, the biased sampling is useful when it is difficult to sample from a probability density function. To achieve this, values are firstly sampled from an appropriate distribution (approximated distribution of the real one) and then weights are given as the function of the sampled values. This is the probabilistically correct sampling. The following specification indicates such a process.

```
&
  @X = #TABLE X(0.0 1.0) PDX(1.0); /** Sampling of X from the
                                   /** approximated distribution
  @W = < Expression to give real probabilities for X >;
  @W = NORMALIZE(0) /** Normalization of weights so that
                       /** the average value is unity
```

If the probabilities for X can be described by some expression, any distribution of $0 < X < 1$ can be expressed (unless sampling is extremely biased). In general, the weights are specified as follows.

```
@W = < Expression of real probability distribution >
     / < Expression of probability distribution for sampling >;
```

Only the relative value of the weight in sampling is meaningful because the total weight is normalized so that it is the number of histories in a batch after sampling from all the sources.

11.9 Examples for Using Sampling Functions

11.9.1 Comparison of Interpolation Methods for #TABLE Function

The interpolation methods available for the #TABLE function (LINEAR, STEP, LOG-LINEAR, LOG-STEP) are compared in this section.

The following SOURCE data block specifies LINEAR for $1.0 \text{ eV} < E < 10 \text{ eV}$ and LOG-LINEAR for $10 \text{ eV} < E < 100 \text{ eV}$.

```
$SOURCE
&
  ...
  @E = #TABLE INT(LINEAR) X(1 3 7 10) PDX(1. 4. 2. 0.);
&
  ...
  @E = #TABLE INT(LOG-LINEAR) X(10 30 70 100) PDX(1. 4. 2. 0.);
$END SOURCE
```

The following SOURCE data block specifies STEP for $1.0 \text{ eV} < E < 10 \text{ eV}$ and LOG-STEP for $10 \text{ eV} < E < 100 \text{ eV}$.

```
$SOURCE
&
  ...
```

```

@E = #TABLE INT(STEP) X(1 3 7 10) PDX(1. 4. 2.);
&
...
@E = #TABLE INT(LOG-STEP) X(10 30 70 100) PDX(1. 4. 2.);
$END SOURCE

```

Figure 11.1 shows the results for sampled values. The values are plotted with the ones per unit lethargy in the logarithmic scale for x -axis and the linear scale for y -axis. Then the results for LOG-LINEAR and LOG-STEP become linear. On the other hand, the results for LINEAR and STEP become linear with the values per unit energy in the linear scales for both x - and y -axes as shown in Fig. 11.2.

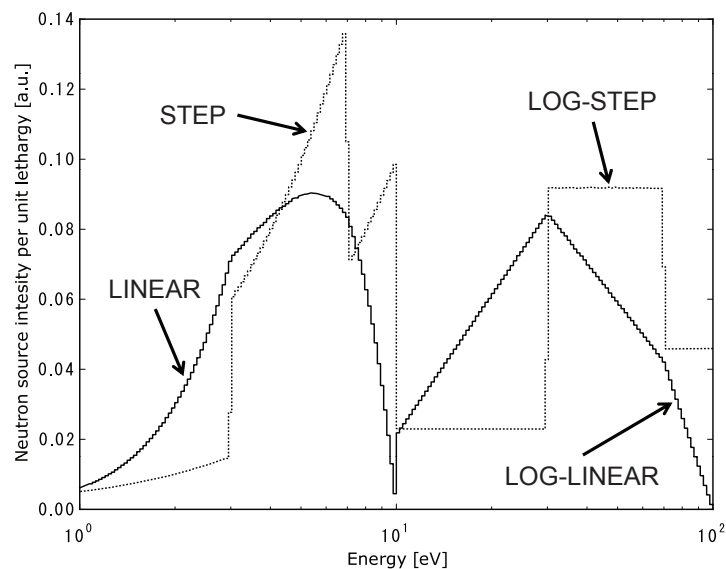


Fig. 11.1 Comparison of interpolation methods for #TABLE function (1)

11.9.2 Sampling Functions with One Variable (#TABLE, #COSINE, #GAUSS) and Selection of Sampled Values

Examples of selection of sampled values with #TABLE, #COSINE and #GAUSS are given in the following. Figure 11.3 shows the results of the sampled values.

■ Examples of input data ■

- (1) #TABLE : Sampling from a table for $1 < X < 10$.

```
@X = #TABLE INT(LINEAR) X(1 3 7 10) PDX(1. 4. 2. 0.);
```

- (2) #COSINE : Cosine distribution for $10 < X < 20$.

```
@X = #COSINE(10. 20.);
```

- (3) #COSINE : Cosine distribution for $20 < X < 30$, and sampled X such that $22 < X < 27$.

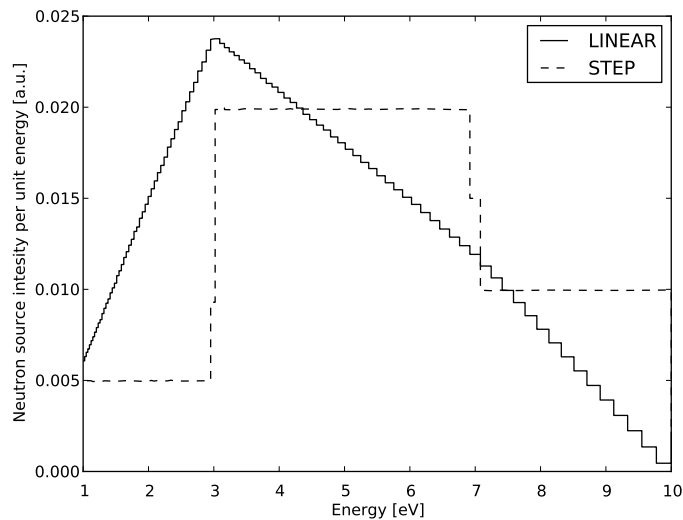


Fig. 11.2 Comparison of interpolation methods for #TABLE function (2) : LINEAR and STEP

```
ACCEPT
  @X = #COSINE(20. 30.);
  WHEN GT(@X, 22.);
  WHEN LT(@X, 27.);
END-ACCEPT
```

(4) #COSINE : Same as above but $30 < X < 40$. (An example of AND)

```
ACCEPT
  @X = #COSINE(30. 40.);
  WHEN AND(GT(@X, 32.), LT(@X, 37.));
END-ACCEPT
  @X = @X-2.;
```

(5) #COSINE : Cosine distribution for $40 < X < 50$, and sampled X such that $X < 42$ or $X > 57$.

```
ACCEPT
  @X = #COSINE(40. 50.);
  WHEN OR(LT(@X, 42.), GT(@X, 47.));
END-ACCEPT
```

(6) #GAUSS : Normal distribution with a mean of 55 and a standard deviation of 5, and sampled X such that $50 < X < 60$.

```
ACCEPT
  @X = #GAUSS(55. 5.);
  WHEN AND(GT(@X, 50.), LT(@X, 60.));
END-ACCEPT
```

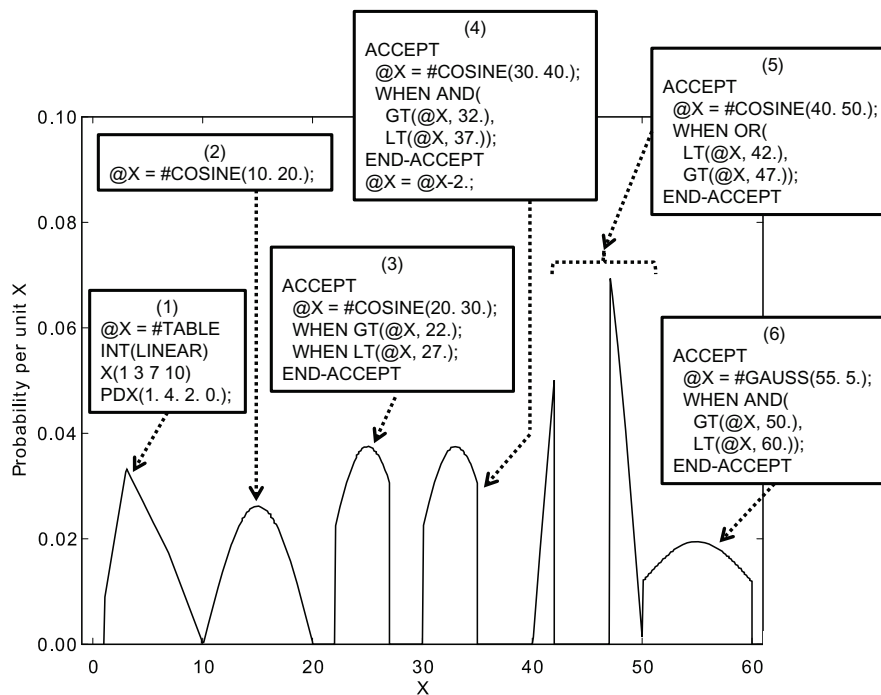


Fig. 11.3 Sampling function with a variable (selection of sampled values with #TABLE, #COSINE, #GAUSS)

11.9.3 Sampling Function with Two Variables: #TABLEXY

A example for the #TABLEXY function is given in the following. The results for sampled values are shown in Figs. 11.4, 11.5 and 11.6.

■ Example of input data ■

```
@(X Y) = #TABLEXY
INTX(LINEAR)  X(0 3 8 10)  PDX(0 1 4 2)
INTY(LINEAR)  Y(0 3 8 10)  PDY(0 1 4 2)
               Y(0 3 8 10)  PDY(1 1 1 1)
               Y(0 3 8 10)  PDY(2 4 1 0);
```

11.9.4 #TABLEXY, Work Variable and Selection of Sampled Values

The following input data is an example for the selection of the sampled values (X, Y) in the previous section; the (X, Y) values are selected such that the distance from (5, 5), which is the middle point of the range of X and Y, is less than 4.5. The result of sampled values is shown in Figs. 11.7, 11.8 and 11.9.

■ Example of input data ■

```
ACCEPT
@(X Y) = #TABLEXY
INTX(LINEAR)  X(0 3 8 10)  PDX(0 1 4 2)
INTY(LINEAR)  Y(0 3 8 10)  PDY(0 1 4 2)
```

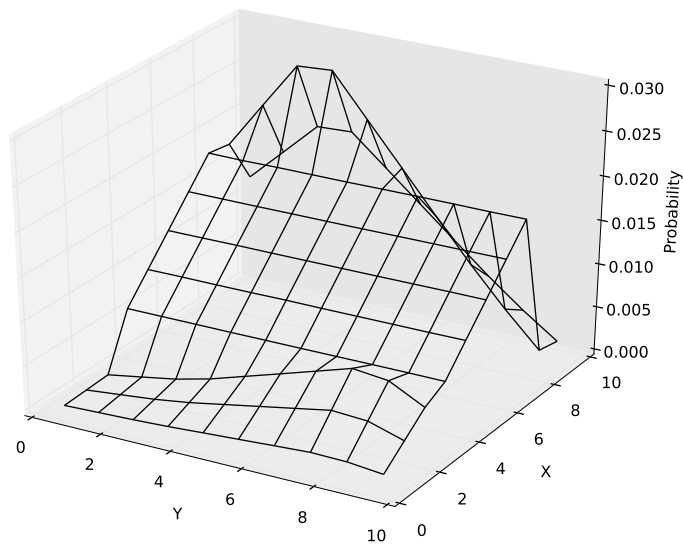


Fig. 11.4 Result for the #TABLEXY function

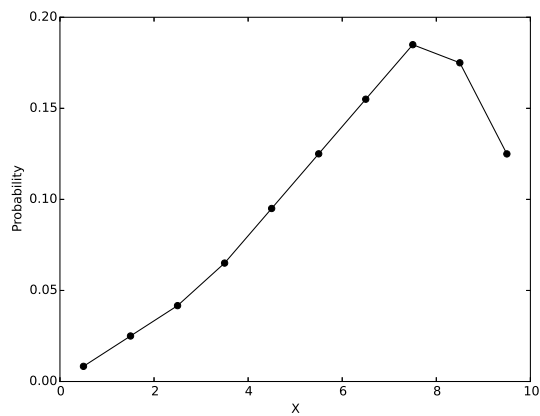


Fig. 11.5 Distribution of X

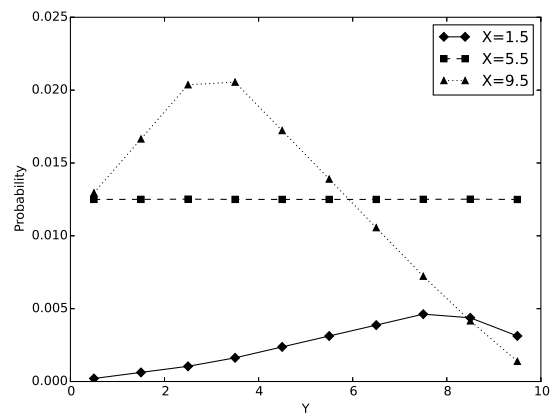


Fig. 11.6 Distribution of Y for each X

```

      Y(0 3 8 10) PDY(1 1 1 1)
      Y(0 3 8 10) PDY(2 4 1 0);
@TEMP = SQRT((@X-5)**2 + (@Y-5)**2);
  WHEN LT(@TEMP, 4.5);
END-ACCEPT

```

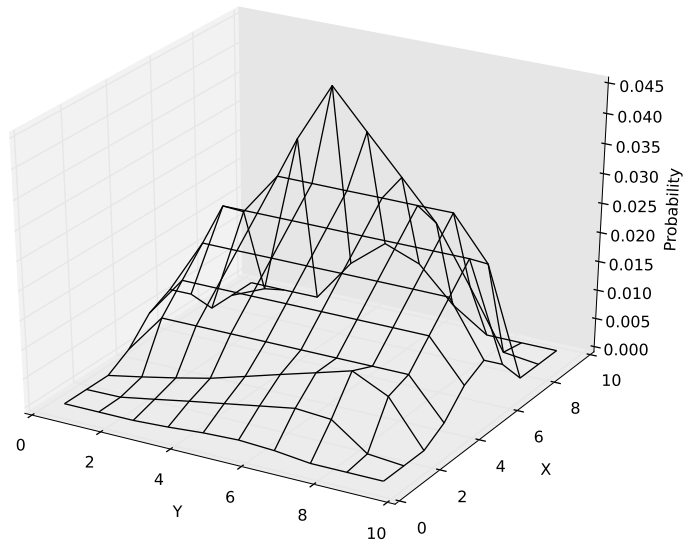


Fig. 11.7 Result of selected sampled values

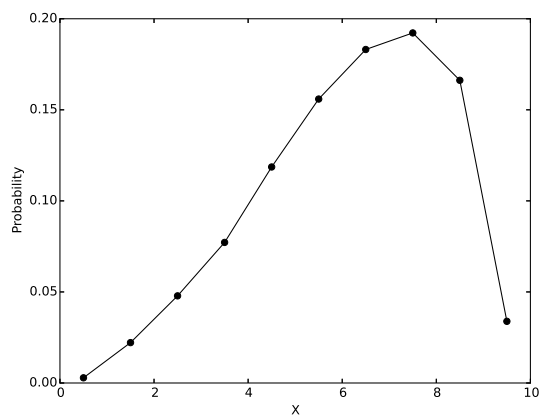


Fig. 11.8 Distribution of X : selected sampled values

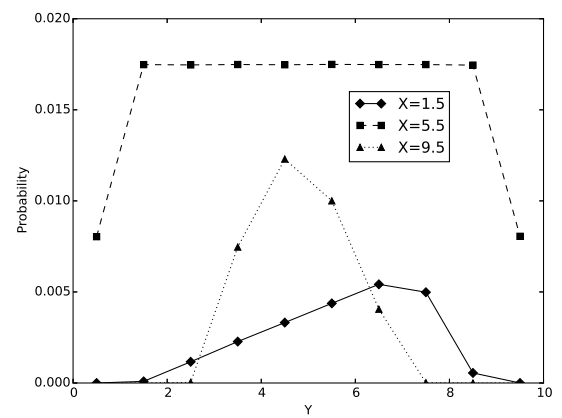


Fig. 11.9 Distribution of Y for each X : selected sampled values

12. Tally Data

12.1 TALLY Data Block

The general form of the TALLY data block is as follows.

```

$TALLY

Specification common to all tallies

&
Specification of Tally 1
&
Specification of Tally 2
....
$END TALLY

```

The form is similar to the specification of material compositions and source information.

12.2 Common Specification to All Tallies

The data described between the line of \$TALLY and the first tally is the common data to all tallies.

NGROUP.N and NGROUP.P

■ Format ■

```

NGROUP.N ( number-of-neutron-energy-bins/groups )
NGROUP.P ( number-of-photon-energy-bins/groups )

```

■ Meaning ■

The number of energy bins (energy groups) to tally for neutrons or photons. NGROUP can be input in the outside of the TALLY data block for backward compatibility. However, NGROUP must be input before the TALLY data block unless it is input in the data block.

ENGYB.N and ENGYB.P

■ Format ■

```

ENGYB.N ( neutron-energy-boundaries )
ENGYB.P ( photon-energy-boundaries )

```

■ Meaning ■

Boundaries of energy groups of neutron or photon. The data can be input in the outside of the TALLY data block for backward compatibility. However, ENGYB must be input before the TALLY data block unless it is input in the data block. ENGYB must be input after NGROUP. The unit is eV. They must be also input in the descending order from the largest value.

The energy segments defined here are the basic units of the energy bins for each tally specified in the TALLY data block.

NTIME

■ Format ■

NTIME (*number-of-time-bins*)

■ Meaning ■

The number of time bins common to time-dependent tallies. When only TIMEB is given, NTIME is calculated from the number of data given in TIMEB.

TIMEB

■ Format ■

TIMEB (*time boundaries*)

■ Meaning ■

Boundaries of time bins common to time-dependent tallies. Currently, the boundaries of time bins cannot be specified by each tally. But the time bins can be restructured by each tally. The unit is second. They must be input in the ascending order from the smallest value.

WTIME

■ Format ■

WTIME (*w_1 w_2 ... w_NTIME*)

■ Meaning ■

Weights set to the common time bins in time-dependent calculations. These data are used as follows.

- Time-dependent variance reduction parameters

The weight parameters of WSRV and WKIL are multiplied by WTIME for each particle's time. These parameters are used with the variance reduction option WEIGHT-WINDOW or RUSSIAN-ROULETTE.

- Time-dependent weights for generation of secondary particles

The weights for generation of fission neutrons WGTF or secondary photons WGTP are multiplied by WTIME for each particle's time. These weights are used in the fixed-source problems where secondary particles are generated (neutron/photon coupled calculations, neutron calculations with fission sources).

TALLY-SURFACE

■ Format ■

TALLY-SURFACE(*surface-ID* *body1* [*body2* *body3* ...])

■ Meaning ■

Specification of a detector surface for a surface crossing estimator. Multiple detector surfaces can be specified in the common data of the TALLY data block.

- *surface-ID* : ID number of a detector surface. It must be a positive integer and cannot be multiply defined.
- *body1* : Body ID which includes the detector surface. This body must be defined in the GEOMETRY data block. It is not necessary that this body must be used to define a ZONE. The normal direction to the detector surface is specified by the sign of *body1*; the direction from inside to outside of the BODY is taken if the sign is positive and the opposite direction is taken if negative.
- *body2* or more : The detector surface is defined as a part of *body1* included in the combinatorial region of *body2* or more. (clipping of tally-surface)

There are following restrictions and attentions to be paid for the specification of the detector surface.

- TALLY-SURFACE can be used with the LATTICE option but the body which defines the location of TALLY-SURFACE must be specified in the absolute coordinates.
- ELT (torus) cannot be used as *body1* for TALLY-SURFACE.
- Parameters cannot be tallied correctly when a part of or the entire detector surface is on the surface of the reflective boundary.
- A part of or the entire detector surface can be on the boundary of ZONES. But if a spatial region (REGION) is specified as the dimension of tallies (DIMENSION) and the REGION names for both sides of the detector surface are different from each other, it is not identified which side physical parameters are tallied for.

ANGLE

■ Format ■

ANGLE(*angle_1* *angle_2* ...)

■ Meaning ■

Angular bins for angle-dependent tallies. Currently, these are used for the surface crossing estimator only. The angle between the flight direction of particles passing through the detector surface and the normal of the detector surface must be specified. The normal vector is directed depending on the sign of the *body1* data for TALLY-SURFACE; the direction is taken from inside to outside of the BODY if the sign is positive and the direction is opposite if negative.

The unit of the angle is degree. The values from 0 to 180 must be input in the ascending order. The user can also specify the bins by cosine values (COSINE). Either ANGLE or COSINE must be used.

COSINE

■ Format ■

COSINE (*cosine_1 cosine_2 ...*)

■ Meaning ■

Angular bins by cosine for angle-dependent tallies. Currently, these are used for the surface crossing estimator only. The cosine of the angle between the flight direction of particles passing through the detector surface and the normal of the detector surface must be specified.

The cosine values from -1 to 1 must be input in the descending order. The user can also specify the bins by degree (ANGLE). Either ANGLE or COSINE must be used.

NPDET

■ Format ■

NPDET (*number-of-detector-points*)

■ Meaning ■

The number of point detectors. NPDET is used to define the number of the input data for XPDET and SPDET described below.

XPDET

■ Format ■

XPDET ($X_1 Y_1 Z_1 R_{\text{reject},1} R_{\text{sing},1}$
 $X_2 Y_2 Z_2 R_{\text{reject},2} R_{\text{sing},2}$
 \dots
 $X_{\text{NPDET}} Y_{\text{NPDET}} Z_{\text{NPDET}} R_{\text{reject},\text{NPDET}} R_{\text{sing},\text{NPDET}}$
)

■ Meaning ■

Parameters to define the coordinates of point detectors and sampling condition by them.

- X_i, Y_i, Z_i

The coordinates of the position for each point detector. They must be specified in the absolute coordinates. They cannot be specified in the CELL coordinates.

- R_{reject}

The point detector estimator evaluates the statistical parameters contributed from collision points of particles in materials. This is processed by generating virtual particles from the collision points to the detector point. Therefore, the large computation time may be required when the number of collisions is large.

MVP/GMVP reduces the number of the virtual particles by selecting stochastically the collision points located at a distance of more than R_{reject} from the detector point, because the contribution from the distant collision points is small in general. In this case, the weight of the virtual particle selected is multiplied by the inverse of the selection probability for statistically correct evaluation. Letting the distance from the collision point to the detector point be D , the selection probability is

$$p(D) = \frac{R_{\text{reject}}^2}{D^2}.$$

- R_{sing}

It is well known that the variance becomes large when collisions occur in the vicinity of the detector point because the variance for the point detector is infinity in theory. In general, special techniques are used in the vicinity to alleviate this drawback. In order to prevent the divergence of the variance, MVP/GMVP uses the method where the contribution from collisions within the distance of R_{sing} from the detector point is replaced with the averaged one. This method assumes the constant collision density within the sphere of the radius of R_{sing} . Strictly speaking, this method may bias the statistical results, thus the user should pay attention to this point.

SPDET

■ Format ■

SPDET (*flag*₁ *flag*₂ ... *flag*_{NPDET})

■ Meaning ■

Integer to specify the calculation method of the uncollided contribution from sources for the point detector. In the description of XPDET, the contributions due to virtual particles from the collision points are mentioned. In addition, we must take account of the contribution from the uncollided particles from the sources. In this case, the angular distribution of the generated particles must be known to determine the weight of the virtual particles. The angular distribution for the generation of secondary particles from collisions is evaluated as accurately as possible. But the anisotropic angular distribution from the source cannot be treated correctly in the current version of MVP/GMVP. The value of *flag*_{*i*} indicates the following.

- *flag*_{*i*} = 0

The angular distribution from the source is assumed to be isotropic.

- *flag*_{*i*} > 0

The angular distribution from the source is assumed to be anisotropic. But this angular distribution cannot be treated correctly in the current version of MVP/GMVP.

- *flag*_{*i*} < 0

The contribution from the source is not evaluated.

Currently, MVP/GMVP can treat the contribution from the source to the point detector accurately in the case of isotropic source.

NRESP

■ Format ■

NRESP (*number-of-energy-group-dependent-responses*)

■ Meaning ■

The number of group-wise response functions to evaluate reaction rates. Unless this data is input in the TALLY block, it must be input before the TALLY block.

RESP.N and RESP.P

■ Format ■

```
RESP .N( r11 r21 ... rNGROUP.N,1
         r12 r22 ... rNGROUP.N,2
         ...
         r1, NRESP.N r2, NRESP.N ... rNGROUP.N, NRESP.N )
RESP .P( r11 r21 ... rNGROUP.P,1
         r12 r22 ... rNGROUP.P,2
         ...
         r1, NRESP.P r2, NRESP.P ... rNGROUP.P, NRESP.P )
```

■ Meaning ■

Group-wise response functions to evaluate reaction rates. “.N” and “.P” indicate the data for neutrons and photons, respectively. If “.N” or “.P” is not given, the data is assumed to be given for the energy structure coupled with neutrons and photons (the number of energy groups NGROUP.N + NGROUP.P).

NSTAL

■ Format ■

NSTAL (*nstal*)

■ Meaning ■

The number of response functions given in the continuous-energy form. (effective in MVP only.) The response functions are specified in STAL or CRESP and read in from the continuous-energy response library (dosimetry library).

STAL or CRESP

■ Format ■

STAL (*nuclide-id*(*MTreaction-ID*) [*nuclide-id*(*MTreaction-ID*) ...])

or

CRESP (*nuclide-id*(MTreaction-ID) [*nuclide-id*(MTreaction-ID) ...])

■ Meaning ■

Specification of response functions given in the continuous-energy form (effective only for MVP). NSTAL sets of *nuclide-id* and MTreaction-ID must be input for nuclide identifiers and reaction MT numbers. See Section G.5 for available response function data, nuclide identifiers and reaction MT numbers.

Example:

```
NSTAL( 2 )
STAL( 3006.03Y( MT105 ) 3007.03Y( MT205 ) )
```

PNPRODUCE

■ Format ■

PNPRODUCE

■ Meaning ■

This option enables MVP to tally selectively the contribution from secondary particles generated in photonuclear reactions and their progeny if the tallies have dimensions of a production reaction (PRODUCE-REGION, PRODUCE-NUCLIDE, or PRODUCE-REACTION). The contribution not only from the secondary particles but also from their progeny is tallied. Unless the dimensions of a production reaction is specified, this option is ignored. Unless this option is specified, MVP tallies the contribution from a collision event just before tallying. It should be noted that no contribution from the source is tallied.

■ Related options ■

This option is effective only if so is the option PHOTO-NUCLEAR (p. 107).

NWORK

■ Format ■

NWORK (*working-area-size-for-TALLY-data-block-processing*)

■ Meaning ■

Size of work memory to process the TALLY data block (words). The default value is the maximum value among NGROUP(=NGROUP.N+NGROUP.P), NTIME, NREG (number of total regions), NTREG (number of total tally regions) that have been already input before this data block, or 4,096.

12.3 Input Data for Each Tally

12.3.1 Specification of Tally ID

■ Format ■

ID(*tally-id-number*)

■ Meaning ■

ID number for a tally. It can be omitted.

12.3.2 Specification of Tally Label

■ Format ■

LABEL("*label string*")

■ Meaning ■

Label to be print with tallies. It is not always necessary. When the character string of the label includes blank characters, the string must be enclosed in double quotation “...” or single quotation ‘...’. When the character string of the label includes a single quotation mark, the string must be enclosed in double quotation “...”.

12.3.3 Specification of Estimator

■ Format ■

EVENT(*event*)

■ Meaning ■

The character string that indicates a tally event or a type of estimators. One of the following character strings can be specified.

- TRACK : Track-length estimator in the specified REGION.
- COLLISION : Collision estimator in the specified REGION.
- SURFACE : CURRENT (*surface-ID*) or SURFACE (*surface-ID*) : Particle current calculated by the surface crossing estimator.
- SURFACE : FLUX (*surface-ID*) : Particle flux calculated by the surface crossing estimator.
- POINT : Point detector.
- LEAKAGE : Scoring weights of particles when they leak from the system. Namely the current is calculated and the specification for tallied quantities such as MICRO, MACRO, IRESP and PRESP is meaningless.

- ANALOG : Scoring weights of particles when they collide with nuclei. The ANALOG-FISSION option must be specified. Only the specification of ANALOG (p. 246) is effective for tallied quantities.

One of the surface crossing estimators given by TALLY-SURFACE must be specified for the *surface-ID* that specifies the surface crossing estimator.

■ Default ■

TRACK. The track-length estimator is used.

12.3.4 Specification of Tally Particle

PHOTON or NEUTRON

■ Format ■

NEUTRON

or

PHOTON

■ Meaning ■

Type of particles to be tallied. This parameter must be always input in the neutron-photon coupled problems (Both the NEUTRON and PHOTON options can be specified.). The form of PARTICLE (*particle-type*) can be also used as described below.

PARTICLE

■ Format ■

PARTICLE(NEUTRON)

or

PARTICLE(PHOTON)

■ Meaning ■

Type of particles to be tallied. When MVP/GMVP is available for particles except for neutron and photon in the future, this form will be used to tally for such particles.

12.3.5 Specification of Tally Dimensions

■ Format ■

DIMENSION(*dim1 dim2 dim3 ...*)

■ Meaning ■

Dimensions to be tallied for. “*dim**” is the character string which represents the dimension.

- REGION : Tally region.
- POINT : Detector point for the point detector.
- ENERGY : Energy.
- TIME : Time.
- ANGLE : Angle. The angle or the cosine of the angle between the normal of the detector surface and the flight direction of particles for the surface crossing estimator.
- GENERATION : Generation of particles in the fixed-source problem.
- SOURCE : Source set ID from which particles are generated.
- SOURCE-REGION : REGION of the source from which particles are generated.
- MARKER-REGION : “marker REGION”. Parameters are tallied for the particles that pass through the specified REGION.
- PRODUCE-REGION : Spatial region where a secondary particle is produced from collisions except for elastic scattering.
- PRODUCE-NUCLIDE : Target nuclide with which a primary particle is collided and a secondary particle is produced from collisions except for elastic scattering.
- PRODUCE-REACTION : Collision reaction (MT number) from which a secondary particle is produced except for elastic scattering.

Parameters are tallied as a multi-dimensional matrix whose components are divided by the boundaries in the above dimensions. The data of this matrix are stored in the reverse order of FORTRAN arrays in the program. That is, the data are stored as $M(\dots, dim3, dim2, dim1)$. Calculation results are also output in this order.

The number of specified dimensions is not more than 5 for each tally. The bins for each dimension are specified by the following input parameters. See the description of the input data for the bins to know the meaning of them.

- REGION : REGION (p. 233)
- POINT : POINT (p. 234)
- ENERGY : IENERGY (p. 234), IENERGY2 (p. 235)
- TIME : ITIME (p. 235), ITIME2 (p. 236)
- ANGLE : IANGLE (p. 236), IANGLE2 (p. 237)
- GENERATION : GENERATION (p. 238), IGENE2 (p. 238)
- SOURCE : SOURCE (p. 239)
- SOURCE-REGION : SOURCE-REGION (p. 239)

- MARKER-REGION : MARKER-REGION (p. 239)
- PRODUCE-REGION : PRODUCE-REGION (p. 240)
- PRODUCE-NUCLIDE : PRODUCE-NUCLIDE (p. 240)
- PRODUCE-REACTION : PRODUCE-REACTION (p. 242)

■ Example ■

If DIMENSION is specified as

```
DIMENSION( REGION TIME ENERGY ),
```

the order of data is

```
(Energy1, Time1, Region1), (Energy2, Time1, Region1) ...
(Energy1, Time2, Region1), (Energy2, Time2, Region1) ...
...
(Energy1, Time1, Region2), (Energy2, Time1, Region2) ...
(Energy1, Time2, Region2), (Energy2, Time2, Region2) ...
...
(Energy1, Time1, Region3), (Energy2, Time1, Region3) ...
(Energy1, Time2, Region3), (Energy2, Time2, Region3) ...
...
```

12.3.6 Specification of Tally Bins

REGION

■ Format ■

```
REGION( region-name-pattern_1 region-name-pattern_2 ... )
```

■ Meaning ■

When a tally has the dimension of spatial regions, a list of the region names must be specified. The name of REGION or TALLY REGION (p. 191) can be specified for the spatial regions.

The names of the specified REGIONS are delimited by more than one blank or a return code in the list. The following forms of REGION or TALLY REGION can be used to specify the names. Each specified REGION is the bin to be tallied for.

- REGION name
Name of a single REGION.
- REGION name with wild card characters
Multiple REGION names are specified with wild card characters. It can include the wild card character “*” (arbitrary character string or no characters), “?” (arbitrary 1 character).
- TALLY REGION name
Multiple REGIONS defined in the data sub-block from the #TALLY REGION line. It is a character string which begins with “@”.

- Combined form of the above forms with “+”. It cannot include blank characters.

■ Example ■

Suppose that the names of REG1, REG2, REG3, REG4, A1, A2 and A01 are defined in the input data and furthermore @TREG1(REG1 REG2 REG3) and @TREG2(REG2 REG4) are defined. In this case, the specification like

```
REGION( REG1 REG* A? @TREG1 A*+REG1+@TREG2 @TREG1+@TREG2 )
```

indicates that parameters are tallied in the following regions.

- REG1 : Tally in REGION “REG1” only.
- REG* : Sum of tallies in REGION “REG1”, “REG2”, “REG3” and “REG4”.
- A? : Sum of tallies in REGION “A1” and “AA”. The tally in A01 is not included.
- @TREG1 : Tally in TALLY REGION “@TREG1”. That is, the sum of tallies in REG1, REG2 and REG3.
- A*+REG1+@TREG2 : Sum of tallies in A1, A2, A01, REG1, REG2 and REG4.
- @TREG1+@TREG2 : Sum of tallies in REG1, REG2, REG3 and REG4. REG2 is included in both @TREG1 and @TREG2 but the parameters are tallied in REG2 only once. Note that “+” means the operation of union for the regions defined in each pattern.

■ Note ■

The region pattern connected by “+” cannot be defined in more than one line.

POINT

■ Format ■

```
POINT( detector-point-ID_1 detector-point-ID_2 ... )
```

■ Meaning ■

The list of point detectors by which physical parameters are tallied. The detector number defined in the common data must be specified. More than one detector number can be specified.

IENERGY

■ Format ■

```
IENERGY( ie_1 ie_2 ... ie_NGROUP[.N/.P] )
```

■ Meaning ■

Bunched energy bins to be tallied for. This data is specified when tallies have a dimension of ENERGY (Form 1). The i-th energy bin defined by ENGYB.N for neutrons or ENGYB.P for photons (See

Section 12.2.) corresponds to the ie_i -th bunched energy bin. ie_i must be a sequential integer that begins with 0 or 1. If $ie_i=0$, then physical parameters are not tallied for this general energy bin.

■ Example ■

When the number of general energy bins for neutrons NGROUP.N is 10,

```
IENERGY( 0 1 1 1 2 2 2 3 3 4 )
```

shows that there are 4 bunched energy bins and the general energy bins are bunched from 2 to 4, from 5 to 7, from 8 to 9 and 10, respectively. The physical parameters are not tallied for the first general energy bin.

■ Note ■

Either IENERGY(...) or IENERGY2(...) can be specified for each tally.

IENERGY2

■ Format ■

```
IENERGY2( ies_1 iee_1 ies_2 iee_2 ... )
```

■ Meaning ■

Bunched energy bins to be tallied for. This data is specified when tallies have a dimension of ENERGY (Form 2). The general energy bins from ies_i -th to iee_i -th correspond to the i -th bunched energy bin. The number of bunched energy bins is arbitrary.

■ Example ■

When the number of general energy bins for neutrons NGROUP.N is 10,

```
IENERGY2( 2 4
           5 7
           8 9
          10 10
           1 10 )
```

shows that there are 5 bunched energy bins and the general energy bins are bunched from 2 to 4, from 5 to 7, from 8 to 9, 10 and from 1 to 10 (whole energy range), respectively. One bunched energy bin can be overlapped with the other like the 5th bin.

■ Note ■

Either IENERGY(...) or IENERGY2(...) can be specified for each tally.

ITIME

■ Format ■

```
ITIME( it_1 it_2 ... it_NTIME )
```

■ Meaning ■

Bunched time bins to be tallied for. This data is specified when tallies have a dimension of TIME (Form 1). The i -th time bin defined by TIMEB corresponds to the it_i -th bunched time bin. it_i must be a sequential integer that begins with 0 or 1. If $it_i = 0$, then physical parameters are not tallied for this general time bin.

■ Example ■

When the number of general time bins NTIME is 10,

```
ITIME( 0 1 1 1 2 2 2 3 3 4 )
```

shows that there are 4 bunched time bins and the general time bins are bunched from 2 to 4, from 5 to 7, from 8 to 9 and 10, respectively. The physical parameters are not tallied for the first general time bin.

■ Note ■

Either ITIME(...) or ITIME2(...) can be specified for each tally.

ITIME2**■ Format ■**

```
ITIME2( its_1 ite_1 its_2 ite_2 ... )
```

■ Meaning ■

Bunched time bins to be tallied for. This data is specified when tallies have a dimension of TIME (Form 2). The general time bins from its_i -th to ite_i -th correspond to the i -th bunched time bin. The number of bunched time bins is arbitrary.

■ Example ■

When the number of general time bins NTIME is 10,

```
ITIME2( 2 4
        5 7
        8 9
        10 10
        1 10 )
```

shows that there are 5 bunched time bins and the general time bins are bunched from 2 to 4, from 5 to 7, from 8 to 9, 10 and from 1 to 10 (whole time range), respectively. One bunched time bin can be overlapped with the other like the 5th bin.

■ Note ■

Either ITIME(...) or ITIME2(...) can be specified for each tally.

IANGLE**■ Format ■**

IANGLE(*ia_1 ia_2 ... ia_NANGLE*)

■ Meaning ■

Bunched angular bins to be tallied for (Form 1). IANGLE is effective if ANGLE is specified in DIMENSION (p. 231). The *i*-th general angular bin defined by ANGEL or COSINE corresponds to the *ia_i*-th bunched angular bin. *ia_i* must be a sequential integer that begins with 0 or 1. If *ia_i*=0, then physical parameters are not tallied for this general angular bin.

■ Example ■

When the number of general angular bins NANGLE is 10,

IANGLE(0 1 1 1 2 2 2 3 3 4)

shows that there are 4 bunched angular bins and the general angular bins are bunched from 2 to 4, from 5 to 7, from 8 to 9 and 10, respectively. The physical parameters are not tallied for the first general angular bin.

■ Note ■

Either IANGLE(...) or IANGLE2(...) can be specified for each tally.

IANGLE2

■ Format ■

IANGLE2(*ias_1 iae_1 ias_2 iae_2 ...*)

■ Meaning ■

Bunched angular bins to be tallied for (Form 2). IANGLE2 is effective if ANGLE is specified in DIMENSION (p. 231). The general angular bins from *ias_i*-th to *iae_i*-th correspond to the *i*-th bunched angular bin. The number of bunched angular bins is arbitrary.

■ Example ■

When the number of general angular bins NANGLE is 10,

```
IANGLE2( 2  4
          5  7
          8  9
          10 10
          1  10 )
```

shows that there are 5 bunched angular bins and the general angular bins are bunched from 2 to 4, from 5 to 7, from 8 to 9, 10 and from 1 to 10 (whole angular range), respectively. One bunched angular bin can be overlapped with the other like the 5th bin.

■ Note ■

Either IANGLE(...) or IANGLE2(...) can be specified for each tally.

GENERATION

■ Format ■

```
GENERATION( igen_1 igen_2 ... )
```

■ Meaning ■

Bunched generation bins to be tallied for. This data is specified when physical parameters are tallied for particles' generation in the fixed-source problems where the secondary particles are generated (neutron/photon coupled problems, neutron problem with fission) (Form 1).

The i -th generation corresponds to the $igen_i$ -th bunched generation bin. $igen_i$ must be a sequential integer that begins with 0 or 1. If $igen_i=0$, then physical parameters are not tallied for this generation.

The number of the maximum generation is arbitrary. It can be larger than the limitation value of the particles' generation MXPGEN (p. 121). But scores for the (MXPGEN+1)-th generation or later are 0 in this case.

■ Example ■

```
GENEARTION( 0 1 2 2 2 3 3 )
```

The bin for the second generation (secondary particles) is the first bunched bin. The bin for the third ~ fifth generation is the second bunched bin. The bin for the sixth ~ seventh generation is the third bunched bin. The physical parameters for the first generation (primary particles) are not tallied.

■ Note ■

Either GENERATION(...) or IGENE2(...) can be specified for each tally.

IGENE2

■ Format ■

```
IGENE2( igs_1 ige_1 igs_2 ige_2 ... )
```

■ Meaning ■

Bunched generation bins to be tallied for. This data is specified when physical parameters are tallied for particles' generation in the fixed-source problems where the secondary particles are generated (neutron/photon coupled problems, neutron problem with fission) (Form 2).

The generation from igs_i -th to ige_i -th corresponds to the i -th bunched generation bin. The number of bunched generation bins is arbitrary.

■ Example ■

```
IGENE2( 2 4
        5 7
        3 -1
        1 -1 )
```

In this example, there are 4 bunched generation bins. The generations of 2 ~ 4 and 5 ~ 7 correspond to the first and second bunched generation bins, respectively. The upper limit of the bunched generation bin is a negative integer for the third and fourth bins. This is a special specification which means that the bunched generation bin includes all generation bins of which generation is larger than the lower limit. The third bunched bin includes the third generation bin and later, the fourth bunched bin includes all generation bins.

■ Note ■

Either GENERATION(...) or IGENE2(...) can be specified for each tally.

SOURCE

■ Format ■

SOURCE (*source-set-ID_1 source-set-ID_2*)

■ Meaning ■

Tallies by the source set (Set of source information (p. 194)) from which particles are generated in the SOURCE data block. The ID of source sets or the ID number which is automatically set in the input order in MVP/GMVP must be specified.

■ Note ■

If the multiple source sets have the same ID, physical parameters are tallied for the sum of these source sets.

SOURCE-REGION

■ Format ■

SOURCE-REGION (*region-name-pattern_1 region-name-pattern_2 ...*)

■ Meaning ■

Tallies by the spatial region where particles are generated from fixed or fission sources in eigenvalue problems. The list of patterns for region names must be input and the rule is the same as the one described in REGION (p. 233).

MARKER-REGION

■ Format ■

MARKER-REGION (*region-name-pattern_1 region-name-pattern_2 ...*)

■ Meaning ■

Tallies by the spatial region which particles pass through ("MARKER-REGION").

■ Description ■

The list of patterns for region names must be input and the rule is the same as the one described in REGION (p. 233). Note that physical parameters are not tallied for each spatial region defined by the patterns but are tallied for the union of the spatial regions. That is, only one MARKER-REGION can be specified for each tally.

The tallies are output by the following three ways.

- (1) Tallies after particles pass through the MARKER-REGION.
- (2) Tallies before particles pass through the MARKER-REGION.
- (3) The sum of the above tallies.

When particles travel in a MARKER-REGION after they are generated in the MARKER-REGION or enter the MARKER-REGION, the codes do not regard the particles as having passed through the MARKER-REGION. The particles in the MARKER-REGION are not marked until the particles enter the neighboring regions of the MARKER-REGION.

PRODUCE-REGION

■ Format ■

```
PRODUCE-REGION( region-name-pattern_1 region-name-pattern_2 ... )
```

■ Meaning ■

This input data specifies tallying the contribution depending on which spatial region a particle is produced in a collision except for elastic scattering. This data can be specified only for MVP. The list of REGION name patterns can be input. The REGION name pattern is described in the tally input data of REGION (p. 233).

PRODUCE-NUCLIDE

■ Format ■

```
PRODUCE-NUCLIDE( nuclide-name-pattern_1 nuclide-name-pattern_2 ... )
```

■ Meaning ■

This input data specifies tallying the contribution depending on which target nuclide a primary particle is collided with and a secondary particle is produced from the collision except for elastic scattering. This data can be specified only for MVP. The list of specified nuclides can include the patterns of nuclide ID's delimited with one or more than one space or line breaks. The following format can be specified for the patterns.

- Single nuclide name specified with 5 characters *zzaaa*. *zz* is the element name (append “0” for 1-character element). *aaa* is the atomic mass (use “000” for a natural element). If a wild card character “*” is used for *aaa*, it means that all isotopes for the element are specified. If *aaa* is omitted and only the element name is specified, it means the same as the natural element.
- Single nuclide names connected with “+” or “-” (no spaces are included). “+” and “-” mean that the following nuclide is added and is removed, respectively. There is no restriction for the number of connecting nuclides.

- ALL. All nuclides given in the cross section input data are specified. The nuclide name pattern must start with ALL. ALL-FE056 is OK, but FE056-ALL is NG.

■ Example ■

The following example shows a general usage.

```
$TALLY
...
PNPRODUCE
& ID(1) NEUTRON
...
DIMENSION(PRODUCE-NUCLIDE)
PRODUCE-NUCLIDE(FE056 AL027)
& ID(2) NEUTRON
...
DIMENSION(PRODUCE-NUCLIDE)
PRODUCE-NUCLIDE(ALL)
$END TALLY
```

The following example shows a usage of a wild card character “*”, “+” and “-”.

```
PRODUCE-NUCLIDE( FE056 FE* FE000+W0000 ALL-FE* )
```

Suppose that materials of iron and tungsten in an input data. The specification above means that MVP calculates following 4 types of tallies.

- FE056: tally only for Fe-56.
- FE*: sum of tallies for Fe-54, 56, 57, 58.
- FE000+W0000: sum of tallies for Fe-54, 56, 57, 58, W-182, 183, 184, 186.
- ALL-FE*: sum of tallies for W-182, 183, 184, 186. ALL means tallies for all isotopes for Fe and W, but -FE* removes tallies for Fe.

■ Note ■

A nuclide ID pattern connected with “+” or “-” cannot be specified in multiple lines. “-” cannot always remove isotopes from a natural element (“000”). For example, if the iron element of FE000 is specified, the specification of -FE054-FE056-FE057-FE058 is not effective since the natural element cannot be decomposed into each isotope with regard to neutron reactions. If the iron element needs to be removed, the specification must be -FE*. For photonuclear reactions, isotopes can be removed with “-” from the element specification with “000” since the element is decomposed into each isotope. For photoatomic reactions, isotopes cannot be removed with “-” from the element specification with “000” if neutron or photonuclear reactions are specified for the isotopes; isotopes can be removed only if all the isotopes are removed for neutron or photonuclear reactions. Unless neutron or photonuclear reactions are specified for the isotopes, isotopes can be removed with “-” (namely, photon-only calculations).

PRODUCE-REACTION

■ Format ■

```
PRODUCE-REACTION( mt-n_1 mt-n_2 ... mt-n_NMT
                  mt-p_1 mt-p_2 ... mt-p_NMTP
                  mt-pn_1 mt-pn_2 ... mt-pn_NMTPN )
```

■ Meaning ■

This input data specifies tallying the contribution depending on reactions from which a secondary particle is produced except for elastic scattering. The numbers of reaction types to be specified for a problem including photonuclear reactions are 120 for neutron reactions (NMT=120), 60 for photon reactions (NMTP=60), and 135 for photonuclear reactions (NMTPN=135). See Appendix H for each reaction. The user must specify non-negative integers for each reaction type. Each integer specifies a reaction bin for the tallies. If “0” is specified for reaction types, no contribution from the reactions is tallied. If the number of input data for the reaction types is less than 315, the rest of data are treated as 0. This data can be specified only for MVP. If the DELAYED-NEUTRON option is effective and the generation of delayed neutrons is possible, fission neutrons are separated into prompt neutrons (MT=18) and delayed neutrons (MT=98). If the user would like to calculate the contribution from the delayed neutrons, a reaction bin must include the 98th reaction type (MT=98).

■ Example ■

The following input data shows an example of calculating the contribution from delayed neutrons generated in photofissions.

```
$TALLY
PNPRODUCE
...
& ID(1)
  LABEL('CAPTURE BY DELAYED NEUTRON FROM PHOTO-FISSION IN U-238')
  EVENT(TRACK)
  NEUTRON
  DIMENSION(REGION PRODUCE-REACTION PRODUCE-REGION
             PRODUCE-NUCLIDE)
  REGION(@DETR)
  PRODUCE-REGION(BALL)
  PRODUCE-NUCLIDE(U0238)
  PRODUCE-REACTION(120(0) 60(0) 97(0) 1 R(0))
  MACRO(CAPTURE)
$END TALLY
```

In this example, all neutron and photon reaction types are set to 0, and only the 98th photonuclear reaction type (delayed neutron from photofissions) is set to 1. MVP calculates the macroscopic capture reaction rate in the tally region of @DETR only from the contribution of delayed neutrons generated in photofissions of U-238 in the region of BALL.

The following example shows the specification of multiple reaction bins.

```
PRODUCE-REACTION( 4(0) 1 10(0) 1 5(0) 1 5(0) 1 22(0) 41(1) 29(0)
                  60(0)
                  4(0) 2 10(0) 3 5(0) 3 5(0) 3 21(0) 42(4) 6(0) 5
                  37(0) )
```


Reaction bin 1 includes neutron reactions (MT=5, 16, 22, 28), reaction bin 2 a photonuclear reaction (MT=5), reaction bin 3 photonuclear reactions (MT=16, 22, 28), reaction bin 4 photonuclear reactions (MT=50 – 91), and reaction bin 5 a photonuclear reaction (MT=98).

12.3.7 Specification of Reaction Rate Calculation

This and the following sections describe the specification of tally types. Unless available tallies are specified, the particle flux is selected by the code. Namely, the default of the tally type is the particle flux.

IRESP

■ Format ■

IRESP (*n*)

■ Meaning ■

Specify energy-dependent response functions (multigroup form) to be used. The *n*-th response function in RESP (p. 228) is used.

PRESP

■ Format ■

PRESP (*n*)

■ Meaning ■

Specify energy-dependent response functions (continuous-energy form) to be used. The *n*-th response function in STAL or CRESP (p. 228) is used.

MACRO – Neutron Reaction

■ Format ■

MACRO (*reaction*)

■ Meaning ■

This parameter is available only in MVP. The neutron macroscopic cross section stored in the bank (SMAC) is used as the energy-dependent neutron response function.

The following reactions can be specified.

- TOTAL : Total
- NUFSSION : Production ($\nu \Sigma_f \phi$)
- FISSION : Fission ($\Sigma_f \phi$)

- ELASTIC : Elastic scattering ($\Sigma_{el}\phi$)
- CAPTURE : Capture ($\Sigma_c\phi$) (Absorption - Fission)
- INELASTIC : Inelastic scattering ($\Sigma_{iel}\phi$)
- N2N : (n,2n) ($\Sigma_{n,2n}\phi$)
- LOSS : Loss ($\Sigma_{loss} = \Sigma_a - \Sigma_{n,2n} - 2\Sigma_{n,3n} - 3\Sigma_{n,4n}$)

MACRO – Photoatomic Reaction

■ Format ■

MACRO (*reaction*)

■ Meaning ■

This parameter is available only in MVP. The photon macroscopic cross section stored in the bank (SMAC) is used as the energy-dependent photon response function.

The following reactions can be specified.

- TOTAL : Total
- COHERENT : Coherent scattering (Thomson scattering)
- INCOHERENT : Incoherent scattering (Compton scattering)
- PAIRPRODUCTION : Electron pair production
- PHOTELECTRIC : photo-electron production

MICRO – Neutron Reaction

■ Format ■

MICRO : *nucid* (*reaction*)

or

MICRO : *nucid* : SPREAD (*reaction*)

■ Meaning ■

This parameter is available only in MVP. The neutron microscopic cross section (SMIC) for the specified nuclide ID is used as the energy-dependent neutron response function.

If “:SPREAD” is specified, the reaction rate per one atom is calculated whether the nuclide exists in the spatial region to be tallied for or not. Unless “:SPREAD” is specified, the reaction rate multiplied by the atomic number density in the tallied region is calculated.

The following reactions can be specified.

- TOTAL : Total
- NUFISSION : Production ($\nu\sigma_f\phi$)

- FISSION : Fission ($\sigma_f \phi$)
- ELASTIC : Elastic scattering ($\sigma_{el} \phi$)
- CAPTURE : Capture ($\sigma_c \phi$) (Absorption - Fission)
- INELASTIC : Inelastic scattering ($\sigma_{iel} \phi$)
- N2N : (n,2n) ($\sigma_{n,2n} \phi$)
- LOSS : Loss ($\sigma_{loss} = \sigma_a - \sigma_{n,2n} - 2\sigma_{n,3n} - 3\sigma_{n,4n}$)

MICRO – Photoatomic Reaction

■ Format ■

MICRO: *atomid* (*reaction*)

or

MICRO: *atomid*: SPREAD (*reaction*)

■ Meaning ■

This parameter is available only in MVP. The photon microscopic cross section (SMIC) for the specified element ID is used as the energy-dependent photon response function.

If “:SPREAD” is specified, the reaction rate per one atom is calculated whether the nuclide exists in the spatial region to be tallied for or not. Unless “:SPREAD” is specified, the reaction rate multiplied by the atomic number density in the tallied region is calculated.

The following reactions can be specified.

- TOTAL : Total
- COHERENT : Coherent scattering (Thomson scattering)
- INCOHERENT : Incoherent scattering (Compton scattering)
- PAIRPRODUCTION : Electron pair production
- PHOTELECTRIC : Photo-electron production

MICROPN – Photonuclear Reaction

■ Format ■

MICROPN: *pn-nucid* (*MT*)

or

MICROPN: *pn-nucid*: SPREAD (*MT*)

■ Meaning ■

This input data calculates the microscopic photonuclear reaction rates with photonuclear cross sections as energy-dependent response functions. This input data is effective only for MVP. If “: SPREAD” is specified, MVP calculates the reaction rates per one nucleus regardless of whether the specified nuclide exists in the tally regions or not. Unless “: SPREAD” is specified, MVP calculates the reaction rates multiplied by the atomic density of the nuclide in the tally region. “*pn-nucid*” is the photonuclear nuclide ID specified as a response function (see Section 9.1.4) and it must be specified in the CROSS SECTION data block. Only the photonuclear nuclide ID’s specified in the CROSS SECTION data block can be used; otherwise, photonuclear cross section data are not read. The reactions (MT number) that can be specified for MICROPN must be the one stored in the photonuclear cross section file. If an MT number whose data is not stored in the file is specified, a fatal error occurs and MVP terminates the calculation. See Appendix G.3 for the available MT numbers. The maximum number of specified MT numbers is 100. If multiple MT numbers are specified, the sum of the reaction rates is calculated; each reaction rate is not calculated.

■ Related options ■

This input data is meaningful only if the option PHOTO-NUCLEAR (p. 107) is effective.

■ Example ■

Case of specification for MT=3 only.

```
MICROPN:U02380JEPN( 3 )
```

Case of specification for multiple MT numbers.

```
MICROPN:FE0560CNPN:SPREAD( 16 50 51 52 53 54 )
```

12.3.8 Specification for Reactor Noise Analysis

ANALOG

■ Format ■

```
ANALOG( reaction )
```

■ Meaning ■

The neutron weights are scored when collision events occur. The following reactions can be specified.

- SCATTERING : the neutron weights are scored when scattering events occur.
- ABSORPTION : the neutron weights are scored when absorption events except for fission occur. ((n, γ), (n,p), \cdots reactions)
- CAPTURE : the neutron weights are scored when capture events occur. (only (n, γ) reaction)
- FISSION : the neutron weights are scored when fission events occur.

■ Note ■

The specification of ANALOG is effective only if both EVENT(ANALOG) (p. 230) and the ANALOG-FISSION option are specified. No nuclide name can be specified for ANALOG.

FEYNMAN-Y

■ Format ■

FEYNMAN-Y (*Ngate ID*)

■ Meaning ■

The Feynman's Y values of Eq.(4.54) are calculated. $Ngate$ is the maximum number of the bunched time bins. Then the maximum gate width is $\tau_{max} = Ngate \times \tau_{min}$, where τ_{min} is the minimum gate width. τ_{min} is specified as the width of equal time bins for TIMEB (p. 224). ID is a tally ID of a time spectrum for which the Y values are calculated. If $ID \leq 0$, the tally data for a time spectrum must be defined in this tally specification. Namely, no tally data for a time spectrum is specified in other tally specifications. In this case, only the Y values are output and the time spectrum is not output.

■ Example ■

```
% TMAX = 2.E-1 , NTBIN = 20000 , DT = TMAX/NTBIN
$TALLY
  NTIME( <NTBIN> )
  TIMEB( 0.0 <%NTIME>:<DT> )
  NWORK( <%NTIME*2> )
& ID(10) NEUTRON
  LABEL( 'SCATTERING-DETECTOR IN REFLECTOR' )
  DIMENSION( TIME REGION )
  ITIME ( 1 <%NTIME-1>:1 )
  REGION( @REFL @ALL )
  EVENT( ANALOG )
  ANALOG( ABSORPTION )
& ID(20) NEUTRON
  LABEL( 'FEYNMAN-Y VALUE OR TALLY ID=10' )
  FEYNMAN-Y( 1000 10 )
$END TALLY
```

12.4 Example of the TALLY Data Block

The following input data is an example for the TALLY data block. The time-dependent calculation is performed for this input data and point-wise (continuous-energy) response functions are used. (The TIME-DEPENDENT and RESPONSE options are specified.) The number of energy group for neutrons (NGROUP.N) is defined in prior to the TALLY data block. The TALLY REGIONS (@D13, @D23, @D33, @D43, @D53, @D63, @D73) are defined in the TALLY REGION data sub-block. The energy bin boundaries are defined outside of the TALLY data block.

Example:

```
$TALLY
NSTAL(6)
STAL ( 13027.03Y ( MT107 )
      28058.03Y ( MT16 )
      28058.03Y ( MT103 )
```

```

41093.03Y ( MT16 )
49115.03Y ( MT51 )
79197.03Y ( MT102 )
)
NTIME(96)
TIMEB(
1.00000E-08 1.25890E-08 1.58490E-08 1.99530E-08 2.51190E-08
3.16230E-08 3.98110E-08 5.01190E-08 6.30960E-08 7.94330E-08
1.00000E-07 1.09650E-07 1.20230E-07 1.31830E-07 1.44540E-07
1.58490E-07 1.73780E-07 1.90550E-07 2.08930E-07 2.29090E-07
2.51190E-07 2.75420E-07 3.02000E-07 3.31130E-07 3.63080E-07
3.98110E-07 4.36520E-07 4.78630E-07 5.24810E-07 5.75440E-07
6.30960E-07 6.91830E-07 7.58580E-07 8.31760E-07 9.12010E-07
9.99999E-07 1.09650E-06 1.20230E-06 1.31830E-06 1.44540E-06
1.58490E-06 1.73780E-06 1.90550E-06 2.08930E-06 2.29090E-06
2.51190E-06 2.75420E-06 3.02000E-06 3.31130E-06 3.63080E-06
3.98110E-06 4.36520E-06 4.78630E-06 5.24810E-06 5.75440E-06
6.30960E-06 6.91830E-06 7.58580E-06 8.31760E-06 9.12010E-06
1.00000E-05 1.09650E-05 1.20230E-05 1.31830E-05 1.44540E-05
1.58490E-05 1.73780E-05 1.90550E-05 2.08930E-05 2.29090E-05
2.51190E-05 2.75420E-05 3.02000E-05 3.31130E-05 3.63080E-05
3.98110E-05 4.36520E-05 4.78630E-05 5.24810E-05 5.75440E-05
6.30960E-05 6.91830E-05 7.58580E-05 8.31760E-05 9.12010E-05
1.00000E-04 1.25890E-04 1.58490E-04 1.99530E-04 2.51190E-04
3.16230E-04 3.98110E-04 5.01190E-04 6.30960E-04 7.94330E-04
1.00000E-03 1.00000E+26 )
&
ID(10)
LABEL( "TIME-DEPENDENT FLUX" )
PARTICLE( NEUTRON )
DIMENSION( REGION TIME ENERGY )
REGION( @D13 @D23 @D33 @D43 @D53 @D63 @D73 )
IENERGY ( 1 <%NGROUP.N-1>:1 )
ITIME ( 1 <%NTIME-1>:1 )
&
ID(20)
LABEL( "TIME-DEPENDENT POINT-WISE RESPONSE FOR NI-58(n,p)" )
PARTICLE( NEUTRON )
DIMENSION( REGION TIME )
REGION( @D43 @D53 @D63 @D73 )
ITIME ( 1 <%NTIME-1>:1 )
PRESP( 3 )
&
ID(30)
LABEL( "REGION-WISE FE-56 TOTAL REACTION RATE/ATOM" )
PARTICLE( NEUTRON )
DIMENSION( REGION )
REGION( @D13 @D23 @D33 @D43 @D53 @D63 @D73 )
MICRO:FE0560J40:SPREAD( TOTAL )
$END TALLY

```

- The neutron flux is calculated for TALLY 10. “DIMENSION(REGION TIME ENERGY)” means the flux score is tallied in the specified regions, time bins and energy bins. There are 7 TALLY REGIONs (@D13, @D23, @D33, @D43, @D53, @D63, @D73) for this tally. “IENERGY

(1 <%NGROUP.N-1>:1)” is the same as “IENERGY (1 2 3 ... NGROUP.N)” and the flux is calculated in each energy bin specified by ENGYB.N. “ITIME (1 <%NTIME-1>:1)” is the same as “IENERGY (1 2 3 ... 95 96)” and the flux is calculated in each time bin specified by TIMEB.

- The Ni-58(*n,p*) reaction rate is calculated for TALLY 20. The reaction rate is calculated in 4 TALLY REGIONs and time bins as for TALLY 10. “PRES(3)” means the use of the point-wise cross section data specified by 28058.03Y (MT103) in STAL.
- The Fe-56 collision rate is calculated for TALLY 30. The collision rate is calculated in the same 7 TALLY REGIONs as for TALLY 10. The calculated results are the collision rate per atom because of the SPREAD option.

12.5 Volume of Spatial Regions

There are the following three types of input data for volumes defined in MVP/GMVP.

- VOL : Volume of ZONEs
- RVOL : Volume of REGIONs
- TRVOL : Volume of TALLY REGIONs

The volume of TALLY REGIONs is required for the volume-averaged results such as particle fluxes and reaction rates estimated with collision and track-length estimators. The user must input the volume to obtain the results because the current version of MVP/GMVP cannot calculate the volume automatically. If the volume of TALLY REGIONs is not input or is set to unity, the volume-integrated results are obtained. The user does not have to know the volume of all TALLY REGIONs before the calculation because the volume-averaged results can be obtained easily from the volume-integrated results after the calculation. These data can be input in or outside of the TALLY data block.

Apart from actual simulation of particles, the user can calculate spatial volumes with the MVP/GMVP code as follows.

- Use the IMAGINARY-PARTICLE and FIXED-SOURCE options.
- Specify a planar source perpendicular to one of the coordinate axes.
- Set the TRVOL values to 1.0 (default).
- Estimate particle flux in spatial regions of interest with the track-length estimator.
- The volume of the spatial regions can be obtained by multiplying the particle flux by the area of the planar source.

VOL

■ Format ■

VOL(vol_1 vol_2 ... vol_NINPZ)

■ Meaning ■

The volume of each INPUT ZONE (cm^3). NINPZ is the number of INPUT ZONES. Currently, this parameter is not used for calculation.

■ Default ■

The volume of each INPUT ZONE is $1.0 (\text{cm}^3)$.

RVOL**■ Format ■**

RVOL (rvol_1 rvol_2 ... rvol_NREG)

or

RVOL (!region-name (volume) . . .)

■ Meaning ■

The volume of each REGION. NREG is the number of REGIONS and is automatically calculated from the geometry data.

■ Default ■

Unless this data is input, this data is calculated as the sum of the volumes of ZONES which comprise each REGION (VOL). (But the repeated ZONES are not taken into account in the repeated geometry.) If no VOL values are also input, all RVOL values are set to $1.0(\text{cm}^3)$.

In the case of the latter input form, the undefined volumes of REGIONS are set to $1.0(\text{cm}^3)$.

TRVOL**■ Format ■**

TRVOL (trvol_1 trvol_2 ... trvol_NTREG)

or

TRVOL (!region-name (volume) . . .)

or

TRVOL (@tally-region-name (volume) . . .)

■ Meaning ■

The volume of each TALLY REGION. NTREG is the number of TALLY REGIONS.

■ Default ■

Unless TRVOL is input, it is calculated as the sum of the volumes of REGIONS which comprise each TALLY REGION (RVOL). (But the repeated REGIONS are not taken into account in the repeated geometry.) If no RVOL values are input, all TRVOL values are set to $1.0(\text{cm}^3)$.

In the case of the input form with REGION or TALLY REGION names, the undefined volumes of TALLY REGIONs are set to 1.0(cm³).

■ Note ■

It is recommended to use the input form with TALLY REGION names.

13. Variance Reduction Data

13.1 Available Variance Reduction Techniques

Four variance reduction techniques are available in MVP/GMVP: Russian roulette, importance, weight window, path stretching. A detail explanation for the variance reduction techniques is described in Section 2.9. To use the variance reduction techniques, the user must specify parameters for them. Table 13.1 summarizes the variance reduction parameters that the user must specify.

Table 13.1 Variance reduction parameters that the user must specify

Variance reduction technique	Parameters	Page
Russian roulette	WKIL, WSRV	p. 253, p. 253
Importance	XIMP	p. 252
Weight window	WKIL, WSRV	p. 253, p. 253
Path stretching	PSALP, PSXYZ	p. 254, p. 254

If the user addresses a problem that includes photonuclear reactions, the user can increase the probability that photons make the photonuclear reactions and collisions with nuclides of interest by using the parameters PPNBR (see Section 13.2) and PFCN (see Section 13.2), respectively.

13.2 Variance Reduction Parameters

XIMP or XIMP.N/XIMP.P

■ Format ■

```
(( XIMP ( i, j ), i=1,NGROUP ), j=1,NREG )
```

or

```
XIMP ( !region-name ( importance-for-energy-groups ) ... )
```

or

```
!region-name ( XIMP ( importance-for-energy-groups ) ... )
```

■ Meaning ■

Importance by the energy group and the REGION. (The default value = 1.0.) NREG is the number of REGIONs and is automatically determined from the geometry data in the codes. This parameter is effective with the IMPORTANCE option.

XIMP.N(NGROUP.N,NREG) and XIMP.P(NGROUP.P,NREG) can be input for each particle in the neutron/photon coupled problems.

■ Note ■

See also Data Input by REGION (p. 82) about the input method.

WKIL or WKIL.N/WKIL.P**■ Format ■**

```
(( WKIL( i, j ), i=1,NGROUP ), j=1,NREG )
```

or

```
WKIL( !region-name( russian-roulette-killing-weight ) ... )
```

or

```
!region-name( WKIL( russian-roulette-killing-weight ) ... )
```

■ Meaning ■

Parameters for Russian roulette. The Russian roulette is performed for particles whose weights are not more than the values of these parameters. (The default value = 0.0001) NREG is the number of REGIONS. These parameters are effective with the RUSSIAN-ROULETTE option or the WEIGHT-WINDOW option.

WKIL.N(NGROUP.N,NREG) and WKIL.P(NGROUP.P,NREG) can be input for each particle in the neutron/photon coupled problems.

■ Note ■

All values of WKIL for energy groups and REGIONS must be smaller than WSRV.

See also Data Input by REGION (p. 82) about the input method.

WSRV or WSRV.N/WSRV.P**■ Format ■**

```
(( WSRV( i, j ), i=1,NGROUP ), j=1,NREG )
```

or

```
WSRV( !region-name( surviving-particle-weights ) ... )
```

or

```
!region-name( WSRV( surviving-particle-weights ) ... )
```

■ Meaning ■

Weights given to survived particles after Russian Roulette or splitting of Weight Window. (The default value = 0.0001.)

The splitting of Weight Window is performed for the particles whose weights are larger than $2 \times$ WSRV. These parameters are effective with the RUSSIAN-ROULETTE option or the WEIGHT-WINDOW option.

WSRV.N(NGROUP.N,NREG) and WSRV.P(NGROUP.P,NREG) can be input for each particle in the neutron/photon coupled problems.

■ Note ■

All values for energy groups and REGIONS must be larger than WKIL.

See also Data Input by REGION (p. 82) about the input method.

PSALP

■ Format ■

(PSALP (j) j=1,NREG)

or

PSALP (!*region-name* (*path-stretching-factor*) ...)

or

!*region-name* (PSALP (*path-stretching-factor*) ...)

■ Meaning ■

Parameters for the PATH-STRETCH (p. 105) option. These are specified by each REGION. See PATH-STRETCH (p. 105) for the detailed meaning.

- More than 0 and less than 1 :
Particles scattered toward the input spatial point PSXYZ (p. 254) are biased such that the flight path is stretched.
- Negative values :
Particles scattered against the input spatial point PSXYZ (p. 254) are biased such that the flight path is stretched.
- Zero :
No PATH-STRETCHING is used.
- Larger than 1 :
Illegal input. No calculation is performed.

■ Default ■

0 for all REGIONs.

■ Note ■

See also Data Input by REGION (p. 82) about the input method.

PSXYZ

■ Format ■

PSXYZ (*px py pz*)

■ Meaning ■

The coordinates of the target point for path stretching if the PATH-STRETCH (p. 105) option is specified. See PATH-STRETCH (p. 105) for the detailed meaning.

PSXYZ is always necessary for the PATH-STRETCH option.

PFCN

■ Format ■

```
PFCN( !region-name( probability nuclide-name-pattern_1
nuclide-name-pattern_2 . . . ) . . . )
```

For a case where common data is given to all regions,

```
PFCN( probability nuclide-name-pattern_1 . . . )
```

■ Meaning ■

This parameter gives the probability that photons are forced to make collisions with specified nuclides in each REGION. The specified nuclides in the REGIONS are sampled with the input probability. The probability given with 0 means that MVP does not use forced collision for photons; it does not mean that photons do not make collisions. The probability of forced collisions for photons should be $0 < P_{\text{PFCN}} < 1$. See PRODUCE-NUCLIDE (p. 240) for the specification of nuclide patterns. See also Section 6.6 for the specification of REGION name patterns.

■ Default ■

0.0 for all nuclides in all REGIONS. No forced collisions are made for photons.

■ Related options ■

This parameter is meaningful only if the option PHOTO-NUCLEAR (p. 107) is effective.

■ Note ■

Input nuclides for forced collisions do not necessarily exist in the specified REGIONS. Only nuclides that exist in each specified REGION are used for forced collisions. (Unless an input nuclide exists in the specified REGION, it means $P_{\text{PFCN}}=0$.) When a nuclide including the photonuclear reaction is sampled, the weight of incident photons is adjusted with the probability and the specification of forced collisions. The user should pay attention to the fact that all collision events will not occur if the probability of forced collisions is extremely small. The probability should be $P_{\text{PFCN}} > 1.0\text{E-}5$ for usual cases.

■ Example ■

```
PFCN( !RG01( 0.1 U0235 U0238 ) !RG02( 0.05 FE* ) !RG03( 0.02 FE056 ))
PFCN( 1.0E-2 U0235 U0238 PU* )
```

PPNBR

■ Format ■

```
PPNBR( probability( nuclide-name-pattern_1 nuclide-name-pattern_2 . . . ) . . . )
```

For a case where common data is given to all nuclides,

```
PPNBR( probability )
```

■ Meaning ■

The probability that the photonuclear reaction is selected when photons make collisions with nuclei in a coupled neutron and photon calculation for MVP. The probability of the photonuclear reaction must be $0 < P_{\text{PPNBR}} < 1$. See PRODUCE-NUCLIDE (p. 240) for the specification of nuclide patterns.

■ Description ■

Photonuclear reaction cross sections are about 0.1 to 0.001 times larger than photoatomic reaction cross sections. Therefore, almost all the collision processes of photons are photoatomic interactions. This input parameter is used to increase the selection probability of the photonuclear reactions. This parameter is effective only if the photon energy is higher than the lowest energy of the photonuclear reactions.

The photon weight is adjusted with the parameter PPNBR as follows. If a photonuclear reaction is selected with the probability of PPNBR, then

$$W'_\gamma = W_\gamma \frac{\sigma_t^{\text{pn}}}{\sigma_t^{\text{pa}} + \sigma_t^{\text{pn}} \cdot \text{PPNBR}}.$$

If a photoatomic reaction is selected with the probability of (1-PPNBR), then

$$W'_\gamma = W_\gamma \frac{\sigma_t^{\text{pa}}}{\sigma_t^{\text{pa}} + \sigma_t^{\text{pn}} \cdot (1 - \text{PPNBR})}.$$

The meanings of the parameters are as follows.

- W'_γ : photon weight after adjustment,
- W_γ : photon weight before adjustment,
- σ_t^{pa} : total cross section of the photoatomic reaction,
- σ_t^{pn} : total cross section of the photonuclear reaction.

■ Default ■

The default values = 0.0. The selection probability of the photonuclear reactions is $\sigma_t^{\text{pn}} / (\sigma_t^{\text{pa}} + \sigma_t^{\text{pn}})$.

■ Note ■

If the last probability data is given without specifying any nuclides in the first format input data, the input data is used for remaining unspecified nuclides.

■ Example ■

```
PPNBR( 0.3 (U0235 U0238) 0.5 (FE*) 0.1 )
PPNBR( 0.3 (U0235 U0238) 0.2 (FE054 FE056) 0.1 )
PPNBR( 0.01 (N0014 O0016) )
PPNBR( 0.3 )
```

PMT

■ Format ■

```

PMT ( !region-name ( ( nuclide-name reaction-mt probability ) ... ) ... )
PMT ( !region-name ( nuclide-name reaction-mt probability ) ... )
PMT ( !region-name ( reaction-mt probability ) ... )

```

If a common value is given to all regions,

```

PMT ( ( nuclide-name reaction-mt probability ) ... )

```

If a common value is given to all regions and nuclides,

```

PMT ( reaction-mt probability )

```

■ Meaning ■

PMT is the probability, P_{PMT} , that a photonuclear reaction type of interest is selected when MVP selects a photonuclear reaction. The user can specify only one reaction type for a nuclide in a region. P_{PMT} must be $0 < P_{\text{PMT}} < 1$. If $P_{\text{PMT}} = 0$, the sampling for the selection is made with unbiased probabilities with the cross sections for reaction types. If a specified value for PMT is smaller than the unbiased probability, the sampling is made with the unbiased one. The MT number of a reaction type is one defined in the photonuclear data library (See Section H.3). If the specification of nuclides is omitted, the specified probability is applied for all the nuclides in a region. See PRODUCE-NUCLIDE (p. 240) for how to specify the nuclides. Note that the user cannot use “+” or “-” for the connection of nuclide ID’s. See Section of “Data Input by REGION (p. 82)” for how to specify REGIONS.

■ Description ■

The photofission cross section for U-238 is about 30% of the total cross section in the energy range between 7 MeV and 15 MeV. If the user wants to increase the statistics for delayed neutrons, it is effective to increase the sampling probability of photofission reactions. For example, the user inputs as `PMT ((U0238 18 0.5))` and namely specifies $P_{\text{PMT}} = 0.5$ so that the sampling probability becomes larger than the unbiased probability.

■ Default ■

The default value = 0.0. The unbiased probabilities are used.

■ Related options ■

This parameter is meaningful only if the option PHOTO-NUCLEAR (p. 107) is effective.

14. Secondary Particle Data

14.1 Generation of Secondary Particles

Particles generated directly from sources are called primary particles. On the other hand, fission neutrons and photons generated from the reactions between neutrons and nuclei are called secondary particles in MVP/GMVP.

If the secondary particles are generated in every neutron reaction, the array to store the particles' information (particle bank) may be overflowed. Thus the generation of the secondary particles is determined by performing "Russian Roulette" according to the neutron weight in collision, the production rate of the secondary particles in collision and the reference weight given in the input data.

The reference weight for fission neutrons or secondary photons is WGTF or WGTP, respectively.

If photonuclear reactions are treated explicitly, the primary particles of photons generate the secondary particles of neutrons. The number on the secondary neutrons are then determined with the weight of the photons in collision, the production probability and the yield of secondary neutrons and the reference weight given to the secondary neutrons (WGTPN or WGTPNR).

14.2 Data for Fission Neutrons and Secondary Photons

FKAI

■ Format ■

```
FKAI (  $\chi_{1,1}$   $\chi_{2,1}$  ...  $\chi_{NTGX,1}$ 
       $\chi_{1,2}$   $\chi_{2,2}$  ...  $\chi_{NTGX,2}$ 
      ...
       $\chi_{1,NMAT}$   $\chi_{2,NMAT}$  ...  $\chi_{NTGX,NMAT}$  )
```

■ Meaning ■

Fission spectrum by the material for GMVP. NTGX and NMAT are the number of energy groups for macroscopic cross sections and materials, respectively. NTGX and NMAT are equal to NGP+NGG and NMED, respectively. (See Section 9.3.4.)

■ Note ■

FKAI should be specified for each material in the CROSS SECTION data block.

WGTF or WGTFD

■ Format ■

- (1) For a case where the same value is given to prompt and delayed fission neutrons:

```
WGTF ( weight_1 weight_2 ... weight_NREG )
```


or

WGTF (*!region-name* (*weight*) ...)

or

!region-name (WGTF (*weight*) ...)

(2) For a case where the different values are given to prompt and delayed fission neutrons:

WGTFD (*!region-name* (*prompt-weight* *delayed-weight*) ...)

WGTFD (*!region-name* (*prompt-weight*) *!region-name* (*prompt* *delayed*) ...)

or

WGTFD (*prompt-weight_reg1* *delayed-weight_reg1* *prompt_reg2* *delayed_reg2* ...)

■ Meaning ■

Weights given to fission neutrons in each REGION. NREG is the number of REGIONS. Whenever a neutron collides with a nucleus, the number of fission neutrons is determined as follows:

$$n_f = \frac{W \cdot \nu \sigma_f / \sigma_t}{\text{WGTF}}$$

where W is the weight of a neutron in collision. See also Data Input by REGION (p. 82) about the input method. If the DELAYED-NEUTRON option is effective, the different values can be given to the weights for prompt and delayed fission neutrons. In this case, WGTFD is used. Unless the delayed neutron data exist in the neutron library, the weight for delayed neutrons is meaningless.

■ Default ■

1.0 for all REGIONS.

■ Description ■

● **Case of eigenvalue problems** In eigenvalue problems, fission neutrons are stored in the fission bank in a batch (generation) and then as many fission neutrons as the number of histories per batch NHIST (p. 115) are selected with the equal probability and used as the source for the next batch. In this case, it is desired that the fission neutrons not less than NHIST are stored. Unless it is satisfied, the generation point and the weight of some fission neutron are used more than once. It may cause the large correlation between batches and then the estimated variance of the eigenvalue may become less reliable.

In order to prevent this situation, the averaged value of WGTF should be smaller than a predicted eigenvalue in principle. To be precise, the averaged WGTF value over regions with the weight of the production rate should be smaller. Usually, it should be the predicted eigenvalue multiplied by about 0.8. Since the default values are 1.0, the smaller values should be set for subcritical systems.

When the fission neutrons stored in the fission bank are used as the neutrons source in the next batch (generation), the sum of the weights are normalized to $\text{NHIST} \times 1.0$. Thus the weights of the generated particles in the next batch are not always WGTF.

● **Case of fixed-source problems** The number of fission neutrons increases if WGTF is a small value. Thus it is expected that the statistics is improved for the problems where the behavior of the secondary particles or later is important. But when the variance reduction techniques such as Russian Roulette or Weight Window are used, the random walk of the secondary particles with small weights may be terminated immediately. To prevent it, the RELATIVE-WEIGHT (p. 109) option should be used.

● **Space-dependent WGTF** The value of WGTF can be specified in each REGION. This means the capability of biased sampling. That is,

- A lot of fission neutrons are generated in the regions of the small WGTF value.
- Only a few fission neutrons are generated in the regions of the large WGTF value.

WGTP

■ Format ■

WGTP (*weight_1 weight_2 ... weight_NREG*)

or

WGTP (*!region-name (weight) ...*)

or

!region-name (WGTP (weight) ...)

■ Meaning ■

Weights given to secondary photons generated by neutron reactions in each REGION. NREG is the number of REGIONs. The number of the secondary photons is determined as follows:

$$n_p = \frac{W \cdot \sigma_p / \sigma_t}{WGTP}$$

where W is the weight of a neutron at a collision and σ_p is the photon production cross section. See also Data Input by REGION (p. 82) about the input method.

■ Default ■

1.0 for all REGIONs.

■ Note ■

The similar notes to “Case of fixed-source problems (p. 260)” and “Space-dependent WGTF (p. 260)” held for WGTP.

WGTPN or WGTPNR

■ Format ■

(1) For a general case (where the parameter depends on REGIONs, nuclides and reaction types):

WGTPN (*!region-name (weight (nuclide-name_1 mt_1 nuclide-name_2 mt_2 ...) ...) ...*)

WGTPN (*!region-name (weight (nuclide-name_1 nuclide-name_2 mt_2 ...) ...) ...*)

- (2) For a case where the parameter depends on nuclides and reaction types only:

```
WGTPN( weight ( nuclide-name_1 mt_1 nuclide-name_2 mt_2 ... ) ... )
WGTPN( weight ( nuclide-name_1 nuclide-name_2 mt_2 ... ) ... )
```

- (3) For a case where the parameter depends on REGIONs and nuclides only:

```
WGTPN( !region-name( weight ( nuclide-name-pattern_1 ... ) ... ) ... )
```

- (4) For a case where the parameter depends on nuclides only:

```
WGTPN( weight ( nuclide-name-pattern_1 ... ) ... )
```

- (5) For a case where the parameter depends on REGIONs only:

```
WGTPNR( weight_1 weight_2 ... weight_NREG )
```

or

```
WGTPNR( !region-name( weight ) . . . )
WGTPN( !region-name( weight ) ... )
```

or

```
!region-name( WGTPNR( weight ) ... )
```

■ Meaning ■

This parameter specifies the weight given to secondary neutrons that are generated in photonuclear reactions. This parameter can be given for each REGION, each nuclide and each reaction type. This parameter is available for MVP. In the above format, NREG is the number of REGIONs. The number of the secondary neutrons, n_{pn} , is determined with WGTPN as follows.

$$n_{pn} = \frac{W Y_{pn} \sigma_{pn} / \sigma_t}{WGTPN},$$

where W is the weight of a photon in collision, Y_{pn} is the yield of neutrons from photonuclear reactions, σ_{pn} is the photonuclear reaction cross section, and σ_t is the sum of the photonuclear and photoatomic reaction cross sections. See PRODUCE-NUCLIDE (p. 240) for the specification of nuclide name patterns. See also Section Data Input by REGION (p. 82) for the specification of REGION name patterns. If REGION names are omitted (independent of REGIONs), the same value is used for all the REGIONs. The MT number of a reaction type is the one defined in the photonuclear data library. See Section H.3 for the MT numbers of reaction types. Unless reaction types are specified, the same value is used for all reaction types. If the user wants to specify different values for multiple reaction types of a nuclide, multiple sets of a nuclide ID and a reaction type must be specified. If the DELAYED-NEUTRON option is effective, different values can be given to prompt and delayed neutrons generated in photofissions. In this case, the MT numbers are 18 for prompt neutrons, and 98 or -18 for delayed neutrons. Unless delayed neutron data exist in the photonuclear data library, the value for the delayed neutrons is meaningless.

■ Default ■

0.001 for all REGIONs. Namely, the weight of all particles generated in photonuclear reactions is set to 0.001.

■ Related options ■

This parameter is meaningful only if the option PHOTO-NUCLEAR (p. 107) is effective. If the DELAYED-NEUTRON option (p. 95) is effective, the WGTPN values can be applied to delayed neutrons.

■ Note ■

The different weights can be given to each nuclide in the same REGION.

If MONPNWGT (p. 262), which is the input parameter for monitoring weights of particles generated in photonuclear reactions, is used, WGTPN is automatically set to 0. In this case, unbiased probabilities are used and the weight of an incident photon is passed to secondary particles. It is the same as the case where WGTPN is set to 0.

The similar notes to “Case of fixed-source problems (p. 260)” and “Space-dependent WGTF (p. 260)” held for WGTPN.

■ Example ■

```
WGTPN( !DCONC( 1.E-5(U0*) 5.E-4(C0* SI* S0* CR* MN055 FE* NI*) ) )
WGTPN( !RG01 (1.0E-4) !RG02 (2.0E-4) !RG03 (5.0E-4) )
WGTPN( 0.005 ( U0* FE* CU* ) )
WGTPNR( 28( 0.001 ) )
WGTPN( !DCONC(1.E-5(U0235 18 U0235 98)
          2.E-5(NP237 U0238 18 U0238 -18) ) )
```

MONPNWGT

■ Format ■

```
MONPNWGT( !region-name-pattern( ( nuclide-name-pattern mt-pattern ) ... ) ... )
MONPNWGT( !region-name-pattern( nuclide-name-pattern mt-pattern ) ... )
MONPNWGT( !region-name-pattern( nuclide-name-pattern ) ... )
MONPNWGT( !region-name-pattern( mt-pattern ) ... )
MONPNWGT( !region-name-pattern ... )
```

■ Meaning ■

This parameter specifies REGIONS, nuclides and reaction types for collisions with photons to monitor the weight of secondary particles generated in photonuclear reactions. Only the weights for the specified REGIONS, nuclides and reaction types are output. This parameter can be specified for MVP only. If this parameter is specified, WGTPN (p. 260) is automatically set to 0. The MT number of a reaction type is the one defined in the photonuclear data library. See Section H.3 for the MT numbers of reaction types. See PRODUCE-NUCLIDE (p. 240) for the specification of nuclide name patterns. See also Section Data Input by REGION (p. 82) for the specification of REGION name patterns. “+” can be used to connect REGION names, photonuclear nuclide ID’s and reaction types, but “-” cannot be used. If the user specifies REGION names only, all nuclides and reaction types in the REGIONS are monitored. The monitoring bin is assigned in turn for each set of a pattern innermost specified with parentheses in the input data. Multiple nuclides and reaction types can be specified for the same bin. If the user wants to specify each reaction type of a nuclide in the same REGION for different bins, the nuclide can be specified in multiple times. For the first line in the following example, monitoring bin 1 is for MT=5 of Fe-56 in REGION RG01, monitoring bin 2 is for MT=5 of Fe-54 and Fe-56, and monitoring bin 3 is for MT=5 and MT=18 of U-235 and U-238.

■ Default ■

Not effective.

■ Related options ■

This parameter is meaningful only if the option PHOTO-NUCLEAR (p. 107) is effective.

■ Example ■

```
MONPNWGT( !RG01( (FE056 5) (FE054+FE056 5) ) !RG02 (U0235+U0238 5+18) )
MONPNWGT( !RG01(FE056) !RG02 (U 5+18) )
MONPNWGT( !RG01(18) )
MONPNWGT( !RG01 !RG02 )
```

15. Perturbation Data

15.1 General Rules of Input Data for Perturbation Calculations

15.1.1 PERTURBATION Data Block

The general form of the PERTURBATION data block is given as follow. This is a similar form to the TALLY data block.

```

$PERT[URBATION]
[ Common data to all perturbation data sets ] (currently no data)
&
Data set for perturbation 1
&
Data set for perturbation 2
&
...
$END PERT[URBATION]
```

- \$ and & must be located at the first column of an input line and a data set of a perturbation is terminated by the next & or \$END PERTURBATION. A data set corresponds to a tally to calculate the change in a physical parameter (k_{eff} , reaction rate, flux, etc.) with regard to a perturbation parameter (density, number density, etc.). For example, when the user calculates the first-order change in the effective multiplication factor with regards to a density perturbation, the tally scores contributions to $\partial k / \partial \rho$.
- The PERTURBATION data block must be located after the GEOMETRY data block (\$GEOM — \$END GEOM) and the CROSS SECTION data block (\$XSEC — \$END XSEC).

Table 15.1 shows the list of input parameters for perturbation calculations.

15.2 Input Data for Each Perturbation

15.2.1 ID

■ Format ■

ID(*perturbation_id_number*)

■ Meaning ■

An ID number for a perturbation data set. It must be a positive integer.

15.2.2 LABEL

■ Format ■

`LABEL ("label_string")`

■ Meaning ■

Title for a perturbation data set. It is used for printing output.

15.2.3 Estimator

■ Format ■

`EVENT (event)`

■ Meaning ■

Specification of an estimator. Currently, only *event* = COLLISION is available.

■ Default ■

Collision estimator (*event* = COLLISION).

15.2.4 Tally Particle

■ Format ■

`NEUTRON`

■ Meaning ■

A kind of particles for tallying. Currently, only NEUTRON is available.

■ Default ■

Neutron (NEUTRON).

15.2.5 Tally Parameter

■ Format ■

`KEFF | BEFF`

■ Meaning ■

A physical parameter perturbed by a perturbation parameter. This parameter must be always specified. One of the following parameters can be specified for each data set.

- KEFF : Perturbation for effective multiplication factor (k_{eff}).

- BEFF : Kinetics parameters (effective delayed neutron fraction β_{eff} and neutron generation time Λ).

If more than one parameter are specified for a data set, a fatal error will occur.

Specifying BEFF in the PERTURBATION data block calculates the kinetics parameters with differential operator sampling. The details on the methods are described in references 62 and 63.

15.2.6 Perturbation Parameter

■ Format ■

```
PARAMETER( DENSITY | NUMBER( nucid ) )
```

■ Meaning ■

A parameter to which a perturbation is introduced. This parameter must be always specified. One of the following parameters can be specified for each data set.

- DENSITY : Density perturbation.
- NUMBER : Number (atomic) density perturbation. (*nucid* is a nuclide ID specified in the \$XSEC block.)

15.2.7 Perturbation Calculation Method

■ Format ■

```
METHOD( CORRELATED | DIFFERENTIAL )
```

■ Meaning ■

Monte Carlo perturbation method with which the change in a physical parameter is calculated. One of the following parameters can be specified for each data set.

- CORRELATED : Correlated sampling.
- DIFFERENTIAL : Differential operator sampling.

Currently, the correlated sampling can be used for density perturbations (PARAMETER(DENSITY)). The differential operator sampling method calculates the Taylor series terms up to the second order.

■ Default ■

Differential operator sampling (METHOD(DIFFERENTIAL)).

15.2.8 Number of Generations for Perturbed Source Effect

■ Format ■

PERT[TURBED]-SOUR[CE] (*NGSP*)

■ Meaning ■

When the perturbed source effect is calculated, the contributions from *NGSP* generations ago are taken into account. This option is active only for eigenvalue calculations; it is meaningless for fixed-source calculations. For the fixed-source calculations, *NGSP* is set to 0 automatically in the code.

■ Default ■

NGSP = 10.

15.2.9 Perturbed Quantity

■ Format ■

FRACTIONAL-CHANGE (*value1 value2 ...*)

■ Meaning ■

Fractional change in a perturbation parameter from a reference (unperturbed) value. Multiple values can be specified for differential operator sampling but the number of input values is limited to 10 in the current version. For correlated sampling, only the first value is used; the other values are ignored.

15.2.10 Perturbed Region

■ Format ■

PERT[TURBED]-REG[ION] (*perturbed_region_1* [*perturbed_region_2 ...*])

■ Meaning ■

A region perturbed by the change in a perturbation parameter. *perturbed_region_?* is a TALLY REGION name defined in the last subsection of the GEOMETRY data block (\$GEOM — \$END GEOM): a character string which starts with @ and whose length is less than or equal to 12 characters including @. MVP/GMVP region names with the hierarchical path can be also used for the specification; their input format is the same as for TALLY REGION names with ! and : characters. The wildcards of ? and * can be used (? means any character and * does any characters with any length).

15.2.11 Perturbed Nuclide

■ Format ■

PERT[TURBED]-NUCL[IDE] (*perturbed_nuclide_1* [*perturbed_nuclide_2 ...*])

■ Meaning ■

Nuclide name to which a perturbation is introduced. The user specifies a nuclide ID defined in the CROSS SECTION data block (\$CROSS SECTION — \$END CROSS SECTION). This input data is for a number density perturbation of `PARAMETER(NUMBER)`. If this input data is specified with `PARAMETER(DENSITY)`, a warning message is output and all the nuclides in the perturbed region are perturbed.

When arbitrary temperature libraries are used, the nuclide ID's are changed automatically in the code by adding temperature indices. In such cases, the user must pay attention for the ID names for perturbed nuclides; The wildcards are available like `U02350J40*`.

Table 15.1 List of input parameters for perturbation calculations

Input item	Meaning	Default	Notes
ID	Positive integer	None	
LABEL	Title for a data set	None	
EVENT	Estimator	COLLISION	Currently only COLLISION.
NEUTRON	Tally particle	NEUTRON	Currently, only NEUTRON.
KEFF BEFF	Tally parameter	KEFF	
PARAMETER	Perturbation parameter	None	
METHOD	Perturbation method	None	
FRACTIONAL-CHANGE	Fractional change in a perturbation parameter	None	Only the first value is active for CORRELATED. (Maximum = 10).
PERTURBED-SOURCE	Number of generations for perturbed source effect.	10	Effective only for eigenvalue problems.
PERTURBED-REGION	Perturbed region	None	Specified with TALLY REGION NAMES.
PERTURBED-NUCLIDE	Perturbed nuclide	None	Specified with nuclide ID's.

15.2.12 Example of PERTURBATION Data Block

The following example shows an input data to calculate the change in the effective multiplication factor (k_{eff}) with regard to density perturbations for homogeneous geometry of one region. This input data calculates the change in k_{eff} for density perturbations from a reference (unperturbed) density of 18.74 g/cm³ to 20.0, 21.0, 23.5, 26.0 g/cm³ (fractional changes of 6.7%, 12.6%, 25.4%, 38.74%, respectively).

Example 1:

```
$PERTURBATION
&
  ID(1)
  LABEL('DIFFERENTIAL OPERATOR SAMPLING')
  EVENT(COLLISION)
```

```

METHOD(DIFFERENTIAL)
PARAMETER(DENSITY)
FRACTIONAL-CHANGE(<20.0/18.74 - 1.0> <21.0/18.74 - 1.0>
                  <23.5/18.74 - 1.0> <26.0/18.74 - 1.0>)
PERTURBED-REGION(!FUEL1)
&
ID(2)
LABEL('CORRELATED SAMPLING')
EVENT(COLLISION)
PARAMETER(DENSITY)
METHOD(CORRELATED)
FRACTIONAL-CHANGE(<20.0/18.74 - 1.0>)
PERTURBED-REGION(!FUEL1)
&
ID(3)
LABEL('CORRELATED SAMPLING')
EVENT(COLLISION)
PARAMETER(DENSITY)
METHOD(CORRELATED)
FRACTIONAL-CHANGE(<21.0/18.74 - 1.0>)
PERTURBED-REGION(!FUEL1)
&
ID(4)
LABEL('CORRELATED SAMPLING')
EVENT(COLLISION)
PARAMETER(DENSITY)
METHOD(CORRELATED)
FRACTIONAL-CHANGE(<23.5/18.74 - 1.0>)
PERTURBED-REGION(!FUEL1)
&
ID(5)
LABEL('CORRELATED SAMPLING')
EVENT(COLLISION)
PARAMETER(DENSITY)
METHOD(CORRELATED)
FRACTIONAL-CHANGE(<26.0/18.74 - 1.0>)
PERTURBED-REGION(!FUEL1)
$END PERTURBATION

```

- The perturbed region is specified as FUEL1, which is defined in the geometry data block.
- The perturbation data set for ID(1) calculates the change in the effective multiplication factor with differential operator sampling. The differential operator sampling method calculates the differential coefficients of the effective multiplication factor with regard to a perturbation parameter; thus calculations for any change in the perturbation parameter can be possible with only one perturbation data set.
- The data sets specified from ID(2) to ID(5) calculate the change in the effective multiplication factor with correlated sampling. The correlated sampling method calculates only one change in the effective multiplication factor with one perturbation data set. Thus four data sets are required for the same calculation as with the data set of ID(1).

The following input data shows an example to calculate the kinetics parameters. Neither perturbed regions nor perturbed nuclides are required for this calculation; the user should specify BEFF only.

Example 2:

```
$PERTURBATION
&
  ID(1)
  LABEL('KINETICS PARAMETERS')
  BEFF
$END PERTURBATION
```

16. I/O Files

16.1 FORTRAN I/O Unit

Table 16.1 shows FORTRAN I/O units used in both MVP and GMVP codes and the contents of files allocated on the units. Tables 16.2 and 16.3 show I/O units and files specific to MVP and GMVP, respectively.

The I/O unit for the binary form of P_L cross sections for GMVP can be specified by INPUT or IXTAPE in the CROSS SECTION data block but it should be the default value (=1). The function to output the processed cross sections in GMVP should not be used.

Table 16.1 I/O files common to MVP/GMVP

I/O unit	Format	Contents	I/O
5	Text	Standard input data.	Input
6	Text	Output (standard output).	Output
8	Binary	Neutron source file. It is output as the fission source file in a previous run.	Input
9	Binary	Fission source output in the eigenvalue problem.	Output
10	Binary	Restart file to be input. It is output as the restart file in a previous run.	Input
15	Text	Temporary file to process the geometry data.	Input/Output
16	Binary	Temporary file to process the input data.	Input/Output
20	Binary	Restart file to be output.	Output
30	Binary	Binary output file of calculation results.	Output
55	Text	Temporary file to process the input data.	Input/Output
66	Text	Monitoring output in the process of the input data (arithmetic operation).	Output
75	Text	Temporary file to process the geometry data.	Input/Output

16.2 Specification of I/O Files in a Command Line

When the user executes MVP or GMVP in the UNIX or MS-DOS environment, the user can allocate the required files on the I/O units and provide the information of execution control for MVP/GMVP executables through the command-line parameters. In the UNIX environment, it is more convenient to use the runmvp and rungmvp commands described in Chapter 17 rather than to run the executables in the command line.

(1) Allocation of files on FORTRAN I/O units

The command-line parameters in the following form must be specified.

"/N" + "Control character" + ":" + "File name" ,

Table 16.2 I/O files of MVP

I/O unit	Format	Contents	I/O
24	Text	Index file for arbitrary-temperature neutron cross section libraries.	Input
25	Text	Index file for fixed-temperature neutron cross section libraries.	Input
26	Text	Index file for the photoatomic reaction cross section library.	Input
27	Text	Index file for the electron cross section library (Necessary for Bremsstrahlung X-rays).	Input
28	Text	Index file for the photonuclear reaction cross section library.	Input
29	Text	Index file for continuous-energy response libraries.	Input
49	Binary	Temporary file for the input or processing of the arbitrary-temperature neutron cross section libraries.	Input
50	Binary	Temporary file for the input of the neutron cross section libraries.	Input
51	Text	Temporary file for the exclusive control in outputting the processed results of the arbitrary-temperature neutron cross section libraries.	Output

Table 16.3 I/O files of GMVP

I/O unit	Format	Contents	I/O
INPUT	Binary	Input of the P_L cross section library or the processed library.	Input
OUTPUT	Binary	Output of the processed library.	Output
99	Binary	Input of the double-differential neutron cross section library.	Input
98	Binary	Input of the double-differential photon cross section library.	Input

where

(a) N: I/O unit number (1 through 99)

(b) Control character:

< FORM parameter >

- U or u : Binary file (FORM=UNFORMATTED)
- F or f : Text file (FORM=FORMATTED)

The default character is “U”.

< STATUS parameter >

- S or s : Scratch file (STATUS=SCRATCH)

The file is deleted after processing. But if the specified file exists, it is not deleted.

- R or r : Existing file (STATUS=OLD)
Usually the file to be read only is specified.
- N or n : New file (STATUS=NEW)
- Default : When the file name is given, STATUS=UNKNOWN. When the file name is not given, STATUS=SCRATCH.

Examples are shown in the following.

- /5rf:test.inp
Open the text file named test.inp on I/O unit 5.
- /99:DDL.lib or /99ru:DDL.lib
Open the binary file named DDL.lib on I/O unit 99.
- /30:RESULT or /30un:RESULT
Open the new file named RESULT on I/O unit 30.
- /15sf or /15f
Open a temporary file of the text form on I/O unit 15.
- /16s or /16
Open a temporary file of the binary form on I/O unit 16.

(2) Other control information

- Specification of the number of tasks in multi-tasking
Form : NTASK=n
where
NTASK= : indicates the Specification of the number of tasks. ("ntask=" is also allowed.)
n : the number of tasks

This parameter is ignored for executables without the multi-tasking function.

16.3 Index File for MVP Cross Sections

In the use of MVP, the user must provide the "cross section index file" that indicates the relation between nuclide IDs in the input data and cross section files instead of the direct specification of the files. The index files are accessed on the different FORTRAN I/O units depending on the type of cross sections as listed in Table 16.4.

The index files for neutrons, photons and electrons are the text files written in the following format and relate nuclide ID names in the input data and the cross section files.

- (1) The nuclide ID name and the following full path name of the cross section file are written in one line. They can be placed anywhere in the 1st through 72nd columns.

Example: Format of a nuclide ID name and the file name

(Nuclide ID)	(File name)
H00010J40	/home/MVPLIB/MVPlibV3.H00010J40.dat

Table 16.4 MVP cross section index files and I/O units

I/O unit	Type of cross sections
24	Arbitrary-temperature neutron cross sections
25	Fixed-temperature neutron cross sections
26	Photoatomic reaction cross sections
27	Electron cross sections (for Bremsstrahlung X-rays in photon problems)
28	Photonuclear reaction cross sections
29	Continuous-energy response library

- (2) In order to specify flexibly the path to files on the UNIX or MS-DOS (MS-DOS, Windows etc.) file systems, the “path name” (including directory names and a drive ID) can be defined apart from the file names. The line is called “PATH line” and it can be specified as follows.

PATH (path-name)

The path name is prepended to the file name for each nuclide. A current PATH line is effective until another PATH line is specified.

Unless the PATH line is specified, the default path name is the name of the directory where the index file exists in the UNIX or MS-DOS (Windows) environment.

Example: The next expression is the same as the above example.

```
PATH /home/MVPLIB/
H00010J40      MVPlibV3.H00010J40.dat
```

Example of the MS-DOS (Windows) environment.

```
PATH C:\user\MVPLIB\
H00010J40      MVPlibV3.H00010J40.dat
```

The path name can be specified by not only the absolute path from the root directory but also the relative path from the directory where the index file exists. The character string of the “\$VAR” form (“\$VAR” is an arbitrary character string without directory delimiters.) is also available in the index files and it is replaced with the environment variable of “\$VAR”.

- (3) Blank lines are ignored and the line of which first column is “*” is a comment line.

16.4 Cross Section Files for GMVP

There are two types of cross section files in the binary form available for GMVP. One is the P_L cross sections (ANISN form) and the other is the double-differential cross sections (DDX form).

The binary form of the P_L cross sections is specified by PLTYPE in the CROSS SECTION data block.

- (1) P_L cross sections

(a) PLTYPE(ANISN)

Cross section form of the ANISN code. It includes the following record by the nuclide and the order of the Legendre expansion.

- IGM,IHM,IC,MATNO,TITLE
- ((CRX(i,j),i=1,IHM),J=1,IGM)

(b) PLTYPE(ANISN-MAIL)

The contents of the cross sections is the same as the ANISN-type ones but the cross sections of each energy group are written in each record. It is the form which the MAIL code⁵⁹⁾ in the JACS system outputs.

- IGM,IHM,IC,MATNO,TITLE
- Each single record for J=1,IGM (CRX(i,j),i=1,IHM)

For both (a) and (b),

- IGM : Number of energy groups,
- IHM : Table length per one group,
- IC : Dummy data, (Usually it is the order of the Legendre expansion etc. but it depends on the platforms.)
- MATNO : ID number of each nuclide and each expansion order,
- TITLE : Title (48 characters),
- CRX : Cross section data.

(2) Neutron double-differential cross sections (I/O unit 99)

Each nuclide has the following records.

- MAXN,IDUM,IDUM,MATNO,TITLE
- (MAXSD(i),i=1,MAXN)
- ((CX(j,i),j=1,5),i=1,MAXN)

For $g'=1, \text{MAXN}$,

- ((I(k,g' → g),k=1,MAXMU),i=g',g'+MAXSD(g')-1)
 - MAXN : Number of energy groups
 - IDUM : Dummy data
 - MATNO : ID number of each nuclide
 - TITLE : Title (48 characters)
 - CX : Cross section data
 - * j=1: Scattering cross section for secondary neutron production
 - * j=2: Fission cross section
 - * j=3: Capture cross section
 - * j=4: $\nu \times$ fission cross section
 - * j=5: Total cross section
 - MAXMU : Number of scattering angle bins (currently fixed as 20.)
 - I : Energy-angular distribution of scattering

(3) Photon double-differential cross sections (I/O unit 98)

Each nuclide has the following record.

- MAXP, IDUM, MAXN, MATNO, TITLE
- (MAXSD(i), i=1, MAXP)
- ((CX(j,i), j=1, 5), i=1, MAXP)

For $g'=1, \text{MAXP}$,

- ((I(k, $g' \rightarrow g$), k=1, MAXMU), i= $g', g' + \text{MAXSD}(g') - 1$)
- ($\sigma_p(i)$, i=1, MAXN)
- ((Ip($g' \rightarrow g$), $g=1, \text{MAXP}$), $g'=1, \text{MAXN}$)
 - MAXP : Number of energy groups for photons
 - MAXN : Number of neutron energy groups to give the photon production cross sections
 - IDUM : Dummy data
 - MATNO : ID number of each nuclide
 - TITLE : Title (48 characters)
 - CX : Photon cross section data
 - MAXMU : Number of scattering angle bins (currently fixed as 20.)
 - I : Energy-angular distribution of scattering
 - σ_p : Photon production cross section of neutron
 - Ip : Energy distribution of secondary photons produced by neutrons

16.5 Printing Output

The following contents are output on the standard output unit (unit 6) during MVP/GMVP calculations. (There are some cases where the output order is different from the following.)

16.5.1 Output on Input Data Processing

- Information about the version of MVP/GMVP used currently.
- Input data list.
- Information about selected options.
- Output about cross section processing.
- Output about the calculation geometry.
- Output of cross-sectional views. (PICTURE option)
- Parameters about the variance reduction techniques. (XIMP, WKIL and WSRV)
- Information about secondary particles (fission spectrum : multigroup, WGTF and WGTP)
- Information about sources. The spatial and energy distributions etc. of each source.

- Information about tallies. The boundary of energy bins, the volume of each region, the response function of reaction rate calculations.
- Results of input data check.

The results of input data check are classified into the next 3 subjects and the number of the subjects is output.

- (1) MESSAGE : It does not affect calculations at all.
- (2) WARNING : It is unlikely to affect calculations but attention should be paid.
- (3) FATAL : Fatal error. It is impossible to execute.

The “WARNING” and “FATAL” messages are output with the first characters of “!!!” and “XXX”, respectively.

16.5.2 Output on Random Walk

Stage during Random Walk

The following data are output in each batch.

- Used CPU time until the beginning of the batch.
- Number of particles (Number of histories).
- Averaged and cumulative values of positions, angles and energy (groups) for generated source particles in the batch.
- Number of cumulative histories.
- In the case of eigenvalue problems, the numbers of fissions, generated fission neutrons and those stored in the fission bank, and the eigenvalue.

If lost particles are found, the following information is output.

- Free flight (Subroutine FLIONE outputs the following message.)

```
(FLIONE) n PARTICLES ARE LOST !!
X=xxx Y=yyy Z=zzz MU=aaa ETA=bbb XI=ccc SURFACE=j NO.=i
      ⋮
      (Repeated n times)
      ⋮
```

- Next zone search (Subroutine SEAONE outputs the following message.)

```
!! (SEAONE) n PARTICLES ARE LOST !! ZONE # =k
```

For each lost particle,

```

i   X=xxx Y=yyy Z=zzz MU=aaa ETA=bbb XI=ccc
LEVL = 1 LZZ = lz1 LPOS = lp1
LEVL = 2 LZZ = lz2 LPOS = lp2
Information in the level where the current particle exists or upper
SUBSPACE name of each level (only if the TALLY-LATTICE option is used.)

```

where

```

lzn : ZONE number of the FRAME where the LATTICE in level n is allocated.
lpn : Index that indicates the SUBFRAME position in the LATTICE in level n.

lpn = i-1 + NVLAT(1) ( (j-1) + NVLAT(2) (k-1) )

```

- Source particle generation (Subroutine SOURCE outputs the following message.)

```

!!! CAUTION: YOU ARE GOING TO GENERATE PARTICLES IN A ZONE
WHERE PARTICLES SHOULD NOT EXIST !!!
SOURCE NO. i MAT = m ZONE = iz      n PARTICLES.
THEY ARE TREATED AS LOST !!!

```

Endless Loop Detection

It is difficult to detect the endless loop of particles but MVP/GMVP adopts the following procedure. In each batch, when the number of survived particles is not more than 1% of source particles or not more than 2, MVP/GMVP enters the detection mode. If the ratio of the number of processed particles in the detection mode to that in the batch is not less than ELOOP (The default value is 0.2.), MVP/GMVP judges that some particles possibly enter the endless loop and outputs the following message.

```

XXX   n  PARTICLES ARE PROBABLY IN ENDLESS LOOP.
(MONITOR 30 EVENTS)

```

Then monitoring messages are output 29 times.

```

EVENT :   k  MZONE   iz  MMAX   n

```

After monitoring 30 events, the following message is output and all survived particles are forcibly killed.

```

XXX n PARTICLES ARE PROBABLY IN ENDLESS LOOP AFTER MONITORING
30 EVENTS.

```

Stage after Random Walk

- If lost particles are found, the number and the ratio (%) of the particles are output.
- The following of the information output to a restart file.
 - Title of input data
 - Number of particles
 - Number of batches
 - Sum of the weight of source particles
 - Random number after random walk finished

16.5.3 Calculation Results

Eigenvalue

Eigenvalues are calculated by the production and neutron balance methods with the track length, collision and analog (MVP only) estimators and the averaged values in each batch, those from the first to each batch and those from each to final batch are output. The eigenvalues estimated by the maximum likelihood method are also output. Finally, the table of the eigenvalues estimated without first NSKIP batches is printed with the covariance matrix.

Reaction Rates with Response Functions

The reaction rates R_{res} defined in Eq. (16.1) are output.

$$R_{\text{res}} = \frac{\int_{V_t} d\mathbf{r} \int_g dE R(E) \Phi(\mathbf{r}, E)}{V_{\text{in}}}, \quad (16.1)$$

where

$R(E)$: Response function (input value),

$\Phi(\mathbf{r}, E)$: Particle flux (n/cm²/unit energy/source particle),

V_t : Volume of specified TALLY REGION,

$\int_g dE$: Integral over g -th energy bin ($= \int_{E_g}^{E_{g+1}} dE$)

V_{in} : Input value of TRVOL.

If $V_{\text{in}} = 1.0$ (default value), the reaction rate integrated over the TALLY REGION is output. If $V_{\text{in}} = V_t$, the reaction rate per unit volume is output.

Particle Flux

If the FLUX-PRINT option is specified, the particle flux per lethargy is output. The particle flux is defined as follows:

$$\Phi_g = \frac{\int_{V_t} d\mathbf{r} \int_g dE \Phi(\mathbf{r}, E)}{V_{\text{in}} \Delta U_g}, \quad (16.2)$$

where $\Phi(\mathbf{r}, E)$ is the particle flux (n/cm²/unit energy/source particle), V_t is the volume of the TALLY REGION, V_{in} is the input value of TRVOL, $\int_g dE$ is the integral over g -th energy bin ($= \int_{E_g}^{E_{g+1}} dE$), and ΔU_g the lethargy width (difference of the natural logarithmic values of energy).

In the output of GMVP adjoint calculations, the (adjoint) flux is not divided by ΔU_g .

Macroscopic Reaction Rates and Effective Cross Sections

The macroscopic reaction rates $R_{x,g}$ and the effective cross sections $\Sigma_{x,g}$ defined in the following equations are output.

$$R_{x,g} = \int_{V_t} d\mathbf{r} \int_g dE \Sigma_x(\mathbf{r}, E) \Phi(\mathbf{r}, E), \quad (16.3)$$

$$\Sigma_{x,g} = \frac{R_{x,g}}{\int_{V_t} d\mathbf{r} \int_g dE \Phi(\mathbf{r}, E)}, \quad (16.4)$$

where Σ_x is the macroscopic cross section (cm⁻¹) of reaction type x .

Microscopic Reaction Rates and Average Cross Sections

The microscopic reaction rates $R_{x,g}^k$ and the average cross sections $\hat{\sigma}_{x,g}^k$ defined in the following equations are output.

$$R_{x,g}^k = \int_{V_t} d\mathbf{r} \int_g dE N^k(\mathbf{r}) \sigma_x^k(\mathbf{r}, E) \Phi(\mathbf{r}, E), \quad (16.5)$$

$$\hat{\sigma}_{x,g}^k = \frac{R_{x,g}^k}{\int_{V_t} d\mathbf{r} \int_g dE N^k(\mathbf{r}) \Phi(\mathbf{r}, E)}. \quad (16.6)$$

$\hat{\sigma}_{x,g}^k$ is the microscopic cross section (barn) of reaction type x and nuclide k . N^k is the atomic density of nuclide k .

It should be noted that the average microscopic cross sections calculated with Eq. (16.6) are different from the effective microscopic cross sections which are conventionally used in reactor core calculations. The effective microscopic cross sections are defined as follows:

$$\sigma_{x,g}^k = \frac{R_{x,g}^k}{\frac{\int_{V_t} d\mathbf{r} N^k(\mathbf{r})}{V_t} \left[\int_{V_t} d\mathbf{r} \int_g dE \Phi(\mathbf{r}, E) \right]} = \frac{R_{x,g}^k}{\left[\int_{V_t} d\mathbf{r} N^k(\mathbf{r}) \right] \frac{V_{in} \Delta U_g}{V_t} \Phi_g} \quad (16.7)$$

If the effective microscopic cross sections are necessary, they can be calculated from Eq. (16.7) by calculating $\int_{V_t} d\mathbf{r} N^k(\mathbf{r}) V_{in} \Delta U_g / V_t$ since $R_{x,g}^k$ and Φ_g have been already calculated.

The results of particle fluxes, reaction rates etc. are normalized to the values per source particle. Thus, the total production rate of fission neutrons is not unity but the eigenvalue in eigenvalue problems. The statistical uncertainty is the estimated value of one standard deviation (%1 σ).

Macroscopic Group-to-Group Scattering Rates and Scattering Matrix Moments

If the SCATTERING-MATRIX option is specified, the macroscopic group-to-group scattering rates ($\ell = 0$) and scattering matrix moments are output. They are defined as follows:

$$R_{x,\ell,g \rightarrow g'} = \sum_k \int_{-1}^1 d\mu_0 \int_g dE \int_{g'} dE' \int_{V_t} d\mathbf{r} N^k(\mathbf{r}) \sigma_x^k(E) P_\ell(\mu_0) f_x^k(E \rightarrow E', \mu_0) \Phi(\mathbf{r}, E), \quad (16.8)$$

where

$R_{x,\ell,g \rightarrow g'}$: group-to-group scattering rates ($\ell = 0$) and scattering matrix moments ($\ell \neq 0$) for scattering type x from energy group g to g' ,

x : scattering type (elastic scattering, inelastic scattering, (n,2n)),

g : index of energy group,

ℓ : order of Legendre expansion,

k : nuclide index,

μ_0 : scattering angle cosine in the laboratory system,

$\int_g dE$: integral over g -th energy bin ($= \int_{E_g}^{E_{g+1}} dE$),

V_t : volume of tally region,

N^k : atomic density of nuclide k ,

σ_x^k : microscopic cross section of nuclide k ,

$P_\ell(\mu_0)$: ℓ -th order Legendre function,
 $f_x^k(E \rightarrow E', \mu_0)$: scattering probability from energy E to E' with the scattering angle cosine μ_0 in the laboratory system,
 Φ : neutron flux.

Currently, the scattering matrix cross sections are not calculated in the code. If they are necessary, they can be calculated from Eq. (16.9) with the results of Eq. (16.8) and the group-averaged particle flux.

$$\Sigma_{x,\ell,g \rightarrow g'} = \frac{R_{x,\ell,g \rightarrow g'}}{\int_{V_t} d\mathbf{r} \int_g dE \Phi(\mathbf{r}, E)} \quad (16.9)$$

The user should pay attention if the higher-order scattering cross sections ($\ell \neq 0$) defined by Eq. (16.9) are used for a multigroup transport calculation with a deterministic code. In the multigroup transport calculation, the neutron flux in the right hand side of Eq. (16.8) must be the flux moment in group condensation or spatial homogenization.

Microscopic Group-to-Group Scattering Rates and Scattering Matrix Moments

If the SCATTERING-MATRIX option is specified, the microscopic group-to-group scattering rates ($\ell = 0$) and scattering matrix moments are output. They are defined as follows:

$$R_{x,\ell,g \rightarrow g'}^k = \int_{-1}^1 d\mu_0 \int_g dE \int_{g'} dE' \int_{V_t} d\mathbf{r} N^k(\mathbf{r}) \sigma_x^k(E) P_\ell(\mu_0) f_x^k(E \rightarrow E', \mu_0) \Phi(\mathbf{r}, E), \quad (16.10)$$

where $R_{x,\ell,g \rightarrow g'}^k$ is the scattering rates ($\ell = 0$) and scattering matrix moments ($\ell \neq 0$) for nuclide k and scattering type x from group g to g' . The meaning of the other symbols is the same as in Eq. (16.8).

Currently, the scattering matrix cross sections are not calculated in the code. If they are necessary, they can be calculated from Eq. (16.11) with the results of Eq. (16.10) and the group-averaged particle flux.

$$\sigma_{x,\ell,g \rightarrow g'}^k = \frac{R_{x,\ell,g \rightarrow g'}^k}{\frac{\int_{V_t} d\mathbf{r} N^k(\mathbf{r})}{V_t} \left[\int_{V_t} d\mathbf{r} \int_g dE \Phi(\mathbf{r}, E) \right]} \quad (16.11)$$

Similarly to the macroscopic case, the user should pay attention if the higher-order scattering cross sections ($\ell \neq 0$) defined by Eq. (16.11) are used for a multigroup transport calculation.

Macroscopic Average Scattering Angle Cosine

If the SCATTERING-MUBAR option is specified and the flags for elastic scattering (N4), inelastic scattering (N6), (n,2n) reaction (N7) of the EDIT-MACROSCOPIC-DATA option are more than or equal to one, the macroscopic average scattering angle cosine $\bar{\mu}_{x,g}$ is output. x is the scattering type (elastic scattering, inelastic scattering, (n,2n) reaction) and g is the energy group.

Microscopic Average Scattering Angle Cosine

If the SCATTERING-MUBAR option is specified and the flags for elastic scattering (N4), inelastic scattering (N6), (n,2n) reaction (N7) of the EDIT-MICROSCOPIC-DATA option are more than or equal to one, the microscopic average scattering angle cosine $\bar{\mu}_{x,g}$ is output. x is the scattering type (elastic scattering, inelastic scattering, (n,2n) reaction), g is the energy group, and k is the nuclide.

Perturbation Calculation

If the PERTURBATION option is specified in the OPTION block and KEFF is specified for the tally parameter in the PERTURBATION block, MVP outputs the change (Δk) in the effective multiplication factor (k) for a perturbed parameter (density or atomic density). If the change is estimated with the differential operator sampling, the differential coefficients ($\partial k / \partial a$, a is the perturbation parameter.) are output for both the cases where the perturbed source effect is or is not considered. The output change (Δk) in the effective multiplication factor is as follows:

$$\Delta k = \frac{\partial k}{\partial a} \Delta a + \frac{1}{2} \frac{\partial^2 k}{\partial a^2} (\Delta a)^2,$$

where Δa is the change in the perturbation parameter specified by the user.

If the change is estimated with the correlated sampling, Δk for each Δa is directly output.

The perturbed source effect is estimated by propagating the weight factor for the effect between multiple generations. The perturbed source effect is output as a function of the number of the propagated generations. This output is used to check the convergence of the estimated value of the effect. The uncertainty in the perturbed source effect, however, tends to become larger with the number of the propagated generations. Thus the user can observe this behavior as a rough indicator.

If the PERTURBATION option is specified in the OPTION block and BEFF is specified for the tally parameter in the PERTURBATION block, the effective delayed neutron fraction and the neutron generation time obtained with the perturbation calculation are output. The effective delayed neutron fraction is calculated by perturbing the number of delayed neutrons and is estimated as follows:

$$\beta_{\text{eff}} = \left[\frac{1}{k} \frac{\partial k}{\partial a} \right]_{a=0},$$

where a is the fractional increment of the number of delayed neutrons. On the other hand, the neutron generation time is calculated by inserting the $1/v$ absorber into the system uniformly and is estimated as follows:

$$\Lambda = \left[-\frac{1}{k^2} \frac{\partial k}{\partial a} \right]_{a=0},$$

where a is the fraction of the $1/v$ absorber.

Effective Delayed Neutron Fraction Calculated with Next Fission Probability Approach

If the BETA-EFFECTIVE option is specified, MVP outputs the effective delayed neutron fraction calculated approximately. The conventional definition of the effective delayed neutron is the fraction of the adjoint-weighted numbers of fission neutrons as follows:

$$\beta_{\text{eff}} = \frac{\sum_i \langle \Phi^\dagger(\mathbf{r}, E, \boldsymbol{\Omega}) S_{f,i}^d(\mathbf{r}, E, \boldsymbol{\Omega}) \rangle}{\langle \Phi^\dagger(\mathbf{r}, E, \boldsymbol{\Omega}) S_f(\mathbf{r}, E, \boldsymbol{\Omega}) \rangle}$$

where $\Phi^\dagger(\mathbf{r}, E, \boldsymbol{\Omega})$ is the adjoint flux, $S_{f,i}^d$ is the number of delayed neutrons, and S_f is the total number of fission neutrons. The bracket denotes $\int d\mathbf{r} \int d\boldsymbol{\Omega} \int dE$.

There are three methods for the effective delayed neutron fraction calculated with the BETA-EFFECTIVE option.

- Constant adjoint flux (non-adjoint weighted):

$$\beta_{\text{eff}} \approx \frac{\sum_i \langle S_{f,i}^d(\mathbf{r}, E, \boldsymbol{\Omega}) \rangle}{\langle S_f(\mathbf{r}, E, \boldsymbol{\Omega}) \rangle}.$$

- Nauchi's method (next fission probability approach):

$$\beta_{\text{eff}} \approx \frac{\sum_i \langle G^\dagger(\mathbf{r}, E, \boldsymbol{\Omega}) S_{f,i}^d(\mathbf{r}, E, \boldsymbol{\Omega}) \rangle}{\langle G^\dagger(\mathbf{r}, E, \boldsymbol{\Omega}) S_f(\mathbf{r}, E, \boldsymbol{\Omega}) \rangle},$$

$$L^\dagger G^\dagger(\mathbf{r}, E, \boldsymbol{\Omega}) = \nu \boldsymbol{\Sigma}_f,$$

where L^\dagger is the adjoint net-loss operator.

- Meulekamp's method (next fission probability approach):

$$\beta_{\text{eff}} \approx \frac{\sum_i \langle G^\dagger(\mathbf{r}, E, \boldsymbol{\Omega}) S_{f,i}^d(\mathbf{r}, E, \boldsymbol{\Omega}) \rangle}{\langle G^\dagger(\mathbf{r}, E, \boldsymbol{\Omega}) S_f(\mathbf{r}, E, \boldsymbol{\Omega}) \rangle},$$

$$L^\dagger G^\dagger(\mathbf{r}, E, \boldsymbol{\Omega}) = \boldsymbol{\Sigma}_f.$$

For each method, calculated values with the track length, collision and analog estimators are output for each batch, the average to each batch and the average after each batch. In addition, the combined averaged values with the maximum likelihood estimation are also output. The table for the final results discarding NSKIP batchs is output with the correlation matrix.

16.5.4 Event Monitor

The number of occurrence and the sum of weights are output for the following events.

- Particle generation from neutron sources
- Fission reaction
- Photon production reaction
- All collision reaction
- Splitting
- Particles that cannot be split due to the shortage of the particle bank
- Leakage from the system
- Particles killed by energy cutoff (because the particles reach the lowest boundary of energy.)
- Survived and killed particles in the case of the IMPORTANCE or WEIGHT-WINDOW option.
- Survived and killed particles in the case of the RUSSIAN-ROULETTE option. (weight cutoff)

In addition, the number of occurrence for the following events.

- Free flight of particles
- Passing through ZONE boundaries
- Reflection by reflective surfaces

When the MONITOR option is used, the number and the weight of collisions in each energy group and each REGION (not TALLY REGION), and the number of events of scattering below the lowest energy boundary are output.

16.6 Binary Output of Calculation Results

After the calculation of MVP/GMVP finishes, calculated results are output in the binary form on I/O unit 30. The post-process such as plotting the calculated results etc. can be performed by using this file.

- The basic form is the same for both MVP and GMVP. In the fourth record or later, each record has the “TAG” that consists of 32 characters and indicates the content of the record.
- The contents of the file may be changed by improvement of the codes and implementation of new functions in the future. The version of the file is output in the first record. The current version for MVP and GMVP is 3.08 and 3.04, respectively (as of March in 2016).
- The order of the items described below is not determined in the file but records labeled with TAGs in the items are ordered as described here. In addition, some items may not be output depending on the used options. For example, item (9) is not always output because it is the result by the point detector estimator. Therefore, the content of the TAG should be always checked when the user writes a program to read the file. (See Appendix D for sample programs.)
- The file is output in the binary form of FORTRAN and thus can not be used on different platforms in most cases.
- The data type of the contents in the output records without special description is single-precision floating-point or integer in most cases and the variables obey the FORTRAN rule of implicit definition*.

The content of each record is described as follows. Without any comments, an item such as (1), (2) etc. corresponds to one record.

- (1) 64-character string for the form of the output file. (It may change with the version of codes.) Currently, the following string is output.

- MVP code : 'MVP OUTPUT FILE TYPE 3.08 (FROM 11/AUG/2006) '
- GMVP code : 'GMVP OUTPUT FILE TYPE 3.04 (FROM 27/OCT/1998) '

- (2) 31-character string for the date and time of the calculation. The form is as follows.

'DATE dd-mmm-yyyy TIME hr:mn:sc'

*The variables that begin with A through H or O through Z are single-precision floating-point data and those with I through N are integer data.

The calculation date is “dd”-th of “mmm” month in “yyyy” year and the time is “hr” hour “mn” minute “sc” second.

(3) Title of the input data (72 characters \times 2).

(4) Parameters of random walk.

```
write(30) TAG,
WSUM,NTHIST,NBATCH,NGROUP,NREG,NRESP,NSKIP,NSTAL,NGP1,NGP2,
NTREG,NPDET,NTIME,NSTALY,JNEUT,JPHOT,JRESP,JEIGN,JADJM,JTIME,
IQNTH, IRNTH
```

- TAG = ' PROBLEM PARAMETERS ' (32 characters)
- WSUM : Sum of particle weights (double precision)
- NTHIST : Total number of particles
- NBATCH : Total number of batches
- NGROUP : Number of energy groups
- NREG : Number of REGIONs
- NRESP : Number of multigroup response functions
- NSKIP : Number of skipped batches excluded from statistics in eigenvalue problems
- NSTAL : Number of pointwise response functions (NSTAL=0 for GMVP)
- NGP1 : Number of neutron energy bins (groups)
- NGP2 : Number of photon energy bins (groups)
- NTREG : Number of TALLY REGIONs
- NPDET : Number of point detectors
- NTIME : Number of time bins
- NSTALY : Number of tallies defined in the TALLY data block
- JNEUT : 1/0=track neutrons / not track neutrons
- JPHOT : 1/0=track photons / not track photons
- JRESP : 1/0=reaction rate calculations / no reaction rate calculations
- JEIGN : 1/0=eigenvalue problem / fixed-source problem
- JADJM : 1/0=adjoint calculation / forward calculation (JADJM=0 for MVP)
- JTIME : 1/0=time-dependent calculation / steady-state calculation
- IQNTH : INT(NHIST/I4MAX), where I4MAX is the largest integer expressed by a single-precision integer (MVP only)
- IRNTH : NHIST-IQNTH*I4MAX (MVP only)

(5) Energy boundaries, region volumes, time boundaries

```
write(30) TAG,
(ENGYB(I),I=1,NGP1+1),(ENGPB(I),I=1,NGP2+1),
(TRVOL(I),I=1,NTREG),(RVOL(I),I=1,NREG),
(TIMEB(I),I=1,NTIME+1)
```

- TAG = ' ENERGY & REG.VOLUME ' (32 characters)
- ENGYB : Energy bin boundaries for neutrons (eV)
- ENGPB : Energy bin boundaries for photons (eV)
- TRVOL : Volume of TALLY REGIONs (cm³)
- RVOL : Volume of REGIONs (cm³)
- TIMEB : Time bin boundaries (s)

(6) TALLY REGION names

```
do KR=1,NTREG
  write(30) TAG, KR, LNM, NAME(1:LNM)
end do
```

- TAG = ' TALLY REGION NAME ' (32 characters)
- KR : TALLY REGION number
- LNM : Length of TALLY REGION name
- NAME(1:LNM) : TALLY REGION name (LNM characters)

(7) Particle fluxes and their relative uncertainties (% 1σ) in each TALLY REGION and energy group by the track length estimator

```
do KR=1,NTREG
  write(30) TAG,
    (SFLTR(IG,KR,1),IG=1,NGROUP),(SFLTR(IG,KR,2),IG=1,NGROUP)
end do
```

- TAG = ' FLUX & ERROR: TRACK LENGTH ' (32 characters)
- SFLTR(IG,KR,1) : Particle fluxes per unit lethargy, unit volume and a source particle by the track length estimator or adjoint fluxes per a source particle for GMVP adjoint calculations (double precision).
- SFLTR(IG,KR,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

(8) Particle fluxes and their relative uncertainties (% 1σ) in each TALLY REGION and energy group by the collision estimator

```
do KR=1,NTREG
  write(30) TAG,
    (SFLCL(IG,KR,1),IG=1,NGROUP),(SFLCL(IG,KR,2),IG=1,NGROUP)
end do
```

- TAG = ' FLUX & ERROR: COLLISION ' (32 characters)
- SFLCL(IG,KR,1) : Particle fluxes per unit lethargy, unit volume and a source particle by the collision estimator or adjoint fluxes per a source particle for GMVP adjoint calculations (double precision).
- SFLCL(IG,KR,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

- (9) Particle fluxes and their relative uncertainties (1σ) in each energy group by the point detector estimator

```
do KP=1,NPDET
  write(30) TAG,
    (SFLPD(IG,KP,1),IG=1,NGROUP),(SFLPD(IG,KP,2),IG=1,NGROUP)
end do
```

These are output only if NPDET>0 for GMVP.

- TAG = ' FLUX & ERROR: POINT DETECTOR ' (32 characters)
- SFLPD(IG,KP,1) : Particle fluxes per unit lethargy and a source particle by the point detector estimator or adjoint fluxes per a source particle for GMVP adjoint calculations (double precision)
- SFLPD(IG,KP,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

- (10) Reaction rates and their relative uncertainties (1σ) by the track length estimator with multigroup response functions

```
do N=1,NRESP
  write(30) TAG,
    (SRETR(K,N,1),K=1,NTREG),(SRETR(K,N,2),K=1,NTREG)
end do
```

- TAG = ' REACTION RATE (RESP) : TRACK LENGTH ' (32 characters)
- SRETR(K,N,1) : Reaction rates per unit volume and a source particle by the track length estimator (double precision).
- SRETR(K,N,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

- (11) Reaction rates and their relative uncertainties (1σ) by the collision estimator with multigroup response functions

```
do N=1,NRESP
  write(30) TAG,
    (SRECL(K,N,1),K=1,NTREG),(SRECL(K,N,2),K=1,NTREG)
end do
```

- TAG = ' REACTION RATE (RESP) : COLLISION ' (32 characters)
- SRECL(K,N,1) : Reaction rates per unit volume and a source particle by the collision estimator (double precision).
- SRECL(K,N,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

- (12) Reaction rates and their relative uncertainties (1σ) by the track length estimator with pointwise response functions

```
do N=1,NSTAL
  write(30) TAG,
    (SRSTR(K,N,1),K=1,NTREG),(SRSTR(K,N,2),K=1,NTREG)
end do
```

These are output only when JRESP=1, NSTAL>0.

- TAG = 'REACTION RATE (STAL) : TRACK LENGTH' (32 characters)
- SRSTR(K,N,1) : Reaction rates per unit volume and a source particle by the track length estimator (double precision).
- SRSTR(K,N,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

- (13) Reaction rates and their relative uncertainties (1σ) by the collision estimator with pointwise response functions

```
do N=1,NSTAL
  write(30) TAG,
    (SRSC(L,K,N,1),K=1,NTREG),(SRSC(L,K,N,2),K=1,NTREG)
end do
```

These are output only when JRESP=1, NSTAL>0.

- TAG = 'REACTION RATE (STAL) : COLLISION ' (32 characters)
- SRSC(L,K,N,1) : Reaction rates per unit volume and a source particle by the collision estimator (double precision).
- SRSC(L,K,N,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

- (14) Reaction rates and their relative uncertainties (1σ) by the point detector estimator with multi-group response functions

These are output only if NPDET>0 for GMVP.

```
do N=1,NRESP
  write(30) TAG,
    (SREPD(K,N,1),K=1,NPDET),(SREPD(K,N,2),K=1,NPDET)
end do
```

- TAG = 'REACTION RATE (RESP) : POINT '
- SREPD(K,N,1) : Reaction rates per a source particle by the point estimator (double precision).
- SREPD(K,N,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

- (15) Information about tallies of cross sections, macroscopic and microscopic reaction rates in MVP

```
write(30) TAG,
  NUC,NEMIC,NEMAC,JRTTR,JRTCL,(JMICE(L),L=1,8),(JMACE(L),L=1,8),
  KY,KYM,JSCTM,JSCMU,NPKIND
```

- TAG = ' MICRO & MACRO TALLY DATA ' (32 characters)
- NUC : Number of nuclides
- NEMIC : Number of reaction types of which microscopic cross sections and reaction rates are calculated.
- NEMAC : Number of reaction types of which macroscopic cross sections and reaction rates are calculated.
- JRTTR : 1/0 = Track length estimator is used / Collision estimator is used.
- JRTCL : 1/0 = Collision estimator is used / Track length estimator is used.
- JMICE(L) : 1/0 = Microscopic parameters are calculated for the L-th type reaction. / No microscopic parameters are calculated for the L-th type reaction. See the EDIT-MICRO-DATA option about L=1 ~ 8.
- JMACE(L) : 1/0 = Macroscopic parameters are calculated for the L-th type reaction. / No macroscopic parameters are calculated for the L-th type reaction. See the EDIT-MACRO-DATA option about L=1 ~ 8.
- KY : Number of reactions for which microscopic scattering matrices are calculated.
- KYM : Number of reactions for which macroscopic scattering matrices are calculated.
- JSCTM : $n/0$ = Group-to-group scattering rates and scattering matrix moments of $(n - 1)$ -th order are calculated / No scattering matrices are calculated.
- JSCMU : 1/0 = Average scattering cosines are calculated / No average scattering cosines are calculated.
- NPKIND : Number of particle types. NPKIND is 2 if JNEUT=JPHOT=1, and NPKIND is 1 if either of JNEUT and JPHOT is 1.

(16) Additional information about tallies of cross sections, macroscopic and microscopic reaction rates in MVP

write(30) TAG, (LEMIC(L),L=1,16), (LEMAC(L),L=1,16), (NUCID(N),N=1,NUC)

- TAG = ' MICRO & MACRO TALLY DATA 2 ' (32 characters)
- LEMIC : Indices of positions in SRMIC etc. of each reaction type.
 - LEMIC(L),L=1,8 : The L-th type reaction is calculated as the LEMIC(L)-th reaction. That is, LEMIC(L) is the value of M in SRMIC,RMICR,and XMIC. (M is explained below.) Unless the L-th type reaction is calculated, LEMIC(L)=0. ($0 \leq \text{LEMIC}(L) \leq \text{NEMIC}$)
 - LEMIC(M+8) : Reaction type for the tally of the M-th reaction. ($M \leq \text{NEMIC}$), $1 \leq \text{LEMIC}(M+8) \leq 8$
- LEMAC : Similar information to LEMIC for macroscopic parameters.
- NUCID : ID names of nuclides (16 characters).

(17) Nuclide ID

write(30) TAG, len(NUCID(1)), (NUCID(N),N=1,NUC)

- TAG = ' NUCLIDE ID ' (32 characters)

- NUCID: Nuclide ID (16 characters)
- “len” is a FORTRAN intrinsic function. It returns the length of a character string.

(18) Nuclide temperature

```
write(30) TAG, (TEMPN(N),N=1,NUC)
```

- TAG = ' NUCLIDE TEMPERATURE ' (32 characters)
- TEMPN: Nuclide temperature (K)

(19) Microscopic reaction rates and their relative uncertainties in each energy group and each TALLY REGION.

```
write(30) TAG,
(((SRMIC(IG,K,N,M,1),IG=1,NGROUP),K=1,NTREG),N=1,NUC),M=1,NEMIC),
(((SRMIC(IG,K,N,M,2),IG=1,NGROUP),K=1,NTREG),N=1,NUC),M=1,NEMIC)
```

- TAG = ' MICROSCOPIC REACTION RATE ' (32 characters)
- SRMIC(IG,K,N,M,1): Reaction rates per a source particle (double precision).
- SRMIC(IG,K,N,M,2): Relative uncertainties (double precision). The standard deviation is given in the % unit.

(20) Energy-integrated microscopic reaction rates and their relative uncertainties in each TALLY REGION.

```
write(30) TAG,
( ((RMICR(J,K,N,M,1),K=1,NTREG),N=1,NUC),M=1,NEMIC),
((RMICR(J,K,N,M,2),K=1,NTREG),N=1,NUC),M=1,NEMIC), J=1,NPKIND )
```

where NPKIND is 2 if JNEUT=JPHOT=1, and NPKIND is 1 if either of JNEUT and JPHOT is 1. The index J is the type of particles (neutron/photon). J=1 indicates neutrons and J=2 photons if JNEUT=JPHOT=1.

- TAG = ' MICROSCOPIC REACTION RATE 2 ' (32 characters)
- RMICR(J,K,N,M,1) : Reaction rates per a source particle (double precision).
- RMICR(J,K,N,M,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

(21) Effective microscopic cross sections and their relative uncertainties in each energy group and each TALLY REGION.

```
write(30) TAG,
(((XMIC(IG,K,N,M,1),
IG=1,NGROUP+NPKIND),K=1,NTREG),N=1,NUC),M=1,NEMIC),
(((XMIC(IG,K,N,M,2),
IG=1,NGROUP+NPKIND),K=1,NTREG),N=1,NUC),M=1,NEMIC)
```


where NPKIND is 2 if JNEUT=JPHOT=1, and NPKIND is 1 if either of JNEUT and JPHOT is 1. IG=NGROUP+1 indicates the averaged cross sections over whole energy range of neutrons or photons depending on JNEUT and JPHOT. If NPKIND=2, IG=IGROUP+2 corresponds to photons.

- TAG = ' MICROSCOPIC CROSS SECTION ' (32 characters)
- XMIC(IG,K,N,M,1) : Cross sections (double precision).
- XMIC(IG,K,N,M,1) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

(22) Macroscopic reaction rates and their relative uncertainties in each energy group and each TALLY REGION.

```
write(30) TAG,
(((RMAC(IG,K,M,1),IG=1,NGROUP),K=1,NTREG),M=1,NEMAC),
(((RMAC(IG,K,M,2),IG=1,NGROUP),K=1,NTREG),M=1,NEMAC)
```

- TAG = ' MACROSCOPIC REACTION RATE ' (32 characters)
- RMAC(IG,K,M,1) : Reaction rates per a source particle (double precision).
- RMAC(IG,K,M,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

(23) Energy-integrated macroscopic reaction rates and their relative uncertainties in each TALLY REGION.

```
write(30) TAG,
( ((RMACR(J,K,M,1),K=1,NTREG),M=1,NEMAC),
((RMACR(J,K,M,2),K=1,NTREG),M=1,NEMAC),J=1,NPKIND )
```

The index J is the type of particles (neutron/photon). J=1 indicates neutrons and J=2 photons if JNEUT=JPHOT=1. J=1 indicates photons if JNEUT=0 and JPHOT=1 (photon-only calculation).

- TAG = ' MACROSCOPIC REACTION RATE 2 ' (32 characters)
- RMACR(J,K,M,1): Reaction rates per a source particle (double precision).
- RMACR(J,K,M,2): Relative uncertainties (double precision). The standard deviation is given in the % unit.

(24) Effective macroscopic cross sections and their relative uncertainties in each energy group and each TALLY REGION.

```
write(30) TAG,
(((XMAC(IG,K,M,1),IG=1,NGROUP+NPKIND),K=1,NTREG),M=1,NEMAC),
(((XMAC(IG,K,M,2),IG=1,NGROUP+NPKIND),K=1,NTREG),M=1,NEMAC)
```

where NPKIND is 2 if JNEUT=JPHOT=1, and NPKIND is 1 if either of JNEUT and JPHOT is 1. IG=NGROUP+1 indicates the averaged cross sections over whole energy range of neutrons or photons depending on JNEUT and JPHOT. If NPKIND=2, IG=IGROUP+2 corresponds to photons.

- TAG = 'MACROSCOPIC CROSS SECTION' (32 characters)
- XMAC(IG,K,M,1) : Cross sections (double precision).
- XMAC(IG,K,M,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

(25) Effective microscopic elastic scattering matrix moments and their relative uncertainties in each energy group to group, each TALLY REGION and each nuclide.

```
write(30) TAG,
((((SRMICS(IG1,IG2,K,N,1,IM),IG1=1,NGP1),IG2=1,NGP1),K=1,NTREG),
N=1,NUC),IM=1,JSCTM),
((((SRMICS(IG1,IG2,K,N,2,IM),IG1=1,NGP1),IG2=1,NGP1),K=1,NTREG),
N=1,NUC),IM=1,JSCTM)
```

- TAG = 'MICRO ELAS REAC SCAT MATRIX' (32 characters)
- SRMICS(IG1,IG2,K,N,1,IM) : Scattering matrix moments (double precision).
- SRMICS(IG1,IG2,K,N,2,IM) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

(26) Effective microscopic inelastic scattering matrix moments and their relative uncertainties in each energy group to group, each TALLY REGION and each nuclide.

```
write(30) TAG,
((((SRMICS(IG1,IG2,K,N,1,IM),IG1=1,NGP1),IG2=1,NGP1),K=1,NTREG),
N=1,NUC),IM=1,JSCTM),
((((SRMICS(IG1,IG2,K,N,2,IM),IG1=1,NGP1),IG2=1,NGP1),K=1,NTREG),
N=1,NUC),IM=1,JSCTM)
```

- TAG = 'MICRO INEL REAC SCAT MATRIX' (32 characters)
- SRMICS(IG1,IG2,K,N,1,IM) : Scattering matrix moments (double precision).
- SRMICS(IG1,IG2,K,N,2,IM) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

(27) Effective microscopic (n,2n) reaction scattering matrix moments and their relative uncertainties in each energy group to group, each TALLY REGION and each nuclide.

```
write(30) TAG,
((((SRMICS(IG1,IG2,K,N,1,IM),IG1=1,NGP1),IG2=1,NGP1),K=1,NTREG),
N=1,NUC),IM=1,JSCTM),
((((SRMICS(IG1,IG2,K,N,2,IM),IG1=1,NGP1),IG2=1,NGP1),K=1,NTREG),
N=1,NUC),IM=1,JSCTM)
```

- TAG = 'MICRO N2N REAC SCAT MATRIX' (32 characters)
- SRMICS(IG1,IG2,K,N,1,IM) : Scattering matrix moments (double precision).
- SRMICS(IG1,IG2,K,N,2,IM) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

- (28) Effective macroscopic elastic scattering matrix moments and their relative uncertainties in each energy group to group and each TALLY REGION.

```
write(30) TAG,
((((RMACSM(IG1,IG2,K,N,1,IM),IG1=1,NGP1),IG2=1,NGP1),K=1,NTREG),
IM=1,JSCTM),
((((RMACSM(IG1,IG2,K,N,2,IM),IG1=1,NGP1),IG2=1,NGP1),K=1,NTREG),
IM=1,JSCTM)
```

- TAG = 'MACRO ELAS REAC SCAT MATRIX' (32 characters)
- RMACSM(IG1,IG2,K,N,1,IM) : Scattering matrix moments (double precision).
- RMACSM(IG1,IG2,K,N,2,IM) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

- (29) Effective macroscopic inelastic scattering matrix moments and their relative uncertainties in each energy group to group and each TALLY REGION.

```
write(30) TAG,
((((RMACSM(IG1,IG2,K,N,1,IM),IG1=1,NGP1),IG2=1,NGP1),K=1,NTREG),
IM=1,JSCTM),
((((RMACSM(IG1,IG2,K,N,2,IM),IG1=1,NGP1),IG2=1,NGP1),K=1,NTREG),
IM=1,JSCTM)
```

- TAG = 'MACRO INEL REAC SCAT MATRIX' (32 characters)
- RMACSM(IG1,IG2,K,N,1,IM) : Scattering matrix moments (double precision).
- RMACSM(IG1,IG2,K,N,2,IM) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

- (30) Effective macroscopic (n,2n) reaction scattering matrix moments and their relative uncertainties in each energy group to group and each TALLY REGION.

```
write(30) TAG,
((((RMACSM(IG1,IG2,K,N,1,IM),IG1=1,NGP1),IG2=1,NGP1),K=1,NTREG),
IM=1,JSCTM),
((((RMACSM(IG1,IG2,K,N,2,IM),IG1=1,NGP1),IG2=1,NGP1),K=1,NTREG),
IM=1,JSCTM)
```

- TAG = 'MACRO N2N REAC SCAT MATRIX' (32 characters)
- RMACSM(IG1,IG2,K,N,1,IM) : Scattering matrix moments (double precision).
- RMACSM(IG1,IG2,K,N,2,IM) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

- (31) Microscopic average elastic scattering cosines and their relative uncertainties in each energy group to group, each TALLY REGION and each nuclide.

```
write(30) TAG,
((((SMICMU(IG1,IG2,K,N,1),IG1=1,NGP1),IG2=1,NGP1+1),K=1,NTREG),N=1,NUC),
((((SMICMU(IG1,IG2,K,N,2),IG1=1,NGP1),IG2=1,NGP1+1),K=1,NTREG),N=1,NUC)
```

- TAG = 'MICRO ELAS MUBA SCAT' (32 characters)
- SMICMU(IG1,IG2,K,N,1) : Average scattering cosines (double precision). IG2=NGP1+1 corresponds to summation over secondary energy groups.
- SMICMU(IG1,IG2,K,N,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

(32) Microscopic average inelastic scattering cosines and their relative uncertainties in each energy group to group, each TALLY REGION and each nuclide.

```
write(30) TAG,
(((SMICMU(IG1,IG2,K,N,1),IG1=1,NGP1),IG2=1,NGP1+1),K=1,NTREG),N=1,NUC),
(((SMICMU(IG1,IG2,K,N,2),IG1=1,NGP1),IG2=1,NGP1+1),K=1,NTREG),N=1,NUC)
```

- TAG = 'MICRO INEL MUBA SCAT' (32 characters)
- SMICMU(IG1,IG2,K,N,1) : Average scattering cosines (double precision). IG2=NGP1+1 corresponds to summation over secondary energy groups.
- SMICMU(IG1,IG2,K,N,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

(33) Microscopic average (n,2n) reaction scattering cosines and their relative uncertainties in each energy group to group, each TALLY REGION and each nuclide.

```
write(30) TAG,
(((SMICMU(IG1,IG2,K,N,1),IG1=1,NGP1),IG2=1,NGP1+1),K=1,NTREG),N=1,NUC),
(((SMICMU(IG1,IG2,K,N,2),IG1=1,NGP1),IG2=1,NGP1+1),K=1,NTREG),N=1,NUC)
```

- TAG = 'MICRO N2N MUBA SCAT' (32 characters)
- SMICMU(IG1,IG2,K,N,1) : Average scattering cosines (double precision). IG2=NGP1+1 corresponds to summation over secondary energy groups.
- SMICMU(IG1,IG2,K,N,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

(34) Macroscopic average elastic scattering cosines and their relative uncertainties in each energy group to group and each TALLY REGION.

```
write(30) TAG,
(((RMAMU(IG1,IG2,K,N,1),IG1=1,NGP1),IG2=1,NGP1+1),K=1,NTREG),
(((RMAMU(IG1,IG2,K,N,2),IG1=1,NGP1),IG2=1,NGP1+1),K=1,NTREG)
```

- TAG = 'MACRO ELAS MUBA SCAT' (32 characters)
- RMAMU(IG1,IG2,K,N,1) : Average scattering cosines (double precision). IG2=NGP1+1 corresponds to summation over secondary energy groups.
- RMAMU(IG1,IG2,K,N,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

(35) Macroscopic average inelastic scattering cosines and their relative uncertainties in each energy group to group and each TALLY REGION.

```
write(30) TAG,
(((RMAMU(IG1,IG2,K,N,1),IG1=1,NGP1),IG2=1,NGP1+1),K=1,NTREG),
(((RMAMU(IG1,IG2,K,N,2),IG1=1,NGP1),IG2=1,NGP1+1),K=1,NTREG)
```

- TAG = 'MACRO INEL MUBA SCAT' (32 characters)
- RMAMU(IG1,IG2,K,N,1) : Average scattering cosines (double precision). IG2=NGP1+1 corresponds to summation over secondary energy groups.
- RMAMU(IG1,IG2,K,N,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

(36) Macroscopic average (n,2n) reaction scattering cosines and their relative uncertainties in each energy group to group and each TALLY REGION.

```
write(30) TAG,
(((RMAMU(IG1,IG2,K,N,1),IG1=1,NGP1),IG2=1,NGP1+1),K=1,NTREG),
(((RMAMU(IG1,IG2,K,N,2),IG1=1,NGP1),IG2=1,NGP1+1),K=1,NTREG)
```

- TAG = 'MACRO N2N MUBA SCAT' (32 characters)
- RMAMU(IG1,IG2,K,N,1) : Average scattering cosines (double precision). IG2=NGP1+1 corresponds to summation over secondary energy groups.
- RMAMU(IG1,IG2,K,N,2) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

(37) k_{eff} in each batch

```
write(30) TAG,
((XKB(J,I),I=1,NBATCH),J=1,NES),((XKC(J,I),I=1,NBATCH),J=1,NES)
```

- TAG = 'K-EFF: BATCH & CUMULATIVE' (32 characters)
- XKB(J,I) : Evaluated k_{eff} values in each batch (double precision). J indicates an evaluation method. NES=6(MVP), NES=4 (GMVP).
- XKC(J,I) : Averaged k_{eff} values from the first through I-th batches by each evaluation method. (double precision)

(38) k_{eff} by each estimation method

```
do J=1,NES
  write(30) TAG,
    (XKEF(J,I),I=1,NBATCH-1),(VKEF(J,I),I=1,NBATCH-1)
end do
```

- TAG = 'K-EFF: AFTER EACH BATCH' (32 characters)
- XKEF(J,I) : k_{eff} values evaluated from the I-th batch and later ($I \sim \text{NBATCH}$) by the J-th evaluation method. NES=6(MVP), NES=4 (GMVP)
- VKEF(J,I) : Relative uncertainties (double precision). The standard deviation is given in the % unit.

(39) Estimated k_{eff} by the method of maximum likelihood

```

do J=1,N
  write(30) TAG,
    (XKMLE(J,I),I=1,NBATCH-1),(XKERR(J,I),I=1,NBATCH-1)
end do

```

- TAG = 'K-EFF: MAXIMUM LIKELYHOOD' (32 characters)
- XKMLE(J,I) : Eigenvalues estimated by the maximum likelihood method (double precision). Averaged values over the I-th through NBATCH-th batches.
- XKERR(J,I) : Relative uncertainties (double precision).
J indicates the combination of the evaluation methods (N=6 for MVP, N=5 for GMVP).
MVP:
1 : track length estimators (multiplication + balance)
2 : collision estimators (multiplication + balance)
3 : analog estimators (multiplication + balance)
4 : multiplication (track length + collision + analog)
5 : neutron balance (track length + collision + analog)
6 : all estimators
GMVP:
1 : track length estimators (multiplication + balance)
2 : collision estimators (multiplication + balance)
3 : multiplication (track length + collision)
4 : neutron balance (track length + collision)
5 : all estimators

(40) Tally information specified in the TALLY data block

```

write(30) TAGS, NETALY, NEDTMX, NBINMX, NETRV

```

The results of “special tallies” specified in the TALLY data block are output in this record and the following (41)–(48).

- TAGS = 'TALLIES SPECIAL' (32 characters).
- NETALY : Number of special tallies.
- NEDTMX : Number of data of the special tally that has the maximum number of data in all special tallies.
- NBINMX : The results of special tallies are stored as multi-dimensional array data. NBINMX is the largest size of the dimension in all special tallies.
- NETRV: Number of special tallies whose “real variance” is evaluated.

Figure 16.1 shows the records output as special tallies including this record and the following records from (41) to (47).

```

TAGS      = 'TALLIES SPECIAL'
write(30) TAGS, NETALY, NEDTMX, NBINMX, NETRV

+----- repeat for tally #1 to NETALY -----+
|
| TAG = 'TALLY.'//<tally-#>
|   write(30) TAG, tally-#, number-of-dimension, JRV
|   JRV: flag to show real variance(error) estimation is available
|         ( 0 / 1 = no/yes )
|
| TAG = 'TALLY-LABEL'      (optional)
|   write(30) TAG, length-of-label-string, tally-label
|
| TAG = 'TALLY-HEADER'
|   write(30) TAG, header-array-length, header-data
|
+----- repeat for dimension -----+
|
| TAG = 'TALLY-DIMENSION'
|   write(30) TAG, dimension-type(16 char), dimension-size
|   <Dimension type>
|           REGION ENERGY TIME ANGLE GENERATION
|
| TAG = 'TALLY-BIN' (if dimension type is not region)
|   write(30) TAG, ( lower/upper bin combinations )
|
| TAG = 'TALLY-REGION' (if dimension type is region)
|   write(30) TAG, string-length-of-region-name,(region name list)
|
+-----+
|
| TAG = 'TALLY-DATA'
|   write(30) TAG, size, tally-data-array(..., dim3, dim2, dim1)
|
| TAG = 'TALLY-ERROR'
|   write(30) TAG, size, tally-error-array(..., dim3, dim2, dim1)
|
| TAG = 'TALLY-ERROR-REAL'
|   ('`real error(variance)``' estimation, if JRV.ne.0)
|   write(30) TAG, size, tally-error-array(..., dim3, dim2, dim1)
|
+-----+

```

Fig. 16.1 Output of special tallies in the file of calculated results

(41) Number of dimensions etc. for each special tally

```
write(30) TAG, N, NDIM, JRV
```

- TAG = ' TALLY.n ' (32 characters)
(n is the decimal integer of a tally number.)
- N : Special tally number.
- NDIM : This special tally has NDIM dimensions.
- JRV : Flag of the evaluation of the “real variance” (1/0 = yes/no).

(42) Labels of each special tally

write(30) TAG, L, TLABEL(1:L)

- TAG = 'TALLY-LABEL' (32 characters)
- L : Length of a character string (bytes, not more than 128).
- TLABEL(1:L): Character string of a label.

(43) Information about the structure of each special tally

write(30) TAG, NN, (INFO(J), J=1, NN)

- TAG = ' TALLY-HEADER' (32 characters)
- NN : Number of data of an INFO array (=10).
- INFO(J) :
 - INFO(1): Special tally ID
 - INFO(2): Particle type
 - INFO(3): Method to synthesize direct-tallies
 - 0:** No combination
 - 1:** Weighted sum
 - 2:** Multiplication / division
 - 3:** General expression
 - INFO(4): Number of tally dimensions (NDIM)
 - INFO(5): Starting address of this tally in ETALY(*)
 - INFO(6): Event #
 - INFO(7): Weighting function
 - INFO(8): Nuclide/atom # & reaction #
 - INFO(9): <unused>
 - INFO(10): Custom tally# (non-zero means user customized tally)

(44) Type of all dimensions and tally bins for each special tally

do I=1,NDIM

 TAG = 'TALLY-DIMENSION' (32 characters)

 write(30) TAG, dimension-type(16 char), dimension-size

 TAG = 'TALLY-BIN' (32 characters)

 (if dimension type is not REGION)

 write(30) TAG, (lower/upper bin combinations)

 TAG = 'TALLY-REGION' (32 characters)

 (if dimension type is REGION)

 write(30) TAG, string-length-of-REGION-name,
 (REGION name list)

end do

- TAG = 'TALLY-DIMENSION'
 - < dimension type >: The type of bins for tallies. One of the following character strings.

REGION, POINT, ENERGY, TIME, ANGLE, GENERATION, SOURCE, SOURCE-REGION, MARKER-REGION, PRODUCE-REGION, PRODUCE-NUCLIDE, PRODUCE-REACTION

- * REGION : Tally region.
- * POINT : Position for the point detector.
- * ENERGY : Energy.
- * TIME : Time.
- * ANGLE : Angle. The angle or the cosine of the angle between the normal of the detector surface and the flight direction of particles for the surface crossing estimator.
- * GENERATION : Generation of particles in the fixed-source problem.
- * SOURCE : Source set ID from which particles are generated.
- * SOURCE-REGION : REGION of the source from which particles are generated.
- * MARKER-REGION : “marker REGION”. Parameters are tallied for the particles that pass through the specified REGION.
- * PRODUCE-REGION : Spatial REGION where a secondary particle is produced from collisions except for elastic scattering.
- * PRODUCE-NUCLIDE : Target nuclide with which a primary particle is collided and a secondary particle is produced from collisions except for elastic scattering.
- * PRODUCE-REACTION : Collision reaction (MT number) from a secondary particle is produced except for elastic scattering.

– $\langle \text{dimension-size} \rangle$: Number of bins for this dimension.

- TAG = 'TALLY-BIN'

The upper and lower boundary values of bins when $\langle \text{dimension type} \rangle$ is not “REGION”.

- TAG = 'TALLY-REGION'

The character string of TALLY REGION names and a name list when $\langle \text{dimension type} \rangle$ is “REGION”.

(45) Calculated results of each special tally

write(30) TAG, $\langle \text{size} \rangle$, $\langle \text{tally-data-array}(\dots, \text{dim3}, \text{dim2}, \text{dim1}) \rangle$

- TAG = 'TALLY-DATA' (32 characters)
- $\langle \text{size} \rangle$: Size of data (product of the numbers of bins that compose this tally).
- $\langle \text{tally-data-array}(\dots, \text{dim3}, \text{dim2}, \text{dim1}) \rangle$: Calculated results (double precision). These are NDIM-dimensional FORTRAN-like array data.

(46) Standard deviations of each special tally

write(30) TAG, $\langle \text{size} \rangle$, $\langle \text{tally-error-array}(\dots, \text{dim3}, \text{dim2}, \text{dim1}) \rangle$

- TAG = 'TALLY-ERROR' (32 characters)
- $\langle \text{size} \rangle$: Size of data (product of the numbers of bins that compose this tally).
- $\langle \text{tally-error-array}(\dots, \text{dim3}, \text{dim2}, \text{dim1}) \rangle$: Relative uncertainties in the % unit (double precision). These are NDIM-dimensional FORTRAN-like array data.

(47) Standard deviations evaluated as the real variance of each special tally

These are output only when $JRV > 0$.

```
write(30) TAG, < size >, < tally-error-array(..., dim3, dim2, dim1)>
```

- TAG = ' TALLY-ERROR-REAL ' (32 characters)
- < size > : Size of data (product of the numbers of bins that compose this tally).
- < tally-error-array(..., dim3, dim2, dim1) > : Relative uncertainties in the % unit (double precision). These are NDIM-dimensional FORTRAN-like array data.

(48) Output of data arrays in the code for special tallies

The structure of these data is complicated and the knowledge of the structure of special tallies in the MVP/GMVP code is required to understand them. All data in (48) are given in (40)–(47).

```
write(30) TAG, NETALY, NIETAL, NLETAL, NEDTMX, NBINMX
```

```
write(30) TAG, (IETAL(I),I=1,NIETAL)
```

```
write(30) TAG, (ETALY(I,1),I=1,NLETAL),(ETALY(I,2),I=1,NLETAL)
```

- TAG = 'SPECIAL TALLIES ' (32 characters)
- NETALY : Number of special tallies
- NIETAL : Number of IETAL data
- NLETAL : Number of data in the ETALY array.
- NEDTMX : Number of data of the special tally that has the maximum number of data in all special tallies.
- NBINMX : The results of special tallies are stored as multi-dimensional array data. NBINMX is the largest size of the dimension in all special tallies.
- IETAL: Information about special tallies (bins etc.) stored in an integer array.
- ETALY(I,1): Calculated results (double precision). The results for all special tallies are stored.
- ETALY(I,2): Relative uncertainties (double precision). The standard deviations are given in the % unit.

(49) Calculated results of perturbation in each batch by the differential operator sampling method.

```
write(30) TAG, NUCPT, NTPT
```

```
do INC = 1, NUCPT
```

```
do IPT= 1, NTPT
```

```
write(30) ((XKPB(INC,IPT,J,I),I=1,NBATCH),J=1,6)
```

```
end do
```

```
end do
```

- TAG = 'DIFFERENTIAL OF K: BATCH ' (32 characters)
- NUCPT : Number of perturbed nuclides.
- NTPT : Number of total perturbation tallies.
- XKPB(INC,IPT,J,I) : The differential coefficients ($\partial k / \partial \alpha$; α is the perturbation parameter.) (double precision). J indicates an evaluation method.

16.7 Restart File

MVP/GMVP supports the input/output of a restart file. The restart file for a subsequent calculation is output by specifying the RESTART-FILE option. (This option is effective by default.)

- The contents of the restart file are tightly related to the functions of the code. Therefore, a restart calculation is possible only when the version of a restart file in a subsequent MVP/GMVP calculation is the same as that in the previous calculation.
- Since a restart file is output as the binary file of FORTRAN, the file is not available between the different platforms.

16.8 Fission Source File

A fission source file must be input when the sampling function of #FISSIONFILE is specified in the SOURCE data block. The file is output on I/O unit 9 in the other eigenvalue calculation with the SOURCE-OUTPUT option. The user can make the file according to the following format.

The contents of the file is as follows. At first, there are some “header records” shown here.

- Record 1 : File header (32 characters) and the number of header records (NHREC).

```
HEADER = '%MVP SOURCE FILE%'
write(9) HEADER, NHREC
```

- Record 2 : Version of the file format (32 characters).

```
HEADER = 'source file version 1.0'
write(9) HEADER
```

- Record 3 : Title of input data for this MVP/GMVP calculation (72 characters \times 2).

```
write(9) TITLE1, TITLE2
```

- Record 4 : Arbitrary information about the date, time etc. (64 characters).
- Record 5 – Record NHREC : Unused dummy header records. If these records exist, the character string of not less than 32 bytes must be at the head of these records. No records exist in the MVP/GMVP version 3 since NHREC = 4.

The contents and order of Record NHREC+1 or later are not fixed but the following form is repeated.

- Record of a 32-byte character string including a data name etc. (TAG record)
- Record of data

In the current version, the following data are included.

- Number of source particles included in the file

```
TAG = 'SIZES'
write(9) TAG
write(9) NUSE
```

- Coordinates of particle sources (absolute coordinates)

```

TAG = 'X'
write(9) TAG
write(9) (XXXXF(I),I=1,NUSE)
TAG = 'Y'
write(9) TAG
write(9) (YYYYF(I),I=1,NUSE)
TAG = 'Z'
write(9) TAG
write(9) (ZZZF(I),I=1,NUSE)

```

The data of XXXF, YYYYF, ZZZF are double-precision ones. Even in LATTICE problems, LATTICE coordinates are transformed into absolute coordinates.

- Particle weights

```

TAG = 'W'
write(9) TAG
write(9) (W(I),I=1,NUSE)

```

The data of W are single-precision ones.

- List of used nuclide IDs in the case of MVP (16 characters).

```

TAG = 'NUCLIDE ID'
write(9) TAG
write(9) NUC, (NUCID(I),I=1,NUC)

```

NUC is the number of nuclides. NUCID is a character string of 16 characters.

- Nuclide numbers of the nuclides of which fissions occurred in the case of MVP (ordinal numbers in the nuclide ID list in the previous record).

```

TAG = 'INUF'
write(9) TAG
write(9) (INUF(I),I=1,NUSE)

```

- Neutron energy at which fissions occurred in the case of MVP.

```

TAG = 'EEEEF'
write(9) TAG
write(9) (EEEEF(I),I=1,NUSE)

```

The data of EEEF are single-precision ones.

The minimum data required for this file are some header records and coordinates of generated particles. The file output in MVP can be used in GMVP and vice versa.

17. How to Execute MVP/GMVP

This chapter describes how to execute the MVP and GMVP codes for the beginners. It is supposed that the codes are executed on a UNIX or Linux system. See also Appendix A for how to install the codes and attached documents in distributed media for the usage on environments except for UNIX or Linux.

The basic procedure to execute MVP/GMVP is as follows.

(1) Install the programs.

The user compiles distributed source programs to make the executable files of MVP/GMVP. See Appendix A for the details. This step can be skipped if the codes have been already installed.

(2) Prepare the cross section data.

If the user executes the MVP code, the user has to prepare the cross section data for MVP and index files for cross section search on their system. See Appendix A for the details. This step can be skipped if the data have been already prepared. On the other hand, if the user executes the GMVP code, the user has to prepare the multigroup cross section data since no data are attached. There are two methods to input the data; one is embedding the text-form data in the input file on unit 5 and the other is preparing the cross section file apart from the input file.

(3) Set user's environment.

After the MVP/GMVP code is installed on platforms, the execution environment must be set for each user account. To be concrete, environment variables and command search paths must be set. Usually, the user sets the environment only once but there are some cases where they have to modify the environment for updating the codes, changing the MVP cross sections etc.

(4) Prepare input data.

The input data must be written in the text form and the user must always prepare the data for the execution of the codes. The instruction of the data is described in Chapter 6 or later.

(5) Execute MVP/GMVP.

In general, the user can execute the MVP/GMVP code by issuing commands (scripts) with the command line argument of an input file name.

17.1 Setting User's Environment

The following conditions must be satisfied for the user to execute the MVP/GMVP code.

- (1) The executables compiled from the MVP/GMVP sources and related script files are available. Namely, the codes have been already installed.
- (2) The continuous-energy cross sections are available for MVP. The multigroup cross section data has been prepared for GMVP.

The installation of the codes and libraries are described in Appendix A “Install on UNIX Platforms” in detail. When a new user is added in the environment where the codes and libraries have been already installed, the user has to do is setting the environment variables and the command search paths. Since they must be effective when the user logins, the paths should be written in the setting file of a login shell in the user’s home directory. If the shell is a C-shell(csh) or a tcsh, the file is `$HOME/.cshrc`. If the shell is a Korn-shell(ksh) or a Bourne-again-shell(bash), the file is `$HOME/.profile` or etc. “\$HOME” is the shell statement to refer to an environment variable HOME and this description is also used in the following sections.

17.1.1 Setting Environment Variables

If the user executes the MVP/GMVP code by using `runmvp` and `rungmvp` scripts described in the following section, respectively, the user must specify environment variables `MVP_DIR` and `MVPHOST`. These variables are referred to in the scripts. The path to the directory created in expanding archived MVP/GMVP files must be specified for the environment variable `MVP_DIR`. The environment variable `MVPHOST` is a character string to specify a used platform and an operating system. It is defined in the MVP/GMVP code depending on the platforms and would have been selected properly if the codes had been installed. It is usually selected by the “`getsystem`” command in the MVP/GMVP system but the variable is sometimes specified explicitly in installation.

When the user executes the MVP code by using the `runmvp` script, it is recommended that an environment variable `MVPLIB_DIR` should be also defined to specify the directory including the index files of the MVP cross section libraries. In addition, if arbitrary-temperature cross section libraries are used, an environment variable `MVP_ARTPOOL` can be defined to specify the directory where generated temperature-dependent cross section data are stored. By specifying the variable, the user may save the computation time for processing the cross section data.

17.1.2 Command Search Path

A command search path is the path to a directory where the UNIX shell searches executable files. The `PATH` environment variable is the list of the command search paths. “`$MVP_DIR/etc`” and “`$MVP_DIR/bin/$MVPHOST`” must be added in the list.

17.1.3 Example of Shell Startup Files

The example files for the shell initialization files are prepared in the `$MVPDIR/etc` directory for the above specifications. “`cshrc.ex`” and “`profile.ex`” are those for a C-shell and a Bourne shell, respectively.

The following gives the examples. In the actual situation, the user must change the directory path etc.

● Example: Setting for `.cshrc`

```
#      setenv MVP_DIR    /home/codes/mvp-3.0
#      Case of the use of getsystem
#      setenv MVPHOST ` $MVP_DIR/etc/getsystem -i `
#      setenv MVPLIB_DIR /home/nucdata/MVPLIB
#      if( $?path ) then
#          set path=( $path $MVP_DIR/etc $MVP_DIR/bin/$MVPHOST
#      else
#          set path=( $MVP_DIR/etc $MVP_DIR/bin/$MVPHOST
#      endif
```

● Example: Setting for `.profile`

```

MVP_DIR=/home/codes/mvp-3.0 ; export MVP_DIR
MVPHOST=`$MVP_DIR/etc/getsystem -i`
#   # Unless the getsystem command cannot obtain a correct
#   # system name, it must be defined directly.
# MVPHOST=THE_GREAT_SYSTEM
#   export MVPHOST
MVPLIB_DIR=/home/nucdata/MVPLIB ; export MVPLIB_DIR
PATH=${PATH}:$MVP_DIR/etc:$MVP_DIR/bin/$MVPHOST
#   export PATH

```

17.2 How to Execute MVP

17.2.1 Files Related to MVP

The MVP code requires the following files.

- Input data : Text-form data that describe the content of a calculation. Each user must create it.
- Continuous-energy cross section data and index files (MVP)

The MVP cross section files are installed as the binary-form ones and are referred to by using the cross section index files. The index files must be prepared when the MVP cross section libraries are installed. If the libraries are installed and the user's execution environment has been already set, the user needs not to pay attention to the files in using the `runmvp` command.

After an MVP calculation finishes, the following files are output.

- Standard output (printing output)
- Result file (binary form)
- Restart file (binary form)

The restart file may be used as an input file for another run (with the `RESTART` option).

17.2.2 Execution of MVP with the `runmvp` Command

The use of the `runmvp` command is the easiest way to execute the MVP code on UNIX platforms. In this case, if the execution environment is set correctly, the user can execute the code simply by issuing the following command.

```
% runmvp input-data
```

“*input-data*” is an input file in the text form and “%” is a prompt from the system. Though the MVP code requires the cross section index files etc. besides the input file, the `runmvp` command provides such information to the code automatically and the user does not need to care.

Immediately after the user issues the command and waits for a short while, the prompt appears again and waits for your command input. It does not mean that the MVP job has been finished but the job is running in the background. The user can check the MVP execution by the `ps` command. If the user issues the `ps` command, the user can have the following output on the screen. The output may depend on the platform.

```
% ps
PID TTY      TIME    COMMAND
111 ttyp0    0:02    csh
113 ttyp0    0:25    mvp
114 ttyp0    0:00    ps
112 ttyp0    0:00    time
%
```

This is the list of programs the user is executing. PID is the ID number of a process (program in execution), TTY is the name of a terminal from which the command is issued, TIME is the execution time of a process, COMMAND is the name of the issued command and etc. Unless the mvp process is in execution just after issuing the runmvp command, the process may be possibly terminated due to some errors.

File Names of Calculation Output

Whether a calculation is successful or not, the result of the calculation is output on a file. The runmvp command outputs the final result or the result in progress on the following file by default.

Name of the output file : *xxxxx.hostname.date.hour.minuite.second*

where

- *xxxxx* : Name of an input data. If the name is the form of *.inp, .inp is removed.
- *hostname* : Host name of a platform (computer).
- *date.hour.minuite.second* : Date and time when the runmvp command is issued.

For example, when the runmvp command is issued with an input data sample1.inp on figaro (host name) at 15:30:20 on November 18th in 2002, the name of an output file is

```
sample1.figaro.2002Nov18.15.30.20
```

After the calculation finishes, the user should open the file and check calculated results.

Parameters for the runmvp Command

The user can specify parameters for the runmvp command in addition to the above usage.

```
runmvp {input|i=input} [-lb] [-ss/+ss]
      [o=output-list]
      [xa=arbitrary-temp-neutron-xsec-libray-index-file]
      [xi=neutron-xsec-libray-index-file]
      [xip=photon-xsec-libray-index-file]
      [xie=electron-xsec-libray-index-file]
      [xid=dosimetry-xsec-index-file]
      [ri=restart-input]
      [ro=restart-output]
      [si=source-input]
      [so=source-output]
      [rf=result-file (unit 30)]
```



```
[ex=alternate-MVP-executable]
[preex=placed-before-MVP-executable]
[exeopt=additional-command-parameters]
[CAPITAL_STRING=...|/...]
```

(The parameters in parentheses [...] can be omitted.)

If the user executes the `runmvp` command with no parameters, the brief description for the parameters is displayed on a screen.

The available parameters are explained as follows.

- *input* or *i=input*

The name of an input data file (standard input).

- *o=output-list*

The name of a printing output file (standard output, error output). The default name is

xxxxx.hostname.date.hour.minute.second,

where

- *xxxxx* : Name of an input data file. If the name is given in the form of **.inp*, “.inp” is removed.
- *hostname* : Host name of a platform (computer).
- *date.hour.minute.second* : Date and time when the `runmvp` command is executed.

This parameter is specified when the user wants to change the destination and the name of the output file.

- *xa=arbitrary-temp-neutron-xsec-library-index-file*
- *xi=neutron-xsec-library-index-file*
- *xip=photoatomic-xsec-library-index-file*
- *xipn=photonuclear-xsec-library-index-file*
- *xie=electron-xsec-library-index-file*
- *xid=dosimetry-xsec-index-file*

These parameters are used to specify the index files for neutron (arbitrary and fixed temperatures), photoatomic, photonuclear and electron MVP cross section libraries. The default path and the name of the files are as follows.

Neutron (arbitrary temp.)	<code>\$MVPLIB_DIR/neutron.art.index</code>
Neutron (fixed temp.)	<code>\$MVPLIB_DIR/neutron.index</code>
Photoatomic data	<code>\$MVPLIB_DIR/photon.index</code>
Photonuclear data	<code>\$MVPLIB_DIR/photonuclear.index</code>
Electron	<code>\$MVPLIB_DIR/electron.index</code>

“\$MVPLIB_DIR” is an environment variable. “xid” is the parameter to specify the index file for MVP dosimetry libraries and the default name is not defined.

- *ri=restart-input*
Restart file to be input.
- *ro=restart-output*
Restart file to be output.
- *si=source-input*
Fission source file to be input.
- *so=source-output*
Fission source file to be output.
- *rf=result-file*
File for calculated results output on unit 30. If this parameter is omitted, the file is not saved.
- *ex=alternate-MVP-executable*
The default full path name of the MVP executable is `$MVP_DIR/bin/$MVPHOST/mvp`. If the user wants to use other executable, this parameter is specified. Unless the name of the executable is given with a full path name, the path name is `$MVP_DIR/bin/$MVPHOST/alternate-MVP-executable`.
- *preex=placed-before-MVP-executable*
Command located in prior to the MVP executable. For example, the user can specify the “time” command to measure the execution time or the “mpirex” command to execute an MPI program.
- *exeopt=additional-command-parameters*
Additional command parameters for execution.
- *CAPITAL_STRING=... or /...*
The arguments of the form “Capital string[=...]” or “/...” are passed to the executable as they are. The following arguments can be specified.
 - *NTASK=...* : Number of tasks in multitasking.
 - *MEMORY=...* : Default size of dynamically allocated memories.
 - *RUN-MODE=...* : Run mode. (0/1 = normal/processing only input data)

The command line parameters that begin with “-” or “+” are options related to the execution of the script or the program.

- *-lb*
No buffering of the output data to standard output (unit 6). The printing output data is output to a file immediately.
- *-ss/+ss*
-ss/+ss : The contents of the `runmvp` script aren’t/are output to the printing output file.

17.3 How to Execute GMVP

17.3.1 Files Related to GMVP

The GMVP code requires the following files.

- Input data : Text-form data that describe the content of a calculation.
- Multigroup cross section data (ANISN or DDX form).

The user must prepare both the input data and cross section data to execute GMVP because no cross section data for GMVP are given with source programs unlike MVP. The user can prepare the multigroup cross section data with the MOSRA-SRAC code,¹²⁾ the EXPARAM code system,⁶⁴⁾ the TRANSX code⁶⁵⁾ and so on.

After a GMVP calculation finishes, the following files are output as well as MVP.

- Standard output (printing output)
- Result file (binary form)
- Restart file (binary form)

The restart file may be used as one of the input files for another run (with the RESTART option).

17.3.2 Execution of GMVP with the `rungmvp` Command

The use of the `rungmvp` command is the easiest way to execute the GMVP code on UNIX platforms. In this case, if the execution environment is set correctly, the user can execute the code simply by issuing the following command.

```
% rungmvp input-data
```

“*input-data*” is an input file in the text form. If the input file does not include cross section data and the data must be input from the other file, the command parameter to specify the cross section data must be specified.

```
% rungmvp input-data x= /nn:cross-section-file
```

where *nn* is the number of a FORTRAN I/O unit for GMVP cross section data and must be the same as the one specified in the input file, for which the default value is 1 for the ANISN form and 99 for the DDX form.

After the `rungmvp` command is issued, it works just like `runmvp` for MVP.

Parameters for the `rungmvp` Command

The user can specify parameters for the `rungmvp` command in addition to the above usage.

```
rungmvp {input | i=input} [-lb] [-ss/+ss]
        [o=output-list]
        [x=/n:cross-section-file]
```

```
[ri=restart-input] [ro=restart-output]
[si=source-input] [so=source-output]
[rf=binary-result-file]
[ex=alternate-GMVP-executable]
[preex=placed-before-GMVP-executable]
[exeopt=additional-command-parameters]
```

(The parameters in parentheses [...] can be omitted.)

If the user executes the `rungmvp` command with no parameters, the brief description for the parameters is displayed on a screen.

The available parameters are explained as follows.

- *input* or *i=input*

The name of an input data file (standard input).

- *o=output-list*

The name of a printing output file (standard output, error output). The default name is

xxxxx.hostname.date.hour.minuite.second,

where

- *xxxxx* : Name of an input data file. If the name is given in the form of **.inp*, “.inp” is removed.
- *hostname* : Host name of a platform (computer).
- *date.hour.minuite.second* : Date and time when the `rungmvp` command is executed.

This parameter is specified when the user wants to change the destination and the name of the output file.

- *x=/n:cross-section-file*

Multigroup cross section file to be input. This parameter must be specified when the cross section data are input from a file. *n* is the I/O unit for the cross section data. The user should specify 1 for P_L cross section data and 99 for DDX data except for special cases. *cross-section-file* is the name of a cross section file.

- *ri=restart-input*

Restart file to be input.

- *ro=restart-output*

Restart file to be output.

- *rf=result-file*

File for calculated results output on unit 30. If this parameter is omitted, the file is not saved.

- *ex=alternate-GMVP-executable*

The default full path name of the GMVP executable is `$MVP_DIR/bin/$MVPHOST/gmvp`. If the user wants to use other executable, this parameter is specified. Unless the name of the executable is given with a full path name, the path name is `$MVP_DIR/bin/$MVPHOST/alternate-GMVP-executable`.

- *preex=placed-before-GMVP-executable*

Command located in prior to the GMVP executable. For example, the user can specify the “time” command to measure the execution time or the “mpirec” command to execute an MPI program.

- *exeopt=additional-command-parameters*

Additional command parameters for execution.

17.4 Specification of Memory Size

17.4.1 Dynamic Memory Allocation

Recent computer programs can allocate memory dynamically on most computing platforms. The MVP/GMVP can use the dynamic memory allocation if the directive of “SET DYNAMIC” is defined and it is uncommented (namely, the first column is not “*”) in the file of `src/OBJ.$MVPHOST/ASSIM`. In this case, the user can specify the memory size in an input data file or a command line option.

- Specification of the memory size in an input data.

The following option must be specified in the option data block.

```
DYNAMIC-MEMORY( memory )
```

where *memory* is a positive integer to specify the size of memories and the unit is word (the number of bytes for the REAL-type declaration on a platform).

- Specification of the memory size in a command line.

If the `runmvp` or `rungmvp` command is used, the following option can be specified.

```
runmvp other-parameters MEMORY=memory
```

where *memory* is a positive integer to specify the default size of memories and the unit is word. The memory size specified in a command line is the default size when the DYNAMIC-MEMORY option is not specified. So, if the option is specified in an input data, the memory size defined by the option has priority.

The information of the allocated memory size is output as follows.

```
**** MEMORY FOR VARIABLE-SIZED DATA :    200000000  WORDS  ( 4 BYTES/WORD )
      (      762.94 MEGA-BYTES )
```

It is not always successful to allocate the specified memory size on some environments. In such a case, the following message is output.

```
!!! FAILED TO ALLOCATE MEMORY DYNAMICALLY WITH SPECIFIED SIZE ( 500000000 WORDS)
```

Even in this case, the MVP/GMVP code allocates a default memory size (about 1 Mega words) and tries to continue running. However, it is almost impossible to finish the calculation. The reasons why the specified memory size cannot be allocated may be as follows.

- There are too many processes running on the platform.

If many programs that require a large memory size are executing simultaneously, the MVP/GMVP code possibly fails to allocate memory. The user should terminate unnecessary processes and execute the code again.

- The maximum size of virtual memories is small.

UNIX platforms usually have the function of virtual memory and the user can execute a program that requires more than real memory. But the program does not work depending on the parameters related to the function (the storage of a swap disk, the maximum size of virtual memories per process, etc.). In this case, the user should consult the system administrator.

17.4.2 Static Memory Allocation

If computer programs cannot allocate memory dynamically, they allocate memory statically. The MVP/GMVP code uses the static memory allocation if the directive of “SET DYNAMIC” is not specified or it is commented out (namely, the first column is “*”) in the file of `src/obj.$MVPHOST/ASSIM`. The memory size is determined when the code is compiled.

In order to change the size, `src/mvp/aaalloc.f` and `src/gmvp/aaalloc.f` must be modified and re-compiled for MVP and GMVP, respectively. The following PARAMETER statement must be modified for both codes.

```
PARAMETER (MAXMEM = nnnnnnnn )
```

“nnnnnnnn” is the memory size in word.

Acknowledgments

The authors wish to express their gratitude to many contributors and code users. They would like to especially thank the following people. Dr. Masayuki Nakagawa started the development project of a new Monte Carlo code and established the foundation of the MVP/GMVP code. The late Makoto Sasaki supported the project with his excellent programming skill and gave us a lot of useful comments. Mr. Kunio Kaneko supported the project in terms of MVP library generation. Dr. Kazuaki Kosako of Shimizu Corporation supported the development of the photonuclear model in the MVP code. Dr. Toru Yamamoto of NRA supported the development of the exact resonant elastic scattering model. Dr. Motomu Suzuki of CRIEPI supported the development of the 64-bit version MVP code.

The implementation of the photonuclear model is a part of the result of “Development of Active Non-destructive Assay Technique using Photofission for Uranium-Bearing Waste Discharged from Uranium Enrichment Facility” entrusted by the Ministry of Education, Culture, Sport, Science and Technology of Japan (MEXT).

References

- 1) T. Mori, M. Nakagawa, and M. Sasaki. Vectorization of Continuous Energy Monte Carlo Method for Neutron Transport Calculation. *J. Nucl. Sci. Technol.*, **29**[4], pp.325–336 (1992).
- 2) Message Passing Interface Forum. MPI: A Message-Passing Interface Standard (1994).
- 3) T. Mori and M. Nakagawa. MVP/GMVP: General Purpose Monte Carlo Codes for Neutron and Photon Transport Calculations based on Continuous Energy and Multigroup Methods. JAERI-Data/Code 94-007, 152p. (1994). [in Japanese].
- 4) K. Shibata, et al. Japanese Evaluated Nuclear Data Library, Version 3, JENDL-3. JAERI 1319, 516p. (1990).
- 5) T. Nakagawa, et al. Japanese Evaluated Nuclear Data Library, Version 3, Revision 2. *J. Nucl. Sci. Technol.*, **32**, pp.1259–1271 (1995).
- 6) P. F. Rose and C. L. Dunford (Eds.). ENDF-102; Data Formats and Procedures for the Evaluated Nuclear Data File, ENDF. BNL-NCS-44945 (1990).
- 7) M. Herman and A. Trkov (Eds.). ENDF-6 Formats Manual. BNL-90365-2009 (2009).
- 8) Y. Nagaya, K. Okumura, T. Mori, and M. Nakagawa. MVP/GMVP II : General Purpose Monte Carlo Codes for Neutron and Photon Transport Calculations Based on Continuous Energy and Multigroup Methods. JAERI 1348, 388p. (2005).
- 9) T. Mori and Y. Nagaya. Comparison of resonance elastic scattering models newly implemented in MVP continuous-energy Monte Carlo code. *J. Nucl. Sci. Technol.*, **46**[8], pp.793–798 (2009).
- 10) M. Ouisloumen and R. Sanchez. A Model for Neutron Scattering Off Heavy Isotopes That Accounts for Thermal Agitation Effects. *Nucl. Sci. Eng.*, **107**[3], pp.189–200 (1991).
- 11) T. Sakurai, K. Kosako, and T. Mori. A Conceptual Design Study for Active Nondestructive Assay System by Photon Interrogation for Uranium-Bearing Waste with MVP Code and Evaluated Photonuclear Data. *Prog. Nucl. Sci. Technol.*, **2**, pp.318–329 (2010).
- 12) K. Okumura. MOSRA-SRAC; Lattice calculation module of the modular code system for nuclear reactor analyses MOSRA. JAEA-Data/Code 2015-015, 162p. (2015).
- 13) T. Hazama, G. Chiba, W. Sato, and K. Numata. SLAROM-UF : Ultra fine group cell calculation code for fast reactor – version 20090113 –. JAEA-Review 2009-003, 59p. (2009).
- 14) W. R. Martin. Challenges and Prospects for Whole-Core Monte Carlo Analysis. *Nucl. Eng. Technol.*, **44**, pp.151–160 (2012).
- 15) J. E. Hoogenboom, W. R. Martin, and B. Petrovic. Monte Carlo Performance Benchmark for Detailed Density Calculation in a Full Size Reactor Core, Benchmark specifications Revision 1.2, July 2011 (2011).
- 16) N. Horelik, B. Herman, B. Forget, and K. Smith. Benchmark for Evaluation and Validation of Reactor Simulation (BEAVRS). *Proc. of M&C 2013*, pp. 2986–2999, Sun Valley, Idaho, USA, 5-9 May 2013.

- 17) M. Suzuki and Y. Nauchi. Analysis of BEAVRS Benchmark Problem by Using Enhanced Monte Carlo Code MVP with JENDL-4.0. *Proc. of ANS M&C2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method*, Nashville, Tennessee, USA, 19-23 April 2015.
- 18) M. B. Emmett. MORSE-CGA, A Monte Carlo Radiation Transport Code with Array Geometry Capability. ORNL-6174 (1985).
- 19) L. M. Petrie and N. F. Landers. KENO-V.a: An Improved Monte Carlo Criticality Program with Supergrouping, Vol. II, Sect. F11 of SCALE: A Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation. NUREG/CR-0200, Rev.5 (ORNL/NUREG/CSD-2R5) (1997).
- 20) X-5 Monte Carlo Team. MCNP — A General Monte Carlo N-Particle Transport Code, Version 5. LA-UR-03-1987 (2003).
- 21) R. N. Blomquist. Status of the VIM Monte Carlo Neutron/Photon Transport Code. *Proceedings of 12th Biennial RPSD Topical Meeting of the American Nuclear Society*, Santa Fe, New Mexico, USA, April 14-18 2002.
- 22) G. I. Bell and S. Glasstone. *Nuclear Reactor Theory*. Van Nostrand Reinhold Company, New York (1970).
- 23) J. Spanier and E. M. Gelbard. *Monte Carlo Principles and Neutron Transport Problem*, chapter 2.6 The Track Length Estimator, pp. 66–76. Addison-Wesley Publishing Company (1969).
- 24) A. Dubi, T. Elperin, and H. Rief. On Confidence Limits and Statistical Convergence of Monte Carlo Point-Flux Estimators with Unbounded Variance. *Ann. Nucl. Energy*, **9**, pp.675–682 (1982).
- 25) S. K. Fraley and T. J. Hoffman. Bounded Flux-at-a-Point Estimators for Multigroup Monte Carlo Computer Codes. *Nucl. Sci. Eng.*, **70**, pp.14–19 (1979).
- 26) T. Mori, K. Okumura, and Y. Nagaya. Status of JAERI's Monte Carlo Code MVP for Neutron and Photon Transport Problems. *Monte Carlo 2000 Conference*, pp. 625–630, Lisbon, 23-26 October 2000.
- 27) T. Mori, K. Okumura, Y. Nagaya, and H. Ando. Monte Carlo Analysis of HTTR with the MVP Statistical Geometry Model. *Trans. Am. Nucl. Soc.*, **83**, pp.283–284 (2000).
- 28) T. Mori, M. Nakagawa, and M. Sasaki. One-, Two- and Three Dimensional Transport Codes Using Multi-Group Double-Differential Cross Sections. JAERI 1314, 151p. (1988).
- 29) T. Mori, M. Nakagawa, and K. Kaneko. Neutron Cross Section Library Production Code System for Continuous Energy Monte Carlo Code MVP — LICEM. JAERI-Data/Code 96-018 (1996). [in Japanese].
- 30) K. Okumura, et al. Validation of a Continuous-Energy Monte Carlo Burn-up Code MVP-BURN and Its Application to Analysis of Post Irradiation Experiment. *J. Nucl. Sci. Technol.*, **29**[11], pp.128–138 (2000).
- 31) M. Nakagawa, T. Mori, and M. Sasaki. Comparison of Vectorization Methods Used in a Monte Carlo Method. *Nucl. Sci. Eng.*, **107**, pp.58–66 (1991).

- 32) T. Mori, Y. Nagaya, K. Okumura, and K. Kaneko. Production of MVP Neutron Cross Section Libraries Based on the Latest Evaluated Nuclear Data Files. JAERI-Data/Code 2004-011, 119p. (2004). [in Japanese].
- 33) R. C. Little. New Photon Library from ENDF Data. LA-UR-03-0164 (2003).
- 34) RSIC COMPUTER CODE SELECTION, MCNP4, Monte Carlo Neutron and Photon Transport Code System. CCC-200A/B (1991).
- 35) C. J. Everett and E. D. Cashwell. MCP Code Fluorescence Routine Revision. LA-5240-MS (1973).
- 36) IAEA. Handbook on Photonuclear Data for Applications Cross-sections and Spectra. IAEA-TECDOC-1178 (2000).
- 37) D. E. Cullen, et al. The 1989 ENDF/B Pre-processing Codes. IAEA-NDS-39 (1989).
- 38) H. Akima. Algorithm 433, Interpolation and Smooth Curve Fitting Based on Local Procedure. *Comm. ACM*, **15**, pp.914–918 (1972).
- 39) J. Spanier and E. M. Gelbard. *Monte Carlo Principles and Neutron Transport Problem*, chapter 5.4 Refinements of the Formulation, pp. 192–198. Addison-Wesley Publishing Company (1969).
- 40) F. B. Brown, W. R. Martin, and D. A. Calahan. A Discrete Sampling Method for Vectorized Monte Carlo Calculations. *Trans. Am. Nucl. Soc.*, **38**, pp.354–355 (1981).
- 41) V. V. Kolesov and V. F. Ukraintsev. Temperature effects and resonance elastic cross section influence on secondary energy distributions of scattered neutrons in the resolved resonance region. *Proc. of PHYSOR-2006*, Vancouver, BC, Canada, 10-14 September 2006.
- 42) R. Dagan and C. H. M. Broeders. On the effect of resonance dependent scattering-kernel on fuel cycle and inventory. *Proc. of PHYSOR-2006*, Vancouver, BC, Canada, 10-14 September 2006.
- 43) D. Lee, K. Smith, and J. Rhodes. The impact of ^{238}U resonance elastic scattering approximations on thermal reactor Doppler reactivity. *Ann. Nucl. Energy*, **36**, pp.274–280 (2009).
- 44) R. Dagan, B. Becker, and D. Roubtsov. Evaluation of the CANDU 6 neutron characteristics in view of application of the resonance dependent scattering kernel in MCNP(X). *Proc. Nucl. Sci. Technol.*, **2**, pp.782–787 (2011).
- 45) T. Ueki, T. Mori, and M. Nakagawa. Error Estimations and Their Biases in Monte Carlo Eigenvalue Calculations. *Nucl. Sci. Eng.*, **125**, pp.1–11 (1997).
- 46) R. P. Feynman, F. de Hoffmann, and R. Serber. Dispersion of the Neutron Emission in U-235 Fission. *J. Nucl. Energy*, **3**, pp.64–69 (1956).
- 47) J. Terrell. Distribution of Fission Neutron Numbers. *Phys. Rev*, **108**, pp.783–789 (1957).
- 48) L. Koblinger. Direct Sampling from the Klein-Nishina Distribution for Photon Energies Above 1.4 MeV. *Nucl. Sci. Eng.*, **56**, pp.218–219 (1975).
- 49) Y. S. Horowitz, A. Dubi, and S. Mordechai. The Monte Carlo Nonuniform Sampling Technique Applied to the Klein-Nishina Probability Density Function. *Nucl. Sci. Eng.*, **60**, pp.461–463 (1976).
- 50) P. Vertes. Test Calculations with IAEA Photonuclear Data Library. INDC(HUN)-035 (2001).

- 51) I. Murata, T. Mori, M. Nakagawa, and H. Itakura. Continuous Energy Monte Carlo Calculations of Randomly Distributed Spherical Fuels Based on Statistical Geometry Model. JAERI-Research 96-015, 44p. (1996). [in Japanese].
- 52) I. Murata, A. Takahashi, T. Mori, and M. Nakagawa. New Sampling Method on Continuous Energy Monte Carlo Calculation for Pebble Bed Reactors. *J. Nucl. Sci. Technol.*, **34**[8], pp.734–744 (1997).
- 53) I. Murata, T. Mori, M. Nakagawa, and H. Shirai. Packing Simulation Code to Calculate Distribution Function of Hard Spheres by Monte Carlo Method: MCRDF. JAERI-Data/Code 96-016, 79p. (1996). [in Japanese].
- 54) R. K. Meulekamp and S. C. van der Marck. Calculating the Effective Delayed Neutron Fraction with Monte Carlo. *Nucl. Sci. Eng.*, **152**, pp.142–148 (2006).
- 55) Y. Nauchi and T. Kameyama. Proposal of Direct Calculation of Kinetic Parameters β_{eff} and Λ Based on Continuous Energy Monte Carlo Method. *J. Nucl. Sci. Technol.*, **42**[6], pp.503–514 (2005).
- 56) Y. Nagaya, G. Chiba, T. Mori, D. Irwanto, and K. Nakajima. Comparison of Monte Carlo calculation methods for effective delayed neutron fraction. *Ann. Nucl. Energy*, **37**, pp.1308–1315 (2010).
- 57) T. Mori, K. Okumura, Y. Nagaya, and M. Nakagawa. Application of Continuous Energy Monte Carlo Code MVP to Burn-up and Whole Core Calculations Using Cross Sections at Arbitrary Temperatures. *Proc. of Int. Conf. on Mathematics and Computer, Reactor Physics and Environmental Analysis in Nuclear Applications (M&C'99)*, pp. 987–996, Madrid, Spain, 27-30 September 1999.
- 58) W. W. Engle, Jr. A user manual for anisn, a one dimensional discrete ordinates transport code with anisotropic scattering. K-1693 (1967).
- 59) Y. Komuro, et al. MAIL3.0: A Computer Program Calculating Cross Section Sets for SIMCRI, ANISN, KENO-IV, MULTI-KENO and MULTI-KENO-II. JAERI-M 90-126, 125p. (1990). [in Japanese].
- 60) N. M. Green, et al. XLACS: A Program to Produce Weighted Multigroup Neutron Cross Sections from ENDF/B. ORNL-TM-3646 (1972).
- 61) K. D. Lathrop. DTF-IV, A Fortran Program for Solving the Multigroup Transport Equation with Anisotropic Scattering. LA-3373 (1965).
- 62) Y. Nagaya and T. Mori. Calculation of effective delayed neutron fraction with Monte Carlo perturbation techniques. *Ann. Nucl. Energy*, **38**, pp.254–260 (2011).
- 63) Y. Nagaya. Calculation of reactor kinetics parameters with Monte Carlo differential operator sampling. *Ann. Nucl. Energy*, **82**, pp.226–229 (2015).
- 64) S. Iijima, Y. Kato, K. Takasaki, and S. Okajima. Fast Reactor Nuclear Physics Parameters Calculation Code System “EXPARAM”. JAERI-Data/Code 2004-016, 91p. (2004). [in Japanese].
- 65) R. E. MacFarlane. TRANSX 2 : A Code for Interfacing MATXS Cross-Section Libraries to Nuclear Transport Codes. LA-12312-MS (1992).
- 66) K. Kosako. Present Status of Cross Section Libraries. *Proc. of 3rd Specialists' Meeting on Nuclear Data for Fusion Reactors*, JAERI-Conf 96-005, pp. 55–62, Tokai, Japan, 29-30 November 1996.

Appendix

A. Installation of MVP/GMVP

This chapter describes how to install the MVP/GMVP code and cross section data on UNIX or Linux platforms. The user requires the knowledge about some UNIX commands to install and execute the code on the platforms.

A.1 How to Install the MVP/GMVP Code on UNIX or Linux Platforms

The procedure to install the MVP/GMVP code is briefly described in the following. The details of the procedure are described in the next section or later. The user should see some documents attached to the distributed source files (INSTALL.guide etc.).

(1) Check available computer resources.

- Disk space of about 80 Mbytes is required to make the executables from the program sources.
- A large amount of disk space is required for the cross section data of the MVP code. To install the cross section libraries for MVP (MVP libraries), disk space of about 1.1 Gbytes is required for JENDL-4.0. Doubled disk space is necessary for the 64-bit version of MVP (more than 2 Gbytes of memory is available) since the cross section data becomes double-precision real data.
- FORTRAN and C compilers and utilities such as make, sed etc. are required to compile source programs.
- Some amount of memory is required to run the MVP code. The user should check the available memory for a large-scale calculation.

(2) Retrieve data from a distributed medium.

If the MVP package data is distributed with a DVD disk, the user copies and decompresses it on a hard disk drive where there is enough space available.

(3) Set the environment to execute the programs.

The user sets the environment variables of directory names for program sources and MVP libraries and the command search path.

(4) Compile/link programs.

The user compiles and links the MVP and GMVP source files and some utility programs such as a converter of MVP libraries. The user only has to execute some simple commands for the supported platforms.

(5) Process cross section data.

The user converts the distributed cross section data (usually text form) into those in the available form.

(6) Run executables with sample input data.

Some sample input data and their results are attached. The user checks the programs by running the executables.

The above process requires a lot of time for slow machines (in particular, compilation, process of cross section data and test calculations).

After finishing the process, the user who installed programs can use the programs on the installed platform. If the user wants to share the programs with other users or install on the same type of platforms, the user should perform a part of the above process.

- Share the programs with other users:

Process "Set the environment to execute the programs" should be performed for a new user.

- Install on the same type of platform:

If the machine type and operating system of the platform are the same, the user can copy the files and directory structure to the platform where the programs are not installed and set the environment to execute the programs. If the file systems are shared with NFS etc., it is unnecessary to copy the files and directory structure.

If the user wants to install the programs on other platforms, basically the user must follow all the above process. But if the platforms have the same binary and floating formats, the cross section data can be shared.

There are some cases where the procedure is somewhat different from the standard one described in the following. See A.5 and documents attached to the source files.

A.1.1 Extracting Files

First, the user must prepare enough disk space to create files related to MVP/GMVP. The files are classified into 2 parts.

- (a) Program source files, shell scripts, sample input data etc.

They occupy the disk space of about 80 Mbytes after installation.

- (b) Cross section libraries for MVP.

The user requires about 1.1 Gbytes (about 2.1 Gbytes for 64-bit version) for the JENDL-4.0 library distributed with MVP/GMVP Version 3.

Unless there is enough space to store both (a) and (b) in the same disk, each can be stored in the different disks.

Extracting Files from Archive Files

The archived file of program source files is `mvp3.0.tar.gz`. The user copies this file to an appropriate directory on a hard disk and extract the source files in a directory where the user installs the code. If the archived file is distributed with a DVD, the user can mount it and extract directly from the mounted directory.

Example:

```
tar xvfz mvp3.0.tar.gz
```

or

```
gzip -dc mvp3.0.tar.gz | tar xvf -
```

When this process is normally terminated, the directory of “mvp-3.0” is created. Directories LOCAL, burnrun, etc, gmvplib, gmvprun, mvprun, porting, src and template are also created in the mvp-3.0 directory. The content of each directory is listed in Table A.1. The bin directory is automatically created in compilation.

Table A.1 Content of directories related to MVP/GMVP code

Directory	Content
src	Program sources, object files, Makefile etc.
bin	Executables
etc	Command files to install or execute programs (shell script) etc.
mvprun	Sample inputs and outputs for MVP
gmvprun	Sample inputs and outputs for GMVP
gmvplib	Cross section data for GMVP sample inputs
template	Template files for MVP/GMVP input data
testcase	Sample inputs for special-purpose calculations
artrun	Sample inputs and outputs for MVP-ART
burnrun	Sample inputs and outputs for MVP-BURN
porting	Commands to export the MVP/GMVP code on non-UNIX environment etc.
LOCAL	Directory of code maintenance or working area for the code user

A.1.2 Setting Up the Environment to Execute Programs

In order to execute MVP/GMVP, the user must set up the environment as follows.

(1) Define environmental variables MVP_DIR, MVPHOST and MVPLIB_DIR.

- MVP_DIR: Path name of the directory where the files related to the MVP/GMVP code (Table. A.1) are stored.
- MVPHOST: Platform name which is uniquely determined in the MVP/GMVP code.
- MVPLIB_DIR: Name of the directory where index files for MVP cross section data are stored.

MVP_DIR and MVPHOST must be specified. MVPLIB_DIR is not necessary to be defined but it is desirable when MVP is executed with the runmvp command.

(2) Set the command search path to directories \$MVP_DIR/bin/\$MVPHOST and \$MVP_DIR/etc.*

Since the above environment is desired to be set when the user logs in to a machine, the user should add the above definitions to the initial setup file of a login shell. If the user's login shell is C-shell or tcsh, the setup file is .cshrc. If the user's login shell is B-shell or bash, it is .profile (or .bashrc etc.). If no setup file exists, the user can create it. Examples for .cshrc and .profile are shown below. Since the sample files (cshrc.ex and profile.ex) are stored in the directory \$MVP_DIR/etc, the user can copy either.

*MVP_DIR, MVPHOST etc. are environmental variables.

- Example for .cshrc

```
#
# -- C-shell startup script for MVP/GMVP --
#
#      mmmmmmmmm : directory path on which MVP/GMVP files are installed.
#      xxxxxxxxx : directory path on which MVP library is placed.
#
#      Insert in user's .cshrc file and modify mmmmmmmmm & xxxxxxxxx.
#
setenv MVP_DIR      mmmmmmmmm
setenv MVPLIB_DIR   xxxxxxxxx
setenv MVPHOST ` $MVP_DIR/etc/getsystem -i `
if( $?MVPHOST ) then
    if( $?path ) then
        set path=($MVP_DIR/etc $MVP_DIR/bin/$MVPHOST $path)
    else
        set path=($MVP_DIR/etc $MVP_DIR/bin/$MVPHOST)
    endif
endif
endif
```

- Example for .profile

```
#
# -- Bourne or Korn-shell startup script for MVP/GMVP --
#
#      mmmmmmmmm : directory path on which MVP/GMVP files are installed.
#      mmmmmmmmm : directory path on which MVP/GMVP files are installed.
#      xxxxxxxxx : directory path on which MVP library is placed.
#
#      Insert in user's .profile file and modify mmmmmmmmm & xxxxxxxxx.
#
MVP_DIR=mmmmmmmmmm
export MVP_DIR
MVPLIB_DIR=xxxxxxxxx
export MVPLIB_DIR
if [ "$MVPHOST" = "" ]; then
    MVPHOST=`$MVP_DIR/etc/getsystem -i`
fi
export MVPHOST
PATH=$MVP_DIR/etc:$MVP_DIR/bin/$MVPHOST:$PATH
export PATH
```

When the user adds the above definitions to a setup file, the user only has to define the environmental variables MVP_DIR, MVPLIB_DIR by changing the character strings mmmmmmmmm and xxxxxxxxx.

In the above scripts, the environmental variable MVPHOST is defined automatically but there are some cases where a correct platform is not selected. In such a case, the user must define MVPHOST explicitly.

The user should be careful for the location of the definitions in the setup file because some parts of the file may be skipped by conditional statements.

After the .cshrc or .profile file is modified, the user should logout once and then login. The user can check the definition of the environmental variables with the printenv or env command.

Output example:

```

...
PATH=/home/codes/mvp-3.0/etc:/home/codes/mvp-3.0/bin/LINUXGLIBC:./bin:
/usr/bin:/usr/X11R6/bin
...
MVP_DIR=/home/codes/mvp-3.0
MVPLIB_DIR=/server1/MVPlib
MVPHOST=LINUXGLIBC
...

```

The user should check whether the environmental variable `PATH` includes directories for MVP/GMVP and the variables `MVP_DIR`, `MVPLIB_DIR` and `MVPHOST` are defined. Unless the variables are correctly defined, there may be an error in `.cshrc` or `.profile` or incorrect directories may be specified. The environmental variable `MVPHOST` is defined automatically depending on platforms as shown in Table A.2.

Table A.2 Environmental variable `MVPHOST` for each platform

MVPHOST	Platform (machine or OS)
LINUXGLIBC	Linux x86(glibc2)
FREEBSD	FreeBSD x86
X86SOL2	Solaris2.x x86
CYGWIN	Cygwin (Windows)
WIN	MS-Windows
UNKNOWN	Unsupported platform

Even if the user is installing on a platform where `MVPHOST` is not `UNKNOWN` in Table A.2, there may be some cases where `MVPHOST` is not correctly defined or `MVPHOST` is defined as `UNKNOWN`. In this case, the user should issue the command “`uname -a`” and inform developers.

Even if the platform is not recognized correctly, the user can proceed to the next step “Compilation and Linkage”. However, the additional effort may be required in the installation.

A.1.3 Compilation and Linkage

After setting the environmental variables and the command search path, the user can compile and link programs by using the `makemake` and `mvpmake` commands. These commands can be issued in any directory (for example, `LOCAL` directory) if the command search path is properly set. The user should also read “Platform Dependence” (Section A.5) before the installation.

Execution of the `makemake` Command

This command generates `Makefile` for the user’s platform.

Command:

```
makemake
```

The `makemake` command has some options and they can be shown with the “`-h`” option.

```

makemake [-src source-directory] [-c] [-ic] [-noc] [-h] [-a]
          [sys=system] [mf=...] [mm=...] [as=...]
-src source-directory: use non-default source directory.
-c      : use current system for OBJ.xxx (non interactive)
-ic     : add include file dependency rules.
-noc    : do not use C-language coded source files.
-h      : only print this message.
-a      : linkage thru *.a files.
src=source-directory: use non-default source directory.
          (alternative form to -src)
sys=... : use OBJ.sys (non interactive. override -c)
mf=...  : specify name of Makefile.
mm=...  : specify alternate "MakeMacro" file.
as=...  : specify alternate "ASSIM" file.

```

When makemake is executed, the list of available systems is displayed on a screen as shown below. If the selected system is correct, the user enters “0” (zero) and presses the ENTER key. If the selected system is incorrect, the user should choose a correct system and press the ENTER key. If the user wants to generate an executable for MVP parallelized with MPI, the user should select the number with “.mpi” for a target system. If the user wants to generate an executable for the 64-bit version of MVP, the user should select the number with “.mvp64” for a target system.

=== Makefile builder for target system ==

```

Default source directory : /home/codes/mvp-3.0/src
(1) CYGWIN
(2) CYGWIN.mpi
(3) FREEBSD
(4) LINUXGLIBC
(5) LINUXGLIBC.mpi
(6) LINUXGLIBC.mvp64
(7) LINUXGLIBC.mvp64.mpi
(8) UNKNOWN
(9) WIN
(10) X86SOL2
( current system : LINUXGLIBC )
select target system by number (*:all 0:current q:quit) -->

```

If the following message is output, the process is normally terminated and Makefile is created in the directory \$MVP_DIR/src/OBJ.\$MVPHOST.

```

-----
Makefile making completed.
-----

```

If there are some troubles, the user should check the following things and then try again.

- Ensure that the directory \$MVP_DIR/src/OBJ.\$MVPHOST has a write permission.
- Ensure that the command \$MVP_DIR/etc/makemake has an execute permission.

Execution of the mvpmake Command

“mvpmake” is the command to create the executables of the MVP/GMVP code. If the directory path is correctly set, the command can be executed in any directory. The standard usage of the command is as follows.

```
mvpmake target ...
```

target is the character string for an executable to be created and is specified as follows.

- Making the MVP executable

```
mvpmake mvp (for single version)
```

or

```
mvpmake .mpi mvp (for MPI version)
```

or

```
mvpmake .mvp64 mvp (for 64-bit version)
```

- Making the GMVP executable

```
mvpmake gmvp (for single version)
```

or

```
mvpmake .mpi gmvp (for MPI version)
```

or

```
mvpmake .mvp64 gmvp (for 64-bit version)
```

- Making the cross section processing tools (NLB2TXT, NTXT2LB, GMVPLBCV) for MVP/GMVP

```
mvpmake xsectool
```

If *target* is omitted, the MVP and GMVP executables are created at once.

When the command is terminated, the user should check the last 2 lines of the command output (in the case of MVP).

```
%%% information of MVP executable %%%
-rwxrwxr-- 1 user group 1159468 Nov 01 15:01
../bin/LINUXGLIBC/mvp
```

(The third line follows the end of the second line.) The second line shows the information about the executable including the date and time of the creation (Nov 01 15:01 in the above example). Unless the above information is output, some errors possibly occur in compiling or linking. (If executables have been already created, the information is not output. But the user can distinguish because no information about compilation and linkage is also output in this case.)

The mvpmake command can be usually terminated without troubles unless the environmental variable MVPHOST is “UNKNOWN”. But some errors may occur in the case where the environment is different in the version of OS, compilers etc. In particular, some errors possibly occur in the environment of “UNKNOWN”. If some errors occur, the user can check the cause of the errors by saving the output list. To

save the list, the user executes the `mvpmake` command with the output destination specified. (Example: `mvpmake mvp >& makeerr.log`) The user should consult the developers if the errors cannot be dissolved or source codes should be modified.

If the user wants to re-compile the sources because of the update of a compiler, the user must delete object files. This can be also done with the `mvpmake` command.

```
mvpmake cleanmvp      (Delete all object files for MVP)
mvpmake cleangmvp     (Delete all object files for GMVP)
mvpmake cleanall      (Delete all object files for MVP and GMVP)
```

See Appendix F about directories where the program sources for MVP/GMVP are stored and the procedure to execute “make” automatically depending on platforms.

A.2 Preparation of Cross Section Data

A.2.1 Processing of MVP Cross Section Data

MVP utilizes the binary form of cross section data, which are stored as files for each nuclide and particle. Since the text form of the cross section data are usually distributed, the user must convert them into the binary form. The conversion is performed with a cross section processing tool `NTXT2LB` compiled by the `mvpmake` command. The user must prepare it before the conversion. See Appendix E for the usage of `NTXT2LB`.

The shell script to use the processing tool is prepared for each released cross section data library. The shell script is bundled with the archived library file and the usage of the script. Table A.3 shows the set of standard libraries to run MVP.

Table A.3 Shell scripts to prepare MVP libraries

Shell script	Library to be processed
<code>Makelib_nJ40.sh</code>	Neutron arbitrary-temperature library for JENDL-4.0
<code>Makelib_paB68.sh</code>	Photoatomic library based on ENDF/B-VI.8
<code>Makelib_pn.sh</code>	Photonuclear library
<code>Makelib_el.sh</code>	Electron library
<code>Makelib_ds.sh</code>	Dosimetry library

The usage of these shell scripts is as follows.

Command:

```
./Makelib_nJ40.sh
```

The user does not have to install all the libraries above depending on problems to be solved. If there is enough space on a hard disk drive, it is convenient to address any types of problems.

There are some cases where the user wants to install another MVP neutron library to perform integral tests for evaluated nuclear data. In such a case, the user can install the library with the bundled shell script. For example, if the user wants to install the neutron cross section library of ENDF/B-VII.0, the user executes `Makelib_nB70.sh` as described above. A file named `neutron.art.B70.index` is automatically created and the user appends it into `neutron.art.index` as follows.

```
cat neutron.art.B70.index >> neutron.art.index
```

The user can use both JENDL-4.0 and ENDF/B-VII.0 for neutron cross section data.

A.2.2 Processing of GMVP Cross Section Data

The GMVP code does not require specified cross section data unlike the MVP code. GMVP can utilize the ANISN form of cross section data and the DDX form for the MORSE-DD code in principle. This distribution includes the text form of the cross section data for sample calculations.

- Text form of GMVP cross section data in this distribution
 - For sample 2: \$MVP_DIR/gmvplib/ddxj3.txt.Z (compressed by the compress command)
 - For sample 3: \$MVP_DIR/gmvplib/n46g26.txt.Z (compressed by the compress command)

(The cross section data is unnecessary for sample 1.)

The user must convert these data into the binary form because GMVP reads only the binary form in execution.

At first, the user must restore the compressed data.

Command:

```
uncompress ddxj3.txt.Z
uncompress n46g26.txt.Z
```

or using the gzip command instead of the compress command.

```
gzip -d ddxj3.txt.Z
gzip -d n46g26.txt.Z
```

Next, the user execute the conversion program.

Command:

```
gmvplbcv
```

The program inquires a conversion type (binary -> text or text -> binary) as follows. The user should enter "2".

```
*****
*** GMVP/ANISN/KENO library form conversion ***
*****
conversion type
  1.  binary -> text
  2.  text -> binary
INPUT>
```

Then the program also inquires the name of an input file.

```
== INPUT (TEXT) LIBRARY -->
```

Here the user should enter the name of the text file (ddxj3.txt or n46g26.txt).

```
== OUTPUT (BINARY) LIBRARY -->
```

Finally, the user should enter the name of an output file. The name must be different from that of the input file.

A.3 Test Calculation with Samples

A.3.1 Test Calculation for MVP

To check the execution of MVP, sample input files listed in Table A.4 are prepared in a directory \$MVP_DIR/mvprun. The first three of the examples correspond to the examples described in Chapter 6 of Reference 3.

Table A.4 Sample input data files for MVP

File	Problem
sample1.inp	Unit fuel assembly calculation with a lattice (no nested lattice).
sample2.inp	System comprised of multiple fuel assemblies with nested lattices.
sample3.inp	Neutron-photon coupled problem.
sample4.inp	Example of the statistical geometry model.
sample5.inp	Example of the FREE-FRAME-LATTICE option.
sample6.inp	Example of tally dimensions SOURCE-REGION and MARKER-REGION.
sample7.inp	Example of a Feynman- α calculation.
sample8.inp	Example of surface crossing and collision estimators.
sample9.inp	Example of surface crossing and point estimators.

Each example solves the following problem.

- Example 1 : An LMFBR assembly model with a hexagonal LATTICE.
- Example 2 : An assembly model with nested LATTICE geometry.
- Example 3 : Neutron/photon coupling problem for the experimental analysis at FNS/JAERI.
- Example 4 : Statistical geometry. Calculation space except for the perfect reflector is filled with graphite. This calculation is carried out to evaluate volume ratio of zones.
- Example 5 : The doubly (inner and outer) bounded hexagonal LATTICE assembly model described with the FREE-LATTICE-FRAME option.
- Example 6 : An example of special tally functions with tally dimensions SOURCE-REGION and MARKER-REGION. This is also a sample of eigenvalue problem with photon generation.
- Example 7 : Noise analysis for Feynman-Y value calculation. The PERIODIC-TIME and ANALOG-FISSION options and tally functions to estimate the Y value are used.
- Example 8 : Sample of surface crossing and collision estimators defined in the TALLY data block.
- Example 9 : Sample of surface crossing and point estimators defined in the TALLY data block.

If the environment variables are set correctly, MVP can be executed only with the following command.

```
runmvp input-data
```

where *input-data* is the file name of an MVP input data. For example, the following commands should be issued to execute MVP Example 2.

```
cd $MVP_DIR/mvprun
runmvp sample2.inp
```

See Section 17.2.2 for the details of the runmvp command.

The user requires slightly long calculation time to complete the above 3 input data (sample1.inp, sample2.inp, sample3.inp). Simplified input data (sample1t.inp, sample2t.inp, sample3t.inp) are also prepared to check the execution of MVP. The number of histories is decreased by one tenth to one third for the simplified data.

The user can confirm whether the MVP code is in execution with the command ps etc.

Example of the ps command and its output

```
% ps
  PID   TTY      TIME    COMMAND
  111    tty0      0:02     csh
  113    tty0      0:25     mvp
  114    tty0      0:00     ps
  112    tty0      0:00     time
%
```

If MVP is not running immediately after the runmvp command is issued, it is highly possible that MVP is terminated due to a fatal error. In such a case, the output messages should be checked.

Output File of the MVP Code

When MVP is executed with the runmvp command, the results and messages are output to the following file.

```
casename.hostname.date.hour.minuite.second
```

where

- casename : Input file name or a part of it. If the name ends with “.inp”, it is removed.
- hostname : Host name of a machine.
- date.hour.minuite.second : Date and time when the runmvp command is issued.

For example, the user starts executing sample1.inp on a machine named taro at 15:30:20 on November 15, the name of the output file is as follows.

```
sample1.taro.Nov15.15.30.20
```

After finishing the calculation, the result should be checked. Since the output examples are also attached, the user can compare the result. But the result does not always match with the attached one because of the difference of round-off treatment on each platform etc.

A.3.2 Test Calculation for GMVP

To check the execution of GMVP, examples listed in Table A.5 are prepared in a directory \$MVP_DIR/gmvprun.

Each example solves the following problem.

Table A.5 Sample input data files for GMVP

File	Problem
sample1.inp	Core calculation with lattice geometry.
sample2.inp	Calculation with the DDX-form cross sections.
sample3.inp	Neutron-photon coupled problem.
sample4.inp	Example of the statistical geometry model.
sample5.inp	Example of the FREE-FRAME-LATTICE option.
sample6.inp	Example of tally dimensions SOURCE-REGION and MARKER-REGION.
sample7.inp	Example of a Feynman- α calculation.
sample8.inp	Example of surface crossing and collision estimators.

- Example 1 : This calculation geometry is for the benchmark problem of a 3-dimensional transport calculation. The simplified core is expressed with a lattice geometry. The cross section data to be used are written in the input data.
- Example 2 : This is an example with DDX (double-differential cross sections). The neutron angular/energy distributions after scattering in a cylindrical sample for a mono-energetic incident beam are calculated. (It is a kind of DDX measurements.)
- Example 3 : This calculation geometry describes a phantom for the evaluation of radiation dose rate. The geometry is based on the input data written for the MORSE code. The geometry is expressed with 12 types of bodies available in the MVP/GMVP code. This is a neutron/photon calculation with the Legendre-expansion form of the multigroup cross sections that has 46 and 26 group structure for neutrons and photons, respectively.
- Example 4 : Example of the statistical geometry model. Calculation space except perfect reflector part is filled with graphite-like material to evaluate volume ratio of zones.
- Example 5 : The same doubly bounded assembly model with the FREE-LATTICE-FRAME option as sample5.inp for MVP.
- Example 6 : An example of tally dimensions SOURCE-REGION and MARKER-REGION.
- Example 7 : Noise analysis with Feynman-Y value calculation. The PERIODIC-TIME and ANALOG-FISSION options and tally functions to estimate the Y value are used.
- Example 8 : An example of surface crossing and collision estimators defined in the TALLY data block.

If the environment variables are set correctly, GMVP can be executed only with the following command.

```
rungmvp input-data [ x= / n $\tau$  : xsec-file ]
```

where *input-data* is the file name of a GMVP input data. *x= / n τ : xsec-file* must be specified if the cross section data must be input from the other file. *n* is an I/O unit number for the input of cross section data. It must be 99 for the DDX-form cross sections and otherwise depends on the specification in the

input data[†]. *xsec-file* is the file name of the cross section data. See Section 17.3.2 for the details of the `rungmvp` command.

The commands to execute examples 1 to 3 are as follows. (It is assumed that the current directory is `$MVP_DIR/gmvprun` and prepared cross section files are used.)

Command examples:

```
Example 1:  rungmvp sample1.inp
Example 2:  rungmvp sample2.inp x=/99r:../gmvpplib/ddxj3.IEEE
Example 3:  rungmvp sample3.inp x=/1r:../gmvpplib/n46g26.IEEE
```

The binary-form cross sections processed with GMVPLBCV must be specified. The output file name of the result is determined in the same way as MVP.

See `$MVP_DIR/gmvprun/README` for the details of the examples.

A.4 Program Source Files

Program source files and object files in compilation are stored in the directory `$MVP_DIR/src`. Table A.6 lists the directories for the source files and include files in `$MVP_DIR/src`.

Table A.6 Directories for program source files

Directory	Content
<code>src/mvp</code>	Program sources for MVP.
<code>src/mvp/INC</code>	Include files for MVP.
<code>src/artcore</code>	Program sources for processing of MVP arbitrary temperature libraries.
<code>src/artcore/INC</code>	Include files for processing of MVP arbitrary temperature libraries.
<code>src/gmvp</code>	Program sources for GMVP.
<code>src/gmvp/INC</code>	Include files for GMVP.
<code>src/shared</code>	Program sources shared for MVP and GMVP (A).
<code>src/shared/INC</code>	Include files for shared sources (A).
<code>src/mvpart</code>	Program sources for MVP-ART.
<code>src/utills</code>	Program sources shared for MVP and GMVP (B).
<code>src/tools</code>	Program sources for auxiliary programs.

Shared program sources (A) are required for both MVP and GMVP. Shared program sources (B) are used not only in MVP and GMVP but also in auxiliary programs. These have high portability and can be used in other programs. The auxiliary programs include conversion programs of MVP cross section libraries (NTXT2LB, NLB2TXT), conversion programs of GMVP cross section libraries (GMVPLBCV), a source assimilation program (MVPFAT) etc.

The `src` directory includes directories of `OBJ.*`. These directories are used to store object files generated on each platform where MVP/GMVP can be installed. A `Makefile` generated automatically with the `makemake` command is also stored in this directory. “*” in `OBJ.*` is a system name recognized in MVP/GMVP and is a character string listed in Table A.2.

[†]This input is necessary because the I/O unit of the cross section data can be changed for the GMVP code. The user may think that it is unnecessary to change the I/O unit but currently it is possible.

In compilation, a utility program MVPFAT converts source files in the directories `src/mvp`, `src/gmvp` etc. into those suitable for each platform and then they are compiled[‡]. Since object files for each platform can be stored separately, the source files can be compiled even in the case where the source files are shared by different machines through NFS. Executables are also stored in a directory `$MVP_DIR/bin/$MVPHOST` for each platform.

The commands for the MVPFAT utility are written in a file `ASSIM` in `OBJ.$MVPHOST`. See Appendix F for the details.

A.5 Platform Dependence

We describe some notes and additional information about supported platforms.

A.5.1 Linux

MVP/GMVP runs on Linux environments of Intel x86. Supported compilers are as follows.

- GNU compilers (gfortran or g77, gcc)
- Intel comilers (ifort, icc)
- PGI compilers (pgf90, pgcc)
- Fujitsu comilers(frt, fcc)

If the user uses the dynamic memory capability (DYNAMIC-MEMORY option) with gfortran, its version must be 4.1 or later. If the previous versions of compilers are used, the user has to select the option for static memory allocation in generating `Makefile`.

A.5.2 FreeBSD

MVP/GMVP runs on Intel x 86 environments. For FORTRAN compilers, the user can use the f77 simulator (`$MVP_DIR/etc/_f77.linux`) with f2c and g77 as well as Linux. MakeMacro uses g77 by default.

When using f2c under FreeBSD 2.1.6, the print output is out of position because only 1 digit of the power part of floating numbers is output in E format. It seems that the version of f2c is old or compiler options includes errors. It may not affect calculations but it is recommended that the source of f2c should be re-compiled or the new version of Free BSD including g77 should be used.

A.5.3 CYGWIN

CYGWIN is a system which realizes the UNIX-like environment on MS-Windows with GNU software of Free Software Foundation. The MVP/GMVP code can be compiled and executed with gfortran and gcc in this environment.

A.5.4 MS Windows

The MVP/GMVP code runs under Windows environment (32 bit and 64 bit) on Intel CPUs.

The compiled set is distributed in the different package from the UNIX version. The user should see attached documents for the installation and usage.

[‡]The converted source files are output to the different files, so the original files in `src/mvp`, `src/gmvp` etc. are not changed.

B. CGVIEW

The CGVIEW-SLICE code is a tool to check the MVP or GMVP input data by drawing cross-sectional views of the calculation geometry on a display or output image data files. We call the code just the “CGVIEW” code. The CGVIEW code has the following functions.

- Draw cross-sectional views of spatial regions (INPUT-ZONE, ZONE, REGION) with materials (X Window).
 - Show each region by color.
 - Show boundaries.
- Check calculation geometry.
 - Check undefined spatial regions.
 - Check overlapped spatial regions (INPUT-ZONE, REGION) (OVERLAP command).
 - Check the geometry 3-dimensionally by flying particles randomly in a parallelepiped region (FLOOD command).
- Operation for drawing
 - Set a drawing area automatically.
 - Input commands interactively. (effective only when the shell script “mview3” is used.)
 - Expand/shrink, rotate and move on the drawing area.
 - Show the inside of LATTICES.
 - Zoom in/out on a specified spatial region (INPUT-ZONE, ZONE, BODY).
 - Change drawing colors.
- Check material compositions (only for MVP).
- Output image files.
 - Vector data : Postscript.
 - Bitmap data : GIF, BMP, EPS, etc.

B.1 Output of CGVIEW Code

The CGVIEW code outputs a figure on a display as shown in Fig. B.1. A title is output on the top of the drawing area and the numbers specifying colored materials/regions/zones are output on the right side. Currently, only twelve colors are available and thus the same color may be used for different regions in the case where there are many materials to be shown * .

*The user can change the default color assignment with the MCOLOR command etc.

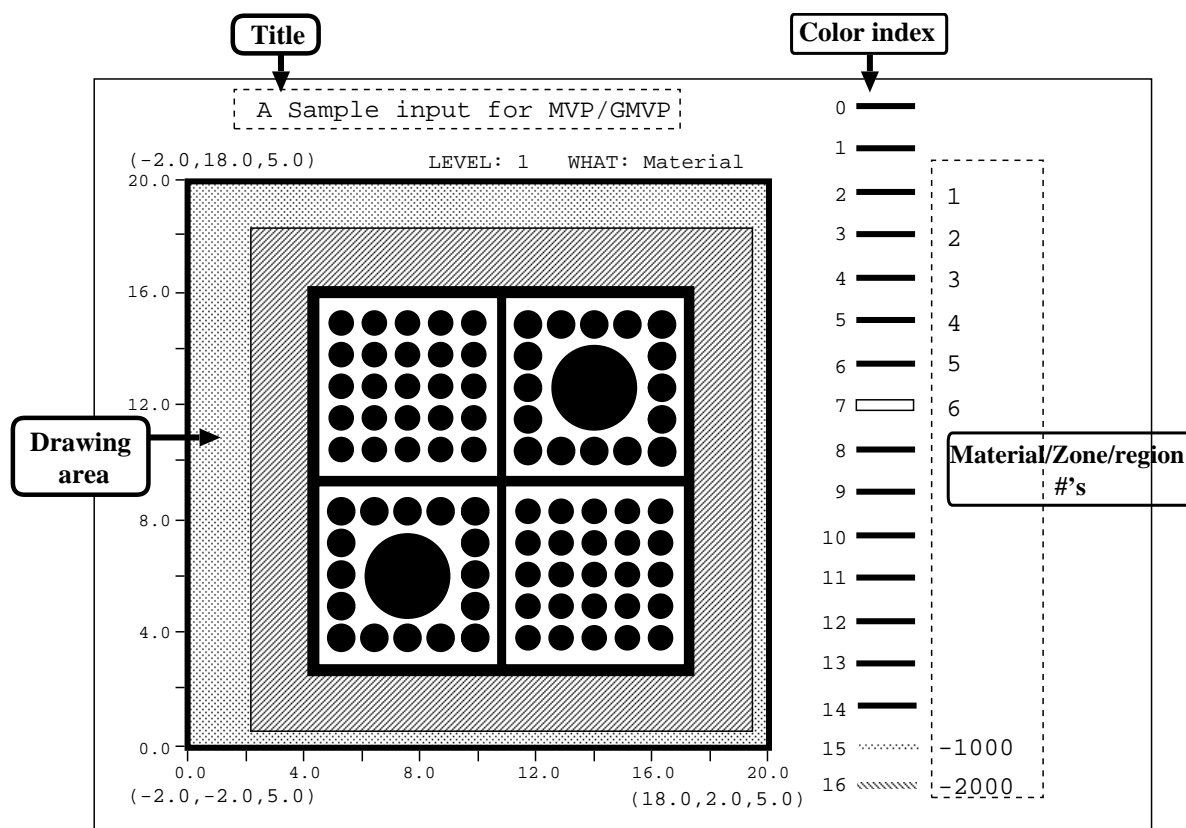


Fig. B.1 Output of the CGVIEW code

B.2 Input Data for CGVIEW Code

The input data for the CGVIEW code can be specified following the termination of the MVP/GMVP input data with “/” (slash). The same input format can be used for the CGVIEW input data.

```

Input data for the MVP/GMVP code
/
CGVIEW input data for the first figure
/
CGVIEW input data for the second figure
/
CGVIEW input data for the third figure
/
....

```

This form of the input data can be used for the MVP/GMVP code as it is. That is, the MVP/GMVP code ignores the CGVIEW input data. The following describes outline and attentions for the CGVIEW input data.

- The CGVIEW input data (also called “CGVIEW commands”) are input with the same form as MVP/GMVP.
 - “data-name(data data ...)”: Specification with the free form of the MVP/GMVP type.

- “/” : Draw a figure with the data input until here.
- In principle, the CGVIEW code can treat both the input data for MVP and GMVP. The user must specify a code name as a command line argument for the CGVIEW executable.
 - `cgview3 c=MVP ...` (for MVP input)
 - `cgview3 c=GMVP ...` (for GMVP input)

In the case of “`cgview3 c=MVP`”, the data in the CROSS SECTION data block for material compositions are read and the composition data can be output by the MAT command of CGVIEW (default). When “`cgview3 c=GMVP`” is specified explicitly, the input of the material compositions is ignored.

The user can use wrapper scripts to execute the CGVIEW code: “`mview3`” for MVP, “`gmview3`” for GMVP.

- A CGVIEW command means the change of a part of specification data and all the data are conserved after a figure is drawn by inputting “/”. Therefore, when the user wants to change a part of the data, the user only has to input a command to change the specification. Suppose that there is the following example. In figure 3, the drawing area is changed by the command “`ZOOM(2)`” (scale factor 2) but the title to be shown is ‘Material’, which is the same as in figure 2. “`WHAT(2)`” is effective in figure 3 and thus regions with the same material are shown with the same color.

```
/* Input data for the MVP/GMVP code */
/
/* CGVIEW input data for figure 1 */
TITLE('Region') /* Title
WHAT(1) /* Show each REGION.
/
/* CGVIEW input data for figure 2 */
TTITLE('Material') /* Change the title
WHAT(2) /* Regions with the same material are shown
/* with the same color.
/
/* CGVIEW input data for figure 3 */
ZOOM(2)
/
...
```

- When the CGVIEW code draws multiple figures on X Window, the user can see the next figure by pressing any key on the screen of CGVIEW. In the interactive mode, if the CGVIEW command “`WAITOFF`” or “`WOFF`” explained later is specified, this process is not necessary and the code begins to draw the next figure immediately after “/” is input.
- The CGVIEW commands can be input with both capital and small characters unlike the MVP/GMVP input data.

Available CGVIEW commands are listed in Table B.1.

Table B.1 CGVIEW commands (1/3)

Command	Operation	Page
BACK or B	Back to the previous figure.	336
BODY(<i>body-ID</i> ...) B(<i>body-ID</i>)	Output information of BODIes for specified ID's.	336
BOUNDARY or BDY	Mark boundaries. Same as STYLE(0 1).	336
CELL(<i>cell-ID</i> [<i>plane</i>])	Show only the specified LATTICE CELL. (<i>cell-ID</i> = 0 : non-cell mode, <i>cell-ID</i> < 0 : inquiry) <i>plane</i> : Direction of the cross-section (1/2/3=xy/xz/yz)	337
CENTER(<i>x y z</i>) C(<i>x y z</i>)	Center the drawing area at coordinates (x, y, z).	337
CHECK	Output the current specification data.	337
DOVERLAP(<i>d</i>) DOVL(<i>d</i>)	Distance of criterion in checking overlapped regions. <i>d</i> > 0 : <i>d</i> *DEPS. <i>d</i> < 0 : ABS(<i>d</i>).	337
FLOOD(<i>mode</i> [<i>num</i>]) FL(<i>mode</i> [<i>num</i>])	Check a geometry with random paths. <i>mode</i> = 1 : Uniform distribution. <i>mode</i> = 2 : Cosine distribution. <i>mode</i> = 0 : Turn off the mode. <i>num</i> : Number of drawing lines.	338
FORGET	Erase memory for adjacent ZONEs.	338
FORWARD or F	Forward to the next figure.	338
GRANGE(<i>x1 x2</i> <i>y1 y2 z1 z2</i>)	Range of coordinates for the check with random paths.	339
HELP or H HELP2 or H2 ...	List the CGVIEW commands. (Show page 2 or later, page 3 or later ...)	339
HELPW or HW	List the CGVIEW commands on the screen.	339
IZCOLOR(<i>izone color</i>) IZC(<i>izone color</i>)	Change INPUT-ZONE colors.	339
IZCR	Set back INPUT-ZONE colors the default set.	339
LEVEL(<i>nest-level</i>)	Level of LATTICEs to be drawn. (The boundary of SUB-FRAMEs is drawn if <i>nest-level</i> <0 and SUBF is specified.)	340
LINES(<i>p</i>)	Density of drawing lines (same as <i>p</i> in PAPER).	340
MAT(<i>material-ID</i> ...)	Output information of material compositions. (only for MVP)	341
MCOLOR(<i>mat color</i>) MC(<i>mat color</i>)	Change material colors.	341
MCR	Set back material colors the default set.	341
MOVE(<i>dx dy dz</i>)	Move the drawing area. (Relative to the screen size)	341
MOVEA(<i>dx dy dz</i>)	Move the drawing area. (cm)	341
OVERLAP(<i>flag</i>) OVL(<i>flag</i>)	Check overlapped regions. <i>flag</i> > 0 : Check, <i>flag</i> = 0 : Does not check	342
PAINT or PNT	Color regions. Same as STYLE(0 0).	342
PAPER(<i>cx cy cz</i> <i>ax ay az</i> <i>bx by bz p</i>)	Position and direction of the drawing area, density of drawing lines. <i>cx, cy, cz</i> : lower left corner <i>ax, ay, az</i> : X-axis vector	342

Table B.1 CGVIEW commands (2/3)

Command	Operation	Page
	bx, by, bz : Y-axis vector $p > 0$: lines per cm $p < 0$: lines on every $\text{abs}(p)$ pixels.	
PRINT([<i>cmd-no</i> [<i>seq</i>]])	Save the display image as a bitmap file.	343
QUIT	Quit CGVIEW.	345
RCOLOR(<i>region color</i>) RC(<i>region color</i>)	Change REGION colors.	345
RCR	Set back REGION colors the default set.	345
REDRAW or RD	Re-draw a figure on X-Window.	345
ROT(d [x [y]])	Rotate the drawing area. (The rotation axis is perpendicular to the screen.) d : Rotation angle counter-clockwise (degree). x, y : Position of the rotation axis (relative to the drawing area, default 0.5, 0.5).	346
ROTX(d [<i>pos</i>])	Rotate the drawing area. (The rotation axis is the transverse direction of the screen.) d : Rotation angle counter-clockwise (degree). pos : Vertical position of the rotation axis (relative to the drawing area, default 0.5).	346
ROTY(d [<i>pos</i>])	Rotate the drawing area. (The rotation axis is the vertical direction of the screen.) d : Rotation angle counter-clockwise (degree). pos : Transverse position of the rotation axis (relative to the drawing area, default 0.5).	346
SCAN(<i>sx sy</i>)	Scan mode for drawing lines. sx : X-SCAN MODE(0/1/2/3 = none/+y/-y/+y/-y/). sy : Y-SCAN MODE(0/1/2/3 = none/+x/-x/+x/-x/).	347
SIZE(<i>xsize ysize</i>) XMAX(<i>xsize ysize</i>)	Size of the drawing area. $xsize$: X-Axis length (cm). $ysize$: Y-Axis length (cm).	348
STYLE(<i>s1</i> [<i>s2</i>])	Style of drawing lines. $s1$: 0 = paint regions, 1 = draw boundaries only. $s2$: 0 = color regions, 1 = mark boundaries.	348
SUBF and NOSUBF	The boundaries of SUBFRAMEs are drawn/not drawn when BOUNDARY is specified.	349
TITLE('title string')	Title.	349
WAITOFF or WOFF	Make any key entry unnecessary to draw the next figure in X-Window.	349
WAITON or WON	Make any key entry necessary to draw the next figure in X-Window.	349
WHAT(<i>what</i>)	Flag of the drawing unit. $what = -1$: ZONE (ZONE specified in the code) $what = 0$: INPUT-ZONE $what = 1$: REGION	349

Table B.1 CGVIEW commands (3/3)

Command	Operation	Page
	<i>what</i> = 2 : Material	
XY YX XZ ZX YZ ZY	Position and direction of the drawing area set automatically. (XY-, XZ-, YZ-planes)	350
ZCOLOR(<i>zone color</i>) ZC(<i>zone color</i>)	Change ZONE colors.	350
ZCR	Set back ZONE colors the default set.	350
ZONE(<i>zone ...</i>) Z(<i>zone ...</i>)	Output information of INPUT-ZONES or ZONES for positive or negative ZONE numbers, respectively.	350
ZOOM(<i>f</i> [<i>cx</i> [<i>cy</i>]])	Zoom in/out the drawing area. <i>f</i> > 1 : zoom in. <i>f</i> < 1 : zoom out. <i>cx</i> , <i>cy</i> : zooming center (default 0.5, 0.5).	350
ZOOMBODY(<i>body-ID</i> [<i>plane</i>]) ZB(<i>body-ID</i> [<i>plane</i>])	Show the figure where a specified BODY is centered. <i>plane</i> : Direction of the cross-section (1/2/3=xy/xz/yz).	351
ZOOMZONE(<i>zID</i> [<i>plane</i>]) ZZ(<i>zID</i> [<i>plane</i>])	Show the figure where a specified INPUT-ZONE (or ZONE) is centered. The ZONE is centered for <i>zID</i> < 0. <i>plane</i> : Direction of the cross-section (1/2/3=xy/xz/yz).	351
ZSEARCH(<i>flag</i>) ZS(<i>flag</i>)	Order of zone search for each drawing line. <i>flag</i> = 0 : Normal direction. <i>flag</i> = 1 : Reverse direction. <i>flag</i> = 2 : Change direction line by line.	352

B.3 CGVIEW Commands

BACK or B

■ Format ■

BACK or B

■ Description ■

Back to the previous figure.

BODY

■ Format ■

BODY(*body-ID* [...])

■ Description ■

Output the information of BODIes specified by their ID.

BOUNDARY or BDY

■ Format ■

BOUNDARY or BDY

■ Description ■

Mark boundaries. The command is the same as STYLE(0 1).

CELL

■ Format ■

CELL(*cell-ID* [*plane*])

■ Description ■

Show only the specified LATTICE CELL. If *cell-ID* = 0, the CELL is not displayed but the whole geometry is displayed. For *cell-ID* < 0, the list of CELLS are output to the standard output.

plane indicates the direction of the cross-section. It can be omitted. For *plane* = 1, 2, 3, the cross-sectional plane corresponds to the xy-, xz- and yz-plane, respectively. If *plane* is omitted, the cross-sectional plane is set to the xy-plane.

CENTER

■ Format ■

CENTER(*x y z*)

■ Description ■

Center the drawing area at coordinates (*x*, *y*, *z*).

CHECK

■ Format ■

CHECK

■ Description ■

Output the current parameters such as the position of the drawing area, the drawing mode, the contents of the figure on the standard output unit. The code begins to output the parameters immediately after it reads the CHECK command.

DOVERLAP or DOVL

■ Format ■

DOVERLAP(*d*) or DOVL(*d*)

■ Description ■

Distance of criterion in checking overlapped regions or multiply defined materials with OVERLAP command.

- $d > 0$: $d \cdot \text{DEPS}$ (cm). DEPS is the infinitesimal distance specified in the MVP/GMVP input.
- $d < 0$: The absolute value of d is the distance of criterion (cm).

FLOOD or FL

■ Format ■

FLOOD (*mode* [*num*]) or FL (*mode* [*num*])

■ Description ■

Check a geometry with random paths. The drawing lines are not generated along the vertical or transverse direction but are drawn from arbitrary positions along the arbitrary directions as shown in Fig. B.2. Since CGVIEW checks the geometry 3-dimensionally, it is desired to use this command for all MVP/GMVP input data.

mode is the parameter to specify the distribution of initial positions for the lines or return to the normal mode.

- *mode* = 1 : Uniform distribution.
- *mode* = 2 : Cosine distribution for each axis (*x*-, *y*-, *z*-axes). The lines are drawn mainly from the center of the drawing region.
- *mode* = 0 : Quit the FLOOD mode and return to the normal mode.

num is the number of the drawing lines. (It can be omitted.) The default value is 2,000. The same values of *mode* and *num* are used until they are changed.

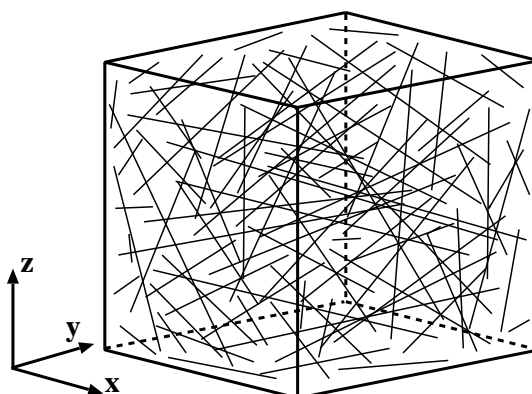


Fig. B.2 Check a geometry with randomly generated lines - FLOOD

FORGET

■ Format ■

FORGET

■ Description ■

Erase the record of adjacent ZONEs currently in memory.

FORWARD or F

■ Format ■

FORWARD or F

■ Description ■

Forward to the next figure. The command is effective as many times as the BACK command is input.

GRANGE

■ Format ■

GRANGE (*x1 x2 y1 y2 z1 z2*)

■ Description ■

Specify the range of coordinates for the initial positions of the drawing lines in checking with the FLOOD command. (See Fig. B.2.) The range is determined from BODY data used for ZONEs by default and is determined automatically when CELL, XY, YX, XZ, ZX, YZ, ZY, ZOOMZONE or ZOOMBODY is used. Therefore, there are many cases where it is unnecessary to specify the GRANGE data explicitly.

HELP or H

■ Format ■

HELP or H

■ Description ■

Output the explanation of the CGVIEW commands on the screen. The explanation is output immediately after the CGVIEW code reads the HELP command. The explanation is delimited by the unit (page) of a standard screen (80×24 characters). HELP2 or H2 outputs page 2 or later.

The CGVIEW commands on the screen (ZOOM, MOVE etc.) are displayed by the HELPW or HW command.

IZCOLOR or IZC

■ Format ■

IZCOLOR (*izone color*) or IZC (*izone color*)

■ Description ■

Change INPUT-ZONE colors. The INPUT-ZONE colors are specified with the numbers in the “Color index” shown in Fig. B.1.

IZCR

■ Format ■

IZCR

■ Description ■

Set back INPUT-ZONE colors the default set.

LEVEL

■ Format ■

LEVEL(*nest-level*)

■ Description ■

Specify the LATTICE level to be drawn for the LATTICE geometry.

Suppose that we think the case shown in Fig. B.3. In this case, there is a LATTICE (LATTICE at level 1) in the CELLS of LATTICE at level 0 and furthermore there is another LATTICE (LATTICE at level 2) in the CELLS of LATTICE at level 1. For LEVEL(0), only the frame of the LATTICE at level 0 is drawn. For LEVEL(1), the frame of the LATTICE at level 1 is also drawn. For LEVEL(2), the geometry is drawn in detail.

The default value for LEVEL is the maximum level of LATTICES in the MVP/GMVP input data.

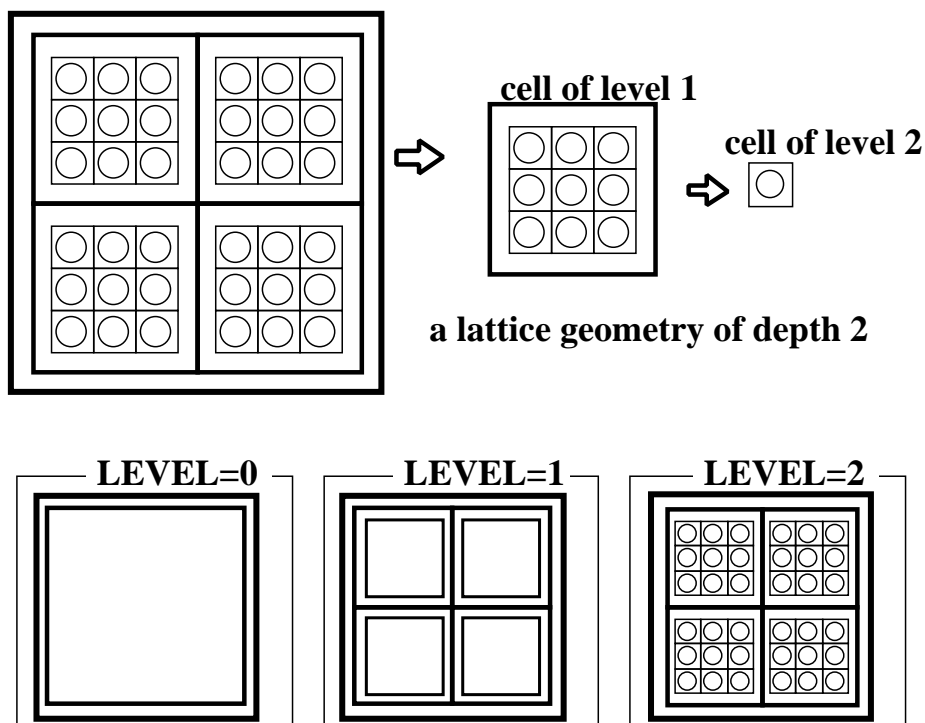


Fig. B.3 Specification of the LATTICE level to be drawn - LEVEL

LINES

■ Format ■

LINES(*p*)

■ Description ■

Specify the interval of the drawing lines. If *p* is a positive value, it means the number of the lines per 1 cm in the real space. If *p* is a negative value, the absolute value indicates the interval of the lines in pixel on the screen (X Window). *p* can be also specified with the PAPER command.

MAT

■ Format ■

MAT (*material-ID ...*)

■ Description ■

Output the information of material compositions on the standard output unit. The command is effective only for the MVP input data (“c=MVP” is given in the command arguments).

MCOLOR or MC

■ Format ■

MCOLOR (*mat color*) or MC (*mat color*)

■ Description ■

Change material colors. The material colors are specified with the numbers in the “Color index” shown in Fig. B.1.

MCR

■ Format ■

MCR

■ Description ■

Set back material colors the default set.

MOVE or M

■ Format ■

MOVE (*dx dy dz*) or M (*dx dy dz*)

■ Description ■

Move the drawing area. *dx* and *dy* are the relative values to the drawing area.

move(1 0 0) /* Display the neighborhood in the X-direction

dz is the relative value to the width of the drawing area. For $dz > 0$, the drawing area is moved upward. The commands MX(*dx*), MY(*dy*) and MZ(*dz*) specify the movement in the X-, Y- and Z-direction, respectively.

MOVEA or MA

■ Format ■

MOVEA (*dx dy dz*) or MA (*dx dy dz*)

■ Description ■

Same as MOVE but *dx*, *dy* and *dz* specify the absolute moving distance in centimeter. The commands MXA(*dx*), MYA(*dy*) and MZA(*dz*) specify the movement in the X-, Y- and Z-direction, respectively.

OVERLAP or OVL

■ Format ■

OVERLAP(*flag*) or OVL(*flag*)

■ Description ■

Check overlapped regions. When this command is specified, the spatial positions and zones of the overlapped regions are displayed to the standard output.

If *flag* > 0, CGVIEW checks the overlapped regions. If *flag* = 0, then CGVIEW does not check. The default value is *flag* = 1.

CGVIEW judges that there is an overlapped region when the distance from the boundary between different-material ZONES to the boundary of the ZONE to be entered next is larger than the specified value. See DOVERLAP for the criterion. Therefore, there is a possibility that the perfectly overlapped region cannot be detected like the following example. (Partially overlapped regions can be detected.)

```
/** Body input **/
RPP( 10  0.0  10.0   0.0 5.0   0.0 20.0 )
RPP( 20  0.0  10.0   0.0 5.0   0.0 20.0 )
...
END
/** Zone input **/
ZONE1: REGION1 : 1 : 10
ZONE2: REGION2 : 2 : 20
...
```

If the drawing lines are drawn with the FLOOD command, the overlapped regions are strictly checked at the initial positions regardless of the OVERLAP command. Therefore, it is desired to use the FLOOD command for checking calculation geometry.

PAINT or PNT

■ Format ■

PAINT or PNT

■ Description ■

Color regions. The command is the same as STYLE(0 0).

PAPER

■ Format ■

PAPER(*cx cy cz ax ay az bx by bz p*)

■ Description ■

Specify the drawing area and the density of drawing lines. (*cx cy cz*) are the coordinates at the lower left corner of the area, (*ax ay az*) and (*bx by bz*) are the vectors to define the transverse and vertical directions of the area, respectively. Since the size of the area is specified with the SIZE or XMAX command described below, only the direction of these vectors has meaning. See Fig. B.4 for the drawing area.

The CGVIEW code draws figures by tracing a lot of drawing lines displayed on the screen and checking the ZONE/REGION/material numbers on the lines. *p* specifies the interval of the lines. If *p* is a positive value, it means the number of lines per 1 cm in the real space. If *p* is a negative value, the

absolute value indicates the interval of the lines in pixel on the screen (X-Window). p can be specified with the LINES command described above.

Unless any PAPER data is input, the direction and size of the drawing area are set as follows.

- Direction : parallel to the X-Y plane.
- Size : The CGVIEW code estimates the space which includes all ZONES from the BODY of the outermost frame (the ZONE of level “0” for a LATTICE geometry). The code draws the cross-sectional view of the space on the X-Y plane at the central position along the Z-direction (vertical direction).

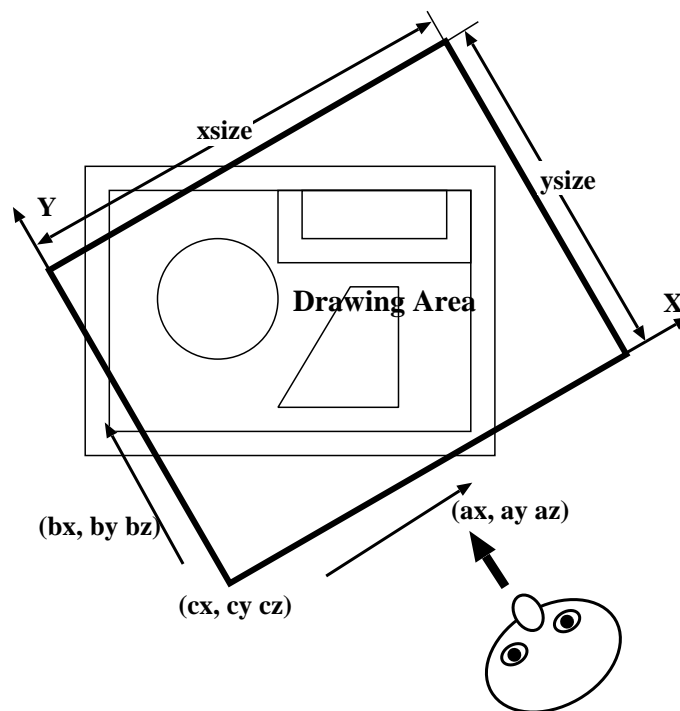


Fig. B.4 Specification of the drawing area for CGVIEW - PAPER, SIZE

PRINT

■ Format ■

```
PRINT( [ cmd-no [seq] ] )
```

■ Description ■

Save the display image as a bitmap file. Though there is another method to save the display image as a file, where a Postscript or EPS file can be specified as the command-line option of CGVIEW (See Section B.4.), no image is displayed on the screen and the size of the output file tends to be large. The PRINT command can save the output image on X-Window to a file as a bitmap image by using one of external image output commands prepared by the user.

cmd-no is the image output command number. *seq* is the sequential number (for each output command) to be added to the image files. If *cmd-no* and *seq* are omitted, CGVIEW uses the first command

and adds sequential numbers automatically. If *seq* is omitted, CGVIEW determines the number of image output for command *cmd-no*.

The user must pay attention to the following.

- CGVIEW does not capture an image and output it to a file for itself but other programs invoked by CGVIEW do. (Examples: *xwd*, *netpbm*, *ImageMagick* etc.) The programs must be prepared to use the PRINT command.
- In general, the graphics outputs with the PRINT command have low-resolution images compared with the images of Postscript or EPS obtained with the command-line option of CGVIEW.

More than one set of the external commands and their option to capture an image can be specified. The user must specify the sets with environmental variables *CGV_PRINT1*, *CGV_PRINT2*, *CGV_PRINT3*, *CGV_PRINT4*.

For example,

```
setenv CGV_PRINT2 'xwd -id %%w | xwdtopnm | ppmtogif > %%f%%p.gif'
```

This specifies operations to output an image file with the second command (*cmd-no* = 2) for the PRINT command. First, the *xwd* command captures the image and *xwdtopnm* of the *netpbm* tool transforms it into the *pnm* format and *ppmtogif* transforms in to the *gif* format. *%%w* etc. are the parameters to be replaced in CGVIEW. The meaning of the parameters is as follows.

- *%%w* : Window ID of the current image.
- *%%f* : Arbitrary character string defined by the user. The user can distinguish the output files by naming like “*cgvimage*.gif*”. *%%f* can be specified with the environmental variable *CGV_PRBASE*.

```
setenv CGV_PRBASE cgvimage
```

- *%%p* : Number added by CGVIEW for the multiple image outputs by the external command with the same command number *cmd-no*. The number can be specified with the second parameter *seq* of the PRINT command.

As one finds from the above description, the PRINT command can utilize any external program that can capture an image and is invoked with the window ID of the image. The following are the examples to use some tools.

- *xwd* only

```
setenv CGV_PRINT1 'xwd -id %%w > %%f%%p.xwd'
```

This means that a standard command for X-Window *xwd* (X Window dump) is specified as the first PRINT command. This command is used unless the environmental variable *CGV_PRINT** is not specified.

- Combination of *xwd* and *netpbm* tools

Netpbm is a set of tools to convert the image formats.

```
setenv CGV_PRINT2 'xwd -id %%w | xwdtopnm | ppmtogif > %%f%%p.gif'
setenv CGV_PRINT3 'xwd -id %%w | xwdtopnm | ppmtotiff > %%f%%p.tiff'
setenv CGV_PRINT4 'xwd -id %%w | xwdtopnm | ppmtobmp > %%f%%p.bmp'
```

The example shows that image data of the GIF, TIFF and BMP format are saved, respectively.

- `import` command of ImageMagick

ImageMagick is a tool to edit and convert images. The `import` command of ImageMagick can be used.

```
setenv CGV_PRINT2 'import -compression LZW -window %w %%f%%p.gif'
setenv CGV_PRINT3 'import -compression LZW -window %w %%f%%p.eps'
setenv CGV_PRINT4 'import -window %w %%f%%p.eps2'
```

These commands save images of the GIF(LZW compression), EPS(LZW compression) and EPS2 (EPS based on Postscript Level II), respectively[†]. The `import` command supports various image formats.

QUIT or EXIT

■ Format ■

QUIT or EXIT

■ Description ■

Quit CGVIEW.

RCOLOR or RC

■ Format ■

RCOLOR(*region color*) or RC(*region color*)

■ Description ■

Change REGION colors. The REGION colors are specified with the numbers in the “Color index” shown in Fig. B.1.

RCR

■ Format ■

RCR

■ Description ■

Set back REGION colors the default set.

REDRAW or RD

■ Format ■

REDRAW or RD

[†]On free UNIX platforms such as many distributions of Linux, there are some cases where ImageMagick does not support the LZW compress because of a patent. In this case, the user must use the option “`-compression RunlengthEncoded`” for EPS or compile ImageMagick with LZW enabled by himself.

■ Description ■

When WAITOFF is specified in the interactive mode, a displayed image may be erased by an overlapped window screen. (So-called “backing store” cannot be used in the case of X-Window.) This command re-draws the displayed figure immediately.

ROT

■ Format ■

ROT (d [x [y]])

■ Description ■

Rotate the drawing area by the rotation angle of d degree counter-clockwise. The rotation axis is perpendicular to the drawing area. The drawing object is not rotated against the drawing area but the area is rotated against the fixed object. See Fig. B.5 for the rotation of the drawing area. x and y specify the position of the rotation axis and are relative values to the drawing area. They can be omitted. The default value is 0.5 for x and y .

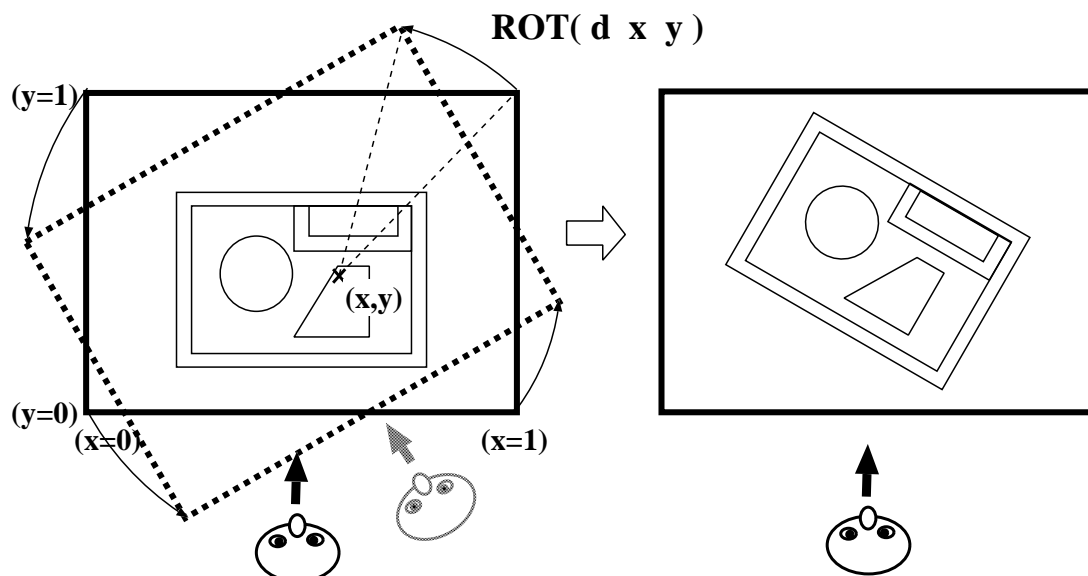


Fig. B.5 Rotation of the drawing area around the rotation axis perpendicular to the screen - ROT

ROTX or ROTY

■ Format ■

ROTX (d [pos]) or ROTY (d [pos])

■ Description ■

Rotate the drawing area around the axis parallel to the transverse (X) or vertical (Y) direction by the rotation angle of d degree counter-clockwise. See Fig. B.6 for the rotation of the drawing area. pos is the transverse or vertical position of the rotation axis and the relative value to the drawing area. It can be omitted. The default value is 0.5.

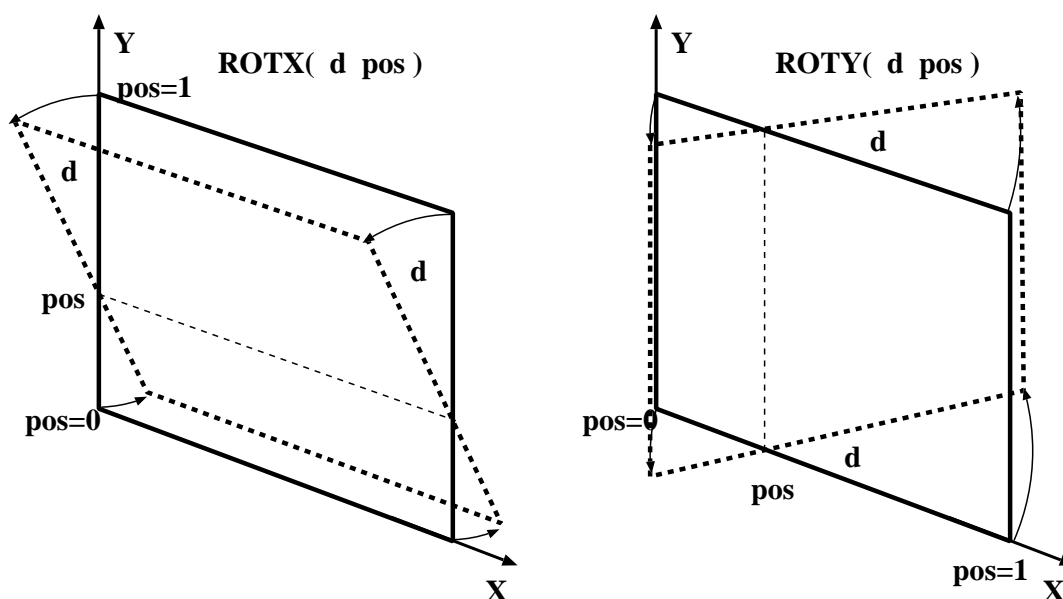


Fig. B.6 Rotation of the drawing area around the rotation axis parallel to the screen - ROTX, ROTY

SCAN

■ Format ■

SCAN(*sx sy*)

■ Description ■

Specify the drawing direction. See Fig. B.7 for the specification of the drawing direction. *sx* and *sy* are the input parameters in the X- and Y-coordinates, respectively. The “scan in the X-coordinate” is the operation to draw lines along the Y-direction (vertical direction on the screen) at the multiple X-coordinates. The “scan in the Y-coordinate” is the operation to draw lines along the X-direction (transverse direction on the screen) at the multiple Y-coordinates.

sx and *sy* can be specified as follows. The default values are *sx*=0, *sy*=3.

- *sx*=0 : Not scan in the X-coordinate.
- *sx*=1 : Scan in the X-coordinate and draw the lines from bottom to top.
- *sx*=2 : Scan in the X-coordinate and draw the lines from top to bottom.
- *sx*=3 : Scan in the X-coordinate and draw line by line from bottom to top/from top to bottom.
- *sy*=0 : Not scan in the Y-coordinate.
- *sy*=1 : Scan in the Y-coordinate and draw the lines from left to right.
- *sy*=2 : Scan in the Y-coordinate and draw the lines from right to left.
- *sy*=3 : Scan in the Y-coordinate and draw line by line from left to right/from right to left.

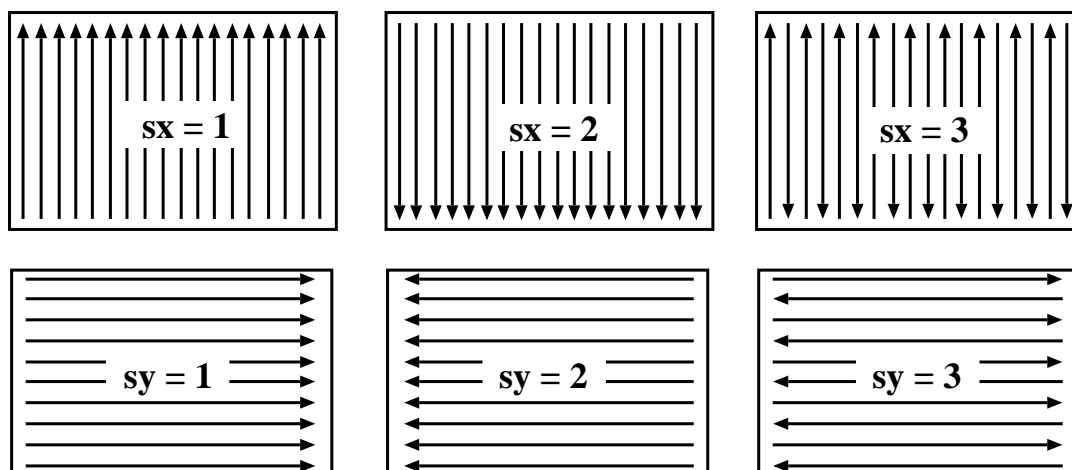


Fig. B.7 Specification of drawing lines for CGVIEW - SCAN(sx sy)

SIZE or XMAX

■ Format ■

SIZE(*x1 y1*) or XMAX(*x1 y1*)

■ Description ■

Specify the size of the drawing area in the transverse (X) and vertical (Y) direction. The unit is centimeter.

STYLE

■ Format ■

STYLE(*s1* [*s2*])

■ Description ■

Specify the drawing style. *s2* can be omitted.

- $s1 = 0$: Draw the distribution of ZONES/REGIONS/materials along the drawing lines.
- $s1 = 1$: Try to draw the boundaries of ZONES/REGIONS/materials as the continuous curves on the basis of the information of ZONES/REGIONS/materials along the drawing lines.

If $s1=0$, $s2$ is effective.

- $s2 = 0$: Show the distribution of ZONES/REGIONS/materials with colors.
- $s2 = 1$: Draw only points at the boundaries of ZONES/REGIONS/materials.

There are many cases where a line drawing is incomplete with $s1=1$. Currently, the user should specify $s1=0$, $s2=1$ and LINES(-1) for the line drawing.

SUBF and NOSUBF**■ Format ■**

SUBF and NOSUBF

■ Description ■

The boundaries of SUBFRAMES are drawn/not drawn when BOUNDARY is specified.

TITLE**■ Format ■**

TITLE(' *title string* ') or TITLE(" *title string* ")

■ Description ■

A character string parenthesized by " " (double quotation marks) or ' ' (single quotation marks) is specified. The string is the title of a figure and is displayed on the top of the figure. Unless any title is input, the first line of the title specified in the MVP/GMVP input data.

WAITOFF or WOFF**■ Format ■**

WAITOFF or WOFF

■ Description ■

When figures are displayed on X-Window, the user must press any key to draw the next figure. However, if this command is specified, the next figure is drawn immediately after "/" is found in the CGVIEW input. This command is useful in the interactive mode of CGVIEW.

WAITON or WON**■ Format ■**

WAITON or WON

■ Description ■

Contrary to WAITOFF or WOFF, the user must press any key to draw the next figure when this command is specified.

WHAT or SPTYP**■ Format ■**

WHAT(*what*) or SPTYP(*what*)

■ Description ■

Flag of the drawing attribute. In the case of the paint mode (STYLE(0)), regions are colored with the following attribute. In the case of the line mode (STYLE(1) or STYLE(0 1)), boundaries are drawn with the following attribute.

- *what* = -1 : ZONE (ZONE specified in the code)

- *what* = 0 : INPUT-ZONE
- *what* = 1 : REGION
- *what* = 2 : Material

XY, YX, XZ, ZX, YZ or ZY

■ Format ■

XY, YX, XZ, ZX, YZ or ZY

■ Description ■

Specify the direction of the drawing area. Specifying XY, XZ and YZ, the area is set in the XY-, XZ- and YZ-plane, respectively. The size and position of the area are defined automatically from the geometry data of MVP/GMVP in the same way as the default values of PAPER. YX, ZX and ZY set the transverse/vertical axes of the area to the Y-/X-axes, Z-/X- axes and Z-/Y-axes, respectively.

ZCOLOR or ZC

■ Format ■

ZCOLOR(*zone color*) or ZC(*zone color*)

■ Description ■

Change ZONE colors. The ZONE colors are specified with the numbers in the "Color index" shown in Fig. B.1.

ZCR

■ Format ■

ZCR

■ Description ■

Set back ZONE colors the default set.

ZONE or Z

■ Format ■

ZONE(*zone* [...]) or Z(*zone* [...])

■ Description ■

Output the information of INPUT-ZONES or ZONES on the standard output unit. The information of INPUT-ZONES or ZONES is output for the positive or negative value, respectively.

ZOOM or ZM

■ Format ■

ZOOM(*zm* [x [y]]) or ZM(*zm* [x [y]])

■ Description ■

Zoom in/out from the drawing area. If zm is larger than unity, the user zooms in the area as shown in Fig. B.8. If zm is positive and smaller than unity, the user zooms out. If zm is a negative value, the magnifying power is $1/|zm|$. x and y are the relative positions of the zooming center on the drawing area, which can be omitted. The default value is $x = y = 0.5$.

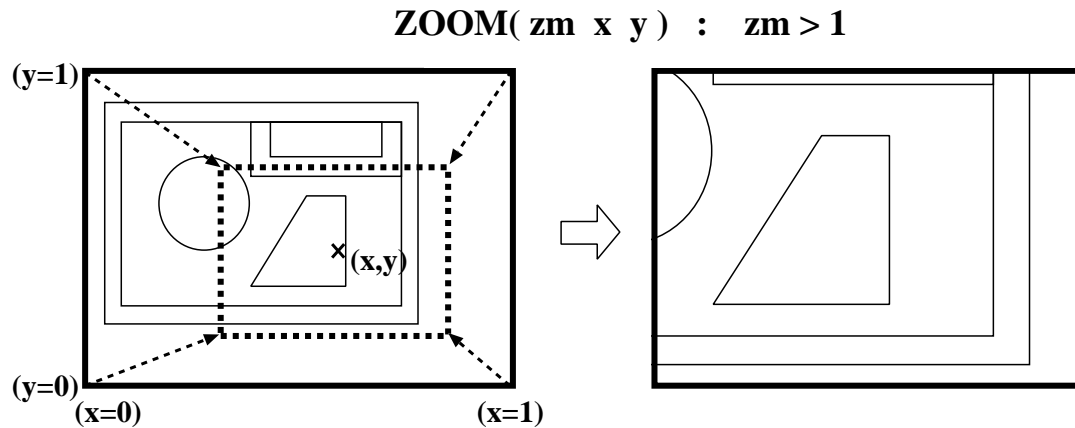


Fig. B.8 Zooming in/out a drawing area - ZOOM

ZOOMBODY or ZB

■ Format ■

ZOOMBODY (*body-ID* [*plane*]) or ZB (*body-ID* [*plane*])

■ Description ■

Show the figure where a specified BODY is centered.

plane indicates the direction of the cross-section. For *plane* = 1, 2 and 3, the cross-sectional plane corresponds to the xy-, xz- and yz-plane, respectively. If *plane* is omitted, the cross-sectional plane is set to the xy-plane.

ZOOMZONE or ZZ

■ Format ■

ZOOMZONE (*zID* [*plane*]) or ZZ (*zID* [*plane*])

■ Description ■

Show the figure where a specified INPUT-ZONE (or ZONE) is centered. For $zID > 0$, the INPUT-ZONE is centered. For $zID < 0$, the ZONE is centered.

plane indicates the direction of the cross-section. For *plane* = 1, 2 and 3, the cross-sectional plane corresponds to the xy-, xz- and yz-plane, respectively. If *plane* is omitted, the cross-sectional plane is set to the xy-plane.

ZSEARCH or ZS

■ Format ■

ZSEARCH(*flag*) or ZS(*flag*)

■ Description ■

Order of zone search for each drawing line. When a particle reaches the boundary of a spatial ZONE, the ZONE to be entered next is searched by MVP/GMVP in the order of ZONEs defined in the input data[‡].

CGVIEW can search for the next ZONE in the reverse order of the MVP/GMVP processing with the ZSEARCH command. That is, CGVIEW can search for the ZONE from the ZONE defined last in the input data. In some cases, the inconsistent geometry model can be found with this procedure. (The displayed geometry with the normal-order search is different from that with the reverse-order search, etc.)

- *flag* = 0 : Normal direction. Same as MVP/GMVP.
- *flag* = 1 : Reverse direction.
- *flag* = 2 : Change direction line by line.

Unless the ZSEARCH command is specified, the default value is *flag* = 2. The user should use the FLOOD command to check the geometry with obvious inconsistency.

B.4 Script to execute CGVIEW

The CGVIEW executable `cgview3` can be run with some options but it is troublesome to specify the options. So, the shell script `mview3` is prepared to execute CGVIEW simply. For GMVP input data, the shell script `gmview3` is available with the same command options. The usage of `mview3` is as follows (same for `gmview3`).

```
mview3 [-help|-h] [-man|-m] [-d device] [-o name]
        [-e command] [-p] [-db=debugger] [files]

-help or -h           : show this help message.

-d device or -ddevice : output device.
    device = x or X    - X-Window display (default)
    device = p[s]      - Postscript
    device = e[ps] or t[ex] - EPS for TeX

-o name or -oname     : graphic output file name for non-display device

-e command or -ecommand : command to run CGVIEW (default is 'cgview3')

-p : enable additional input from tty (interactive mode)
-s[=file] : save input & additional input on file
           (default file is $$newinp )
-w       : do not append WOFF to input
-gmvp    : input is for GMVP (MVP is default)
-l[=logfile] : save output log in interactive mode.
```

[‡]There is a case where the order of zone search is not always specified one because MVP/GMVP utilizes the geometry information learned in the previous path tracing.

```
(default file is $$ .cgvlog )
files : MVP/GMVP input file ( + CGVIEW-command file if necessary)
```

To invoke CGVIEW in the interactive mode, the user should issue the command as follows.

```
mview3 -p input-data
```

In this case, the user can input the CGVIEW commands from a terminal after CGVIEW executes all the CGVIEW commands in the input data and draws figures. When “files” is omitted, the user can input the MVP/GMVP data from standard input. However, if `-p` option is used, the file name cannot be omitted.

B.5 Problems in Using CGVIEW and Solutions

The following lists possible problems in using CGVIEW and some solutions to them.

- **[Problem]** When a drawing line is just on the boundary surface of spatial ZONEs by accident, CGVIEW may be unable to draw the boundary.

[Solution] This is an inevitable error for current CGVIEW. In this case, the user should move the drawing area slightly. For example, `MZ (0.0001)` etc.

- **[Problem]** Spatial regions overlapped with completely the same geometries may not be detected.

[Solution] CGVIEW can check partially overlapped regions but cannot detect fully overlapped regions where more than one geometry are completely the same. (See the description of the OVERLAP command.) There is a case where the overlapped region is found when the edge of the drawing area belongs to the overlapped region and the ZSEARCH mode is 2. When the drawing lines are drawn with the FLOOD command, the overlapped regions are checked at the initial positions strictly. Therefore, it is desired to use the FLOOD command for checking calculation geometry.

- **[Problem]** The number of displayed colors is reduced.

[Solution] This situation happens when the number of colors available on X-Window display is small and other program that requires a lot of colors is running. This is a problem of the display library which CGVIEW uses. The user should terminate the program or invoke X-Window with more colors.

- **[Problem]** When a figure is drawn with the FLOOD command, too many random lines are drawn in a window.

[Solution] There is inconsistency between the drawing area and the size of the spatial region in which the lines are generated with the FLOOD command. The user can adjust them with the CELL, ZOOMZONE, ZOOMBODY commands etc.

- **[Problem]** Displaying a figure without the PAPER command, a lot of “lost particles” occur or the figure at an improper position is drawn.

[Solution] In this case, the user should modify the geometry data so that the position of the drawing area can be set automatically.

- The outside of calculation geometry should be defined. When the whole calculation geometry is surrounded by spatial regions with fictitious materials for boundary conditions, the region should be defined as infinite. (define like all the outside of a single BODY etc.)

- A BODY with a simple shape should be used for the outermost frame of calculation geometry such as RPP, CYL, SPH etc.
- **[Problem]** Once the wrong command is input in the interactive mode, the following commands cannot be input correctly.
[Solution] The user should try to type the right parenthesis “)” and then enter the return key. Sometimes the user can return to the normal situation.

C. MVP-ART

MVP-ART is an auxiliary code to generate cross section data at arbitrary temperatures. Resonance cross sections are generated by Doppler-broadening. $S(\alpha, \beta)$ data and unresolved resonance probability tables are generated by interpolation between the given temperatures. The usage of the MVP-ART code is described in the following.

C.1 Input Data for MVP-ART

To execute the MVP-ART code, the following input data must be input in the free form.

Data #1 IVPOP, IDBG (I4, I4)

IVPOP Option to use vectorized routines for Doppler broadening.
 = 1 Use.
 = 0 Not use. (should be 0 for workstations.)

IDBG Debug option.
 = 1 Not output debug messages.
 = 0 Output debug messages.

Data #2 INIDX (A)

INIDX Path to the index file for the MVP library (including the file name). The input data must start from the first column. The character-string length must be not more than 144.

Data #3 TAGP, PATH (A, A)

TAGP = 'path'. The tag 'path' must start from the first column.
 PATH Directory where generated cross section files are stored. PATH must end with a slash(/). The character-string length including the tag must be not more than 144.

Data #4 TAGI, OUTIDX (A, A)

TAGI = 'index'. The tag 'index' must start from the first column.
 OUTIDX Index file name for generated cross section data. The character-string length including the tag must be not more than 144.

The user input Data #5 and #6 repeatedly as necessary. If the first column is '*' in Data #5 and #6, the line is treated as a comment line.

Data #5 NUCIN, RTEMP (A, R4)

NUCIN Nuclide ID to be processed.
 RTEMP Target temperature (Kelvin).

Data #6 NUCOUT, OFILE (A, A)

NUCOUT Nuclide ID for generated cross sections. The character-string length must be not more than 12.

OFILE File name of the generated cross section data. The character-string length including NUCOUT must be not more than 144. The actual file path is PATH+OFILE.

C.2 Sample Input Data

The following B-shell script is an example to generate temperature-dependent cross sections for MVP input data sample1(p. 326). The input data is written between the line “cat - << END_DATA | \$EXEC > \$OUTLIST 2>&1 ” and the last line “END_DATA”.

```
#!/bin/sh
#
# ... Sample shell script and input data for MVPART ...
#
# (Cross section data is prepared for MVP sample problem 1.)
#
EXEC=$MVP_DIR/bin/$MVPHOST/mvpart
CASE=MVPART.sample01
DATE=`date +%Y%m%d.%H%M%S`
#
# ... neutron index file for arbitrary-temp libraries ...
#
ININDEX=/cygdrive/c/MVP-GMVP/MVPlib/neutron.art.index
#
# ... directory where cross section files are created ...
#
NEWLIBDIR=$PWD/USRlib
#
# ... index file for generated cross section data ...
#
OUTINDEX=sample01.index
#
OUTLIST=$CASE.$DATE.ft06
#
if [ ! -d $NEWLIBDIR ]; then
    mkdir $NEWLIBDIR
fi
#
cat - << END_DATA | $EXEC > $OUTLIST 2>&1
0 0 / IVPOP IDBG
$ININDEX
path $NEWLIBDIR/
index $OUTINDEX
*
* ... Fuel : 293.15 => 1500.0 Kelvin
*
U02350J40 1500.0
U023501500 U02350-T1500.J40.dat
```

```

U02380J40      1500.0
U023801500    U02380-T1500.J40.dat
PU2390J40      1500.0
PU23901500    PU2390-T1500.J40.dat
PU2400J40      1500.0
PU24001500    PU2400-T1500.J40.dat
PU2410J40      1500.0
PU24101500    PU2410-T1500.J40.dat
O00160J40      1500.0
O001601500    O00160-T1500.J40.dat
*
* ... Cladding : 293.15 => 900.0 Kelvin
*
CR0500J40      900.0
CR05000900    CR0500-T0900.J40.dat
CR0520J40      900.0
CR05200900    CR0520-T0900.J40.dat
CR0530J40      900.0
CR05300900    CR0530-T0900.J40.dat
CR0540J40      900.0
CR05400900    CR0540-T0900.J40.dat
MN0550J40      900.0
MN05500900    MN0550-T0900.J40.dat
FE0540J40      900.0
FE05400900    FE0540-T0900.J40.dat
FE0560J40      900.0
FE05600900    FE0560-T0900.J40.dat
FE0570J40      900.0
FE05700900    FE0570-T0900.J40.dat
FE0580J40      900.0
FE05800900    FE0580-T0900.J40.dat
NI0580J40      900.0
NI05800900    NI0580-T0900.J40.dat
NI0600J40      900.0
NI06000900    NI0600-T0900.J40.dat
NI0610J40      900.0
NI06100900    NI0610-T0900.J40.dat
NI0620J40      900.0
NI06200900    NI0620-T0900.J40.dat
NI0640J40      900.0
NI06400900    NI0640-T0900.J40.dat
MO0920J40      900.0
MO09200900    MO0920-T0900.J40.dat
MO0940J40      900.0
MO09400900    MO0940-T0900.J40.dat
MO0950J40      900.0
MO09500900    MO0950-T0900.J40.dat
MO0960J40      900.0
MO09600900    MO0960-T0900.J40.dat
MO0970J40      900.0
MO09700900    MO0970-T0900.J40.dat
MO0980J40      900.0
MO09800900    MO0980-T0900.J40.dat
MO1000J40      900.0

```

```
MO10000900  MO1000-T0900.J40.dat
*
* ... Coolant : 293.15 => 800.0 Kelvin
*
NA0230J40      800.0
NA02300800  NA0230-T0800.J40.dat
END_DATA
```

D. Sample Programs to Read Binary Output File

Sample programs to read an MVP/GMVP binary output file are listed in this Appendix. The binary file is output on FORTRAN unit 30. All the records in the file are tagged except for the first title and date lines. The user can write a simple program to read the binary file by using the tags. Two sample programs are shown below.

D.1 Sample 1

The following program reads energy bin boundaries and flux calculated by the track length estimator and then output the normalized neutron flux (neutron energy spectrum). The program consists of the main program and the subroutine SRTAG. The subroutine searches a specified tag in a file on unit NMVPB.

- main program

```

c
c ... Simple program to read MVP binary output ...
c
    program SMPL01

    character TITLE0*64, DATE*17, TITLE1*72, TITLE2*72, TAG*32
    real*8      WSUM

    real        ENGYB( 100 )
    real*8      SFLTR( 100, 20, 2 )

c
c ... I/O units ...
c
    NSTDI  =  5
    NSTDO  =  6
    NMVPB  =  1
    NTXTO  =  6

c
c ... open file ...
c
    open( NMVPB, file='mvp_binary_results.fort.30',
          & status='old',access='sequential',form='unformatted' )

c
c ... read binary file ...
c
    read( NMVPB ) TITLE0
    write( NTXTO, '(a)') TITLE0

    read( NMVPB ) DATE
    write( NTXTO, '(a)') DATE

    read( NMVPB ) TITLE1, TITLE2

```

```

write( NTXTO, '(a/a)') TITLE1, TITLE2

read( NMVPB ) TAG, WSUM, NTHIST, NBATCH, NGROUP, NREG, NRESP,
& NSKIP, NSTAL, NGP1, NGP2, NTREG, NPDET, NTIME, NSTALY, JNEUT,
& JPHOT, JRESP, JEIGN,JADJM,JTIME
write( NTXTO, * ) TAG, WSUM, NTHIST, NBATCH, NGROUP, NREG, NRESP,
& NSKIP, NSTAL, NGP1, NGP2, NTREG, NPDET, NTIME, NSTALY, JNEUT,
& JPHOT, JRESP, JEIGN,JADJM,JTIME
C
C ----- ENERGY BOUNDARY -----
C
C      0-----1-----2-----3--
TAG    = 'ENERGY & REG.VOLUME          '
C      0-----1-----2-----3--
call SRTAG( NMVPB, TAG, IRC )

if( IRC.eq.0 ) then
  write( NSTDO, * ) ' XXX NO ENERGY BOUNDARY DATA !! STOP '
  stop
end if

read( NMVPB ) TAG, (ENGYB(I),I=1,NGP1+1)
write( NTXTO, * ) TAG, (ENGYB(I),I=1,NGP1+1)
C
C ----- FLUX : TRACK LENGTH -----
C
C      0-----1-----2-----3--
TAG    = 'FLUX & ERROR: TRACK LENGTH    '
C      0-----1-----2-----3--
call SRTAG( NMVPB, TAG, IRC )

if( IRC.eq.0 ) then
  write( NSTDO, * ) ' XXX NO FLUX DATA !! STOP '
  stop
end if

do KR = 1, NTREG
  read( NMVPB ) TAG, (SFLTR(IG,KR,1),IG=1,NGROUP),
& (SFLTR(IG,KR,2),IG=1,NGROUP)
  write( NTXTO, * ) TAG, (SFLTR(IG,KR,1),IG=1,NGROUP),
& (SFLTR(IG,KR,2),IG=1,NGROUP)
end do
C
C ... calculate & output normalized flux ...
C
NT = 1

FN = 0.d0
do IG = 1, NGP1
  FN = FN + SFLTR(IG,NT,1)*log(ENGYB(IG)/ENGYB(IG+1))
end do

write( NSTDO, '(////)' )
write( NSTDO, '(''# GRP'',8x,'ENERGY'',10x,'FLUX'',11x,
```

```

&          ' '1SIGMA' ' ' ' )

      do IG = NGP1, 1, -1
        write( NSTDO, '(1x,i5,3x,e13.5,3x,e13.5,3x,e13.5)' )
&          IG, ENGYB(IG+1), SFLTR(IG,NT,1)/FN,
&          SFLTR(IG,NT,1)*SFLTR(IG,NT,2)/FN/100.d0
      end do
      write( NSTDO, '(1x,i5,3x,e13.5,3x,e13.5,3x,e13.5)' )
&          0, ENGYB(1), 0., 0.
c
c ... close file ...
c
      close( unit=NMVPB, status='keep' )
c
      stop
      end

```

- subroutine SRTAG

```

c
c ===== SEARCH RECORD TAG =====
c
      subroutine SRTAG( NFILE, TAG, IRC )
c
      character*32 TAG
      character*32 STAG
c
      IRC = 0 : RECORD WITH TAG NOT FOUND
      IRC = 1 : RECORD WITH TAG IS FOUND
c
      IRC = 0
1000 read( NFILE, END=999 ) STAG
      if ( STAG.eq.TAG ) then
        IRC = 1
        backspace NFILE
        return
      end if
      go to 1000
c
999 return
      end

```

D.2 Sample 2

The following program reads results for the TALLY data block (results of special tallies) and then output the time-dependent neutron flux . The program consists of the main program and subroutines RDSTL, SRTAG and CCOMP. The subroutine SRTAG is the same as the one in Sample 1. The subroutine RDSTL reads special tally results and output the formatted results. The user should modify the last section of RDSTL to change the output format. The subroutine CCOMP is a utility routine for character strings.

- main program

```

c
c ... Simple program to read MVP binary output ...
c
  program SMPL02

  parameter( MEMORY = 2000000 )

  character TITLE0*64, DATE*17, TITLE1*72, TITLE2*72, TAG*32
  real*8      WSUM

  real        A( MEMORY )

c
c ... I/O units ...
c
  NSTDI  = 5
  NSTDO  = 6
  NMVPB  = 1
  NTXTO  = 6

c
c ... open file ...
c
  open( NMVPB, file='mvp_binary_results.fort.30',
    & status='old',access='sequential',form='unformatted' )

c
c ... read binary file ...
c
  read( NMVPB ) TITLE0
  write( NTXTO, '(a)') TITLE0

  read( NMVPB ) DATE
  write( NTXTO, '(a)') DATE

  read( NMVPB ) TITLE1, TITLE2
  write( NTXTO, '(a/a)') TITLE1, TITLE2

  read( NMVPB ) TAG, WSUM, NTHIST, NBATCH, NGROUP, NREG, NRESP,
    & NSKIP, NSTAL, NGP1, NGP2, NTREG, NPDET, NTIME, NSTALY, JNEUT,
    & JPHOT, JRESP, JEIGN,JADJM,JTIME
  write( NTXTO, * ) TAG, WSUM, NTHIST, NBATCH, NGROUP, NREG, NRESP,
    & NSKIP, NSTAL, NGP1, NGP2, NTREG, NPDET, NTIME, NSTALY, JNEUT,
    & JPHOT, JRESP, JEIGN,JADJM,JTIME

c
c ... read special tallies ...
c
  0-----1-----2-----3--
TAG  = 'TALLIES SPECIAL          '
c
  0-----1-----2-----3--
call SRTAG( NMVPB, TAG, IRC )

if( IRC.eq.0 ) then
  write( NSTDO, * ) ' XXX NO SPECIAL TALLY DATA !! STOP '
  stop
end if

```



```

read( NMVPB ) TAG, NETALY, NEDTMX, NBINMX, NETRV
write( NTXTO, * ) TAG, NETALY, NEDTMX, NBINMX, NETRV

LETALY = 1
if( MOD(LETALY,2).eq.0 ) LETALY = LETALY + 1
LETRV = LETALY + NEDTMX*2*2
LBIN = LETRV + NEDTMX*2
LRNM = LBIN + NBINMX*2*8*2
LTEMP = LRNM + NBINMX*128

if( LTEMP.gt.MEMORY ) then
  write(6,*) ' XXX MEMORY SIZE IS INSUFFICIENT TO TREAT ',
&           'SPECIAL TALLIES.'
  write(6,*) '      AT LEAST ',LTEMP,' WORDS ARE NECESSARY.'
  stop
end if

c ... special tally index ...
c   >0: ID # ; <0: SEQUENTIAL #
c   ISTNO = 11

  call RDSTL( NSTDO, NTXTO, NMVPB, ISTNO,
&           NETALY, A(LETALY), NEDTMX, A(LETRV),
&           A(LBIN), A(LRNM), NBINMX )
c
c ... close file ...
c
c   close( unit=NMVPB, status='keep' )
c
c   stop
c   end

● subroutine RDSTL

  subroutine RDSTL ( N06, NTXTO, NFILE, ISTNO,
&                 NETALY, ETALY, NEDTMX, ETRV,
&                 BIN, RNM, NBINMX )

C=====
C   NFILE : unit of input file
C   ISTNO : SPECIAL TALLY #
C           >0: ID # ; <0: SEQUENTIAL #
C=====

  real*8  ETALY(NEDTMX,2), ETRV(NEDTMX)
  real*8  BIN(NBINMX,2,8)
  character RNM(NBINMX)*128

C
C ... local data ...
C
  integer INFO(10)
  parameter( MAXDIM = 8 )

C
  character TAG*6
  character PLAB*32

```

```

character TLABEL*128
C
character*32 TAGS
C
character DIMLBL( MAXDIM )*16
integer NDIME( MAXDIM )
C-----
C          0----*----1----*----2----*----3--
TAGS  = 'TALLIES SPECIAL'
C          0----*----1----*----2----*----3--

if( ISTNO.lt.0 ) then

    N = ABS( ISTNO )

    if( N.gt.NETALY ) then
        write(N06,*) ' XXX SPECIFIED SPECIAL TALLY # (',N,
&          ' ) IS INCORRECT !! STOP.'
        write(N06,*) ' PLEASE SET CORRECT IRG PARAMETER.'
        stop
    endif

    TAG      = ' '
    write(TAG,'(''.'',i5)') N
    call CCOMP( TAG, LEN(TAG), TAG, IIF )
    LTAG = index(TAG,' ') - 1
    if ( LTAG.lt.0 ) LTAG = len(TAG)

    PLAB      = 'TALLY'//TAG(:LTAG)
    TAGS      = PLAB
    call SRTAG( NFILE, TAGS, IRC )
    if( IRC.eq.0 ) then
        write(N06,*) ' XXX NO SPECIAL TALLY DATA !! STOP '
&          ' , TAGS
        stop
    end if

    read(NFILE) TAGS,N,INFO(4),JRV
    write( NTXTO, * ) TAGS,N,INFO(4),JRV
    read(NFILE) TAGS,L,TLABEL
    write( NTXTO, * ) TAGS,L,TLABEL
    read(NFILE) TAGS,L,(INFO(J),J=1,10)
    write( NTXTO, * ) TAGS,L,(INFO(J),J=1,10)

else

    do N = 1, NETALY
        TAG      = ' '
        write(TAG,'(''.'',i5)') N
        call CCOMP( TAG, LEN(TAG), TAG, IIF )
        LTAG = index(TAG,' ') - 1
        if ( LTAG.lt.0 ) LTAG = len(TAG)

        PLAB      = 'TALLY'//TAG(:LTAG)

```

```

TAGS      = PLAB
CALL      SRTAG( NFILE, TAGS, IRC )

if( IRC.eq.0 ) then
    write(N06,*) ' XXX NO SPECIAL TALLY DATA !! STOP '
&          , TAGS
    stop
end if

read(NFILE) TAGS,NN,INFO(4),JRV
write( NTXTO, * ) TAGS,NN,INFO(4),JRV
read(NFILE) TAGS,L,TLABEL
write( NTXTO, * ) TAGS,L,TLABEL
read(NFILE) TAGS,L,(INFO(J),J=1,10)
write( NTXTO, * ) TAGS,L,(INFO(J),J=1,10)

if( INFO(1).eq.ISTNO ) go to 2000
end do

write(6,*) ' XXX SPECIFIED SPECIAL TALLY ID # (',ISTNO,
&          ') IS INCORRECT !! STOP.'
write(6,*) ' PLEASE SET CORRECT IRG PARAMETER.'
stop

endif

C
2000 continue
C
    write(N06,'(A,I3,A,I5,A)')
&    'Special tally No.', N, ' (ID=',INFO(1),')'
    write(N06,'(1x,''LABEL: <''',a, ''>''/)'') TLABEL
C
    NSIZ      = 1
    do J = 1, INFO(4)
        read(NFILE) TAGS,DIMLBL(J),NDIME(J)
        write(N06,'(A,2X,A,i7)') TAGS,DIMLBL(J)(:8),NDIME(J)
        if( DIMLBL(J)(1:3).eq.'REG' ) then
            read(NFILE) TAGS,L,(RNM(JJ),JJ=1,NDIME(J))
            write( NTXTO, * ) TAGS,L,(RNM(JJ),JJ=1,NDIME(J))
        else
            read(NFILE) TAGS,(BIN(JJ,1,J),BIN(JJ,2,J),JJ=1,NDIME(J))
            write( NTXTO, * )
&          TAGS,(BIN(JJ,1,J),BIN(JJ,2,J),JJ=1,NDIME(J))
        endif
        NSIZ      = NSIZ*NDIME(J)
    end do
C
    write(N06,7020) (DIMLBL(J)(:8),',',J=1,INFO(4)), ' '
7020    format(1X,'  DIMENSION( ',5(A,A,:' ')/(1X,13X,5(A,A,:' ')))
    write(N06,7040) (NDIME(J),J=1,INFO(4))
7040    format(1X,'  SIZE      : ',5(I8,:' ')/(1X,13X,5(I8,:' ')))
C
    read(NFILE) TAGS,L,(ETALY(I,1),I=1,L)
    if( NSIZ.ne.L ) then

```

```

        write(N06,*) ' XXX ',TAGS(1:10),',', NSIZ(',L,') is wrong.'
    end if
    write( NTXTO, * ) TAGS,L,(ETALY(I,1),I=1,L)

    read(NFILE) TAGS,L,(ETALY(I,2),I=1,L)
    if( NSIZ.ne.L ) then
        write(N06,*) ' XXX ',TAGS(1:11),',', NSIZ(',L,') is wrong.'
    end if
    write( NTXTO, * ) TAGS,L,(ETALY(I,2),I=1,L)

    if ( JRV.ne.0 ) then
        read(NFILE) TAGS,L,(ETRV(I),I=1,L)
        if( NSIZ.ne.L ) then
            write(N06,*) ' XXX ',TAGS(1:15),',', NSIZ(',L,') is wrong.'
        end if
        write( NTXTO, * ) TAGS,L,(ETRV(I),I=1,L)
    end if

c
c ... output formatted results ...
c
    write( N06, '(////)' )

    JTIM = 1
    JTRG = 2

    do J = 1, NDIME(JTRG)

        write( N06, '(/''# TALLY REGION:'',a12)') RNM(J)
        write( N06, '(/''# NUM'',8x,'TIME'',12x,'FLUX'',11x,
&                '1SIGMA'')')

        IC = 0
        do I = J, NDIME(JTIM)*NDIME(JTRG), NDIME(JTRG)
            IC = IC + 1
            write( N06, '(1x,i5,3x,e13.5,3x,e13.5,3x,e13.5)')
&            IC, BIN(IC,1,JTIM), ETALY(I,1),
&            ETALY(I,1)*ETALY(I,2)/100.d0
        end do
        write( N06, '(1x,i5,3x,e13.5,3x,e13.5,3x,e13.5)')
&        IC+1, BIN(IC,2,JTIM), 0., 0.

    end do

    return
end

```

● subroutine CCOMP

```

    subroutine CCOMP( OUTCH, LN,      INCH,  IFL )
c
c  GMVP/MVP UTILITY
c
c=====
c  PURPOSE:  'COMPRESS' CHARACTER STRING INCH BY REMOVING BLANK
c            INTO CHARACTER STRING 'OUTCH' UPTO LN CHARACTERS.

```

```

C          ( 'OUTCH' & 'INCH' CAN OVERLAP IN MEMORY BUT
C          THE STARTING ADDRESS OF 'OUTCH' MUST PRECEED OR BE SAME AS
C          THAT OF 'INCH'. )
C
C          IFL : FLAG
C              0 = NORMAL END
C              1 = NUMBER OF NON-BLANK CHARACTERS IN INCH EXCEEDS LN.
C
C  HISTORY:  PROGRAMMED BY M.SASAKI ( 18 MAR 1992 )
C=====
C          character*(*) OUTCH, INCH
C          character*1 CH1
C
C          IFL      = 0
C          LL       = LEN(INCH)
C          NN       = 0
C          do 100 I = 1, LL
C              if ( INCH(I:I).ne.' ' ) then
C                  NN      = NN + 1
C                  if ( NN.le.LN ) then
C                      CH1   = INCH(I:I)
C                      OUTCH(NN:NN) = CH1
C                  end if
C              end if
C          end if
C 100 continue
C          if ( NN.lt.LN ) OUTCH(NN+1:LN) = ' '
C          if ( NN.gt.LN ) IFL = 1
C          return
C          end

```

E. Codes to Convert the Form of Cross Section Libraries

The MVP code can read the binary form of cross section libraries only. Therefore the text form of the libraries must be converted into the binary form when the libraries are provided in the text form. Conversely, there are some cases where the binary form should be converted into the text form to transfer the library data on systems with the different file format or the different presentation of numerical data/characters. NTXT2LB is a code to convert the text form of neutron, photoatomic, photonuclear and electron libraries into the binary form. Conversely, NLB2TXT is a code to convert the binary form of MVP libraries into the text form.

The GMVP code can also read the binary form of multigroup cross section data only. When the text form of the multigroup cross section data is prepared, the user can convert the data into the binary form with the GMVPLBCV code.

E.1 Conversion of MVP Libraries from Text to Binary Form

The NTXT2LB code converts the text form of libraries produced with NLB2TXT into the binary form used in the MVP code. The index data can be also created together with the data conversion; the index data is the list of nuclide IDs with the file names of MVP libraries.

The following data should be input from the standard input. The data must be character data. If more than one character string must be input, the strings must be delimited with blanks. The strings can be put at any position between the first and 72nd columns.

(1) Name of a new index file

Name of the index file where converted file names and nuclide IDs are output. The user should pay attention to the index file name. If the name of an existing index file is specified, the content of the original file is erased.

(2) File name of input text-form data

If the character ' - ' is specified as the input file name, text-form cross section data follow the end of control data (4).

(3) File name of output binary data

(4) Nuclide ID's converted in the input file.

Data (4) can be repeated 64 times. If a blank data is input for data (4), the input of the control data is terminated.

If the input file name is '-', cross section data are input after data (4).

There are following rules for the output file name etc.

- If a character '*' is included in the file name of output binary data (control data (3)), '*' is replaced with the nuclide ID's. If no '*' is included, the nuclide ID's are added after the file name.

- Control data (4) specifies the matching pattern of the nuclide ID's. It can include wild card characters such as '*' and '?'. '*' and '?' indicate any character string not less than one character and any character. If a blank line is input for control data (4), it has the same meaning as *.

Example:

The user converts the text-form cross section data (file name : mvp.nlib.txt) but only cross sections whose nuclide ID includes "B4" for last 2 characters are converted into the binary data named "nuclide-ID.b4lib". The index information is output to the file tempindex.dat.

```

line      .....
1         tempindex.dat
2         mvp.nlib.txt
3         *.b4lib
4         *B4
5
          .....

```

E.2 Conversion of MVP Libraries from Binary to Text Form

The NLB2TXT code converts the binary form of MVP cross section data into the text form. There are following input and output files for NLB2TXT.

- Input file (neutron, photoatomic, photonuclear or electron library):
The file can be specified in following two ways.
 - All cross section data files or a cross section data file of specified nuclide ID's of MVP libraries.
 - A file specified from the standard input.
- Output file:
One or more binary input files are output to one or more text files.

The user must input the following control data from the standard input.

Control input data for NLB2TXT

- (1) Type of particles to be processed.

One of the following character strings must be input from the first column.

- NEUTRON : neutron library
- PHOTON : photoatomic library
- PHOTONUCLEAR : photonuclear library
- ELECTRON : electron library

- (2) Index file name

Index file name which includes the file names of cross section data to be converted or a blank line.

- (3) Specification of input binary data to be converted and the file name of output text data.

The input form of data (3) depends on data (2).

(A) Case of an index file name (not a blank line):

The following record can be input repeatedly any times.

[NUC=nuclide ID matching pattern] [file name of output text data]

- If “NUC=nuclide ID matching pattern” is omitted, all files listed in the index file are converted into one or more text files, depending on the specification of output text file name.
- If “NUC=nuclide ID matching pattern” is specified, only the cross sections matching the pattern are converted. The pattern can include wild card characters ‘*’ (zero or more characters) and ‘?’ any character.
- If the file name of output text data includes ‘*’, ‘*’ is replaced with the nuclide ID for each binary file.
- Unless the file name of output text data includes ‘*’, all the converted data are output to the file.
- If the file name of output text data is omitted, the converted data are added to the last specified output file. Accordingly, the first control data (3) must include the file name of output text data.

(B) Case of a blank line:

The pair of “the file name of input binary data” and “the file name of output text data” can be input any times. If the file name of output text data is ‘+’, the file name is specified in the next line. This is used for a long file name etc. If the output file name is omitted, the converted data are added to the last specified output file. The first output file name cannot be omitted.

Example of input data for NLB2TXT:

The user converts neutron cross sections of all nuclides included in an index file nindex.dat into the text files whose name is “nuclide-ID.txt”.

```
line .....
1      NEUTRON
2      nindex.dat
3      *.txt
      .....
```

The user converts only the cross section data whose nuclide ID matches input patterns in an index file nindex.dat.

```
line .....
1      NEUTRON
2      nindex.dat
3      NUC=*B70   B70lib.txt
4      NUC=PU*J40 J40FIS.txt
5      NUC=U0*J40
      .....
```

The user converts only the cross section files for specified nuclides are converted into 2 text files.

```
line .....
1      NEUTRON
2
3      U02350B70.nlibf2  U0235.textlib
4      U02350J40.nlibf1
5      U02380J40.nlibf1  U0238.textlib
6      U02380B70.nlibf2
      .....
```


E.3 Conversion of File Form for GMVP Libraries

The GMVP code, in principle, reads the binary form of cross section data as well as MVP. GMVPLBCV is a code to convert the data form for GMVP cross sections. By using the code, the user can transfer the cross sections to the different type of systems. The input data must be input from the standard input. GMVPLBCV can convert binary into text data and text into binary data.

Control input data for GMVPLBCV

The first input is the specification of the conversion mode.

- 1 : Binary \rightarrow Text
- 2 : Text \rightarrow Binary
- The following data must be input in case 1 (Binary \rightarrow Text)
 - (a) Library form
 - 1 : KENO-IV
 - 2 : GMVP/ANISN form (Cross section data of all groups are stored in one record.)
 - 3 : ANISN form 2 (Cross section data of each group are stored in one record.)
 - 4 : DDX form
 - (b) Input file name
 - (c) Output file name
 - (d) Following 4 data are required if form 2 or 3 is selected for (a).
 NGP, IHM, IHT, IHS
 - NGP : Number of energy groups
 - IHM : Length of the cross section table
 - IHT : Position of total cross section
 - IHS : Position of self-scattering cross section
- The following data must be input in case 2 (Text \rightarrow Binary)
 - (a) Input file name
 - (b) Output file name

F. FORTRAN-source Assimilation Tool

The FORTRAN language has a grammar specified by ISO, JIS etc. But actually some compilers have quirks and there are some cases where these make it difficult to write portable programs. In such cases, we have to rewrite source programs for a platform to be ported. There is another case where a file system is shared by multiple machines that have different platforms through the network. In this case, we also need portable source programs.

Therefore, we developed a preprocessing tool (MVPFAT: MVP Fortran-source Assimilation Tool) to select compiler-dependent sources. With this tool, we do not have to prepare source programs separately depending on the platforms. The dependent parts for all compilers are written in the same source file and the tool automatically selects the source according to some parameters.

The FORTRAN source programs of MVP/GMVP are written so that this tool can process the programs in compiling.

F.1 Function of MVPFAT

Writing programs for MVPFAT

MVPFAT can process the following program sources.

```
<Program source>
C/#IF expression
<Program source>
C/#ELSEIF expression
<Program source>
C/#ELSE
<Program source>
C/#ENDIF
<Program source>
```

The string “C/#” must be located in columns 1 through 3. A “C/#IF” always needs a corresponding “C/#ENDIF”. The strings “C/#ELSEIF” and “C/#ELSE” can be specified if necessary. A block from “C/#IF” to “C/#ENDIF” can be specified in the other block; namely the nesting structure is available. Multiple blanks or “#”s can be inserted between “#” and “IF”, “ELSE”, “ELSEIF” etc.

“Expression” denotes conditions to select a certain block. MVPFAT judges the conditions and selects the block according to the rule of a ordinary IF block like FORTRAN.

Process of MVPFAT

MVPFAT activates or deactivates source blocks according to the following rules.

- (1) If the first column is “*” in the activated source blocks, “*” is replaced with a blank. But the lines specified with the DIRECTIVE command described below are not changed. If the second column is neither a blank nor a number in the line where the first column is “*”, “*” is deleted and the line is shifted left by a character. If the first column is “C”, “c”, the line remains unchanged.
- (2) If the first column is a blank in the deactivated source blocks, the blank is replaced with “*” and the line is commented out. If the first column is neither “c”, “C” nor a blank, the line is shifted right by a character and “*” is inserted in the first column. But the lines specified with the DIRECTIVE command described below are not changed. If the first column is “C” or “c”, the line remains unchanged.

Since MVPFAT obeys the above rules, the programmer must pay attention to the following things in the C/#IF blocks.

- Line labels of 5 digits or line labels that start from the first column should not be used.
- “c” or “C” not “*” must be used for comment lines.
- The 72nd column may be lost for the deactivated comment lines.

“Expression” is similar to the form of logical expressions in FORTRAN and is described as follows.

```

expression := logical-operator expression
            := expression logical-operator expression
            := (expression)
            := parameter-name
            := parameter-name(sub-parameter-names)

```

```

logical-operator := .NOT.
                := .AND.
                := .OR.
                := .XOR. (exclusive OR)

```

The priority of the operators is .NOT. > .AND. > .OR. = .XOR. .

“parameter-name” is the name of a parameter set/unset with the SET command described below.
 “sub-parameter-name” is the name of an additional parameter for the parameter.

Evaluation of Expression

- (a) Case of expression := parameter-name

If “parameter-name” is set with the SET command below, the expression is true. If not, it is false.

- (b) Case of expression := parameter-name(sub-parameter-names)

If “parameter-name” with one of the “sub-parameter-names” is set with the SET command, the expression is true. If not, it is false. Wild card characters “*” (zero or more characters) and “?” (any character) can be used for “sub-parameter-names”.

- (c) Case where “expression” is a logical expression

If the logical expression is true, “expression” is also true. If not, “expression” is false.

MVPFAT commands

In order to enable the control with “C/#”, MVPFAT commands to set/unset the parameter and sub-parameter names must be specified. The commands can be specified from standard input data or in program sources. When the first column of lines is “*” in the standard input data, the lines are ignored as comments.

The following MVPFAT commands can be specified in program sources.

- C/#SET parameter-name[(sub-parameter-names)]
A parameter and its sub-parameters are set in program sources.
- C/#UNSET parameter-name[(sub-parameter-names)]
A parameter and its sub-parameters are unset in program sources.

The other MVPFAT commands must be specified from the standard input data.

- INPUT
The name of an input source file.
- OUTPUT
The name of an output source file.
- SET,UNSET
A parameter is set/unset. If sub-parameters exist, they are also set/unset.

Example:

```
SET UNIX
SET SYSTEM(LINUXELF)
```

The first line sets a parameter UNIX. The second line sets a parameter SYSTEM with a sub-parameter LINUXELF.

The UNSET command unsets parameters or sub-parameters set with the SET command. If a parameter involves sub-parameters and only the parameter name is specified, all the sub-parameters are unset.

Example:

```
UNSET UNIX
UNSET SYSTEM(LINUXELF)
```

The first line unsets a parameter UNIX. If the parameter UNIX involves sub-parameters, all the sub-parameters are unset. The second line unsets only the sub-parameter LINUXELF of the parameter SYSTEM. Even if the parameter SYSTEM involves other sub-parameters, they are not unset.

- INPUT, OUTPUT
The INPUT and OUTPUT commands can be specified only in the standard input data. The names of an input source file to be processed and an output source file can be specified, respectively. These commands can be specified only once. The same file name cannot be specified for INPUT and OUTPUT. If specified, the input file is destroyed. Unless the INPUT command is specified, the source file is read from I/O unit 10 (The MVPFAT program is written in FORTRAN.) or the standard input following the SOURCE command described below. Unless the OUTPUT command is specified, the processed source is output on I/O unit 20.

- **DIRECTIVE**

The **DIRECTIVE** command specifies directives that begin with “*” (*VOCL line for Fujitsu-VPP, etc.). If lines start with a character string specified with this command, they remain unchanged in both the activated and deactivated blocks.

- **MONITOR**

The **MONITOR** command specifies FORTRAN I/O unit to output the process of MVPFAT and error messages. This I/O unit must not be 10, 20 or 6 (standard output). Unless this command is specified, the I/O unit is 0. I/O unit 0 is assigned as the standard error output for almost all compilers on UNIX. For a system where output to I/O unit 0 causes some errors, an I/O unit except for 10, 20 and 6 must be specified.

- **SOURCE**

The **SOURCE** command can be specified in the standard input data. The standard input data followed by this command is considered to be a program source. The specification of the **INPUT** command is ignored.

F.2 Input and Output of MVPFAT

(1) Standard input:

- MVPFAT command (SET, UNSET, etc.).
- Program source followed by the **SOURCE** command.

(2) Input source:

File specified by the **INPUT** command, standard input data following the **SOURCE** command or I/O unit 10.

(3) Output source:

File specified by the **OUTPUT** command or I/O unit 20.

(4) Standard output:

Only the number of lines transformed is output (0 or positive integer). If errors occur in MVPFAT, a negative integer is output. With this output, the user can replace an input file with an output file only when a positive integer is output.

(5) Monitoring output:

Output the process of the transformation. Messages are output on an I/O unit specified by the **MONITOR** command.

F.3 Example of the Process with MVPFAT

Let us consider the following subroutine as an input data for MVPFAT. This subroutine retrieves a date on various platforms. In general, the subroutine to retrieve a date is system-dependent. MVPFAT selects a system-dependent call automatically in the following subroutine.

```

line      .....+.....
1          SUBROUTINE HIZUKE(XDATE)
2          CHARACTER*(*) XDATE
3          character*3 month(12)
4          data month /'Jan','Feb','Mar','Apr','May','Jun','Jul','Aug',
5              &      'Sep','Oct','Nov','Dec' /
6
7          C/#IF SYSTEM(FACOM)
8          C
9          C ----- FACOM M or VP series
10         C
11         CHARACTER*8 TDATE
12         XDATE = ' '
13         CALL DATE(TDATE)
14         READ(TDATE,'(I2,1X,I2,1X,I2)') IY,IM,ID
15         WRITE(XDATE,'(I2,'''-''',a3,'''-''',I2)') ID,MONTH(IM),IY
16     C/#ELSEIF SYSTEM(HP*)
17     C
18     C ----- HP-9000 SERIES ---
19     C
20     *$NOSTANDARD SYSTEM
21     *      XDATE = ' '
22     *      CALL IDATE(IM,ID,IY)
23     *      WRITE(XDATE,'(I2,'''-''',a3,'''-''',I2)') ID,MONTH(IM),IY
24     C
25     C/#ELSEIF SYSTEM(VMS)
26     C
27     C ----- IN VMS ---
28     C
29     *      XDATE = ' '
30     *      CALL DATE(XDATE(1:9))
31     C/#ELSE
32     C
33     C ----- IN UNKNOWN SYSTEM ---
34     C
35     *      XDATE = 'DATE?????'
36     C/#ENDIF
37     RETURN
38     END

```

When this input source file is processed with a MVPFAT command:

```
SET SYSTEM(HP700),
```

the output file is as follows.

```

7          C/#IF SYSTEM(FACOM)
8          C
9          C ----- FACOM M or VP series
10         C
11         *      CHARACTER*8 TDATE
12         *      XDATE = ' '
13         *      CALL DATE(TDATE)
14         *      READ(TDATE,'(I2,1X,I2,1X,I2)') IY,IM,ID
15         *      WRITE(XDATE,'(I2,'''-''',a3,'''-''',I2)') ID,MONTH(IM),IY
16     C/#ELSEIF SYSTEM(HP*)
17     C
18     C ----- HP-9000 SERIES ---
19     C
20     $NOSTANDARD SYSTEM
21     XDATE = ' '
22     CALL IDATE(IM,ID,IY)
23     WRITE(XDATE,'(I2,'''-''',a3,'''-''',I2)') ID,MONTH(IM),IY
24     C

```

```

25      C/#ELSEIF SYSTEM(VMS)
26      C

```

Lines 11 to 15 are commented out and lines 20 to 23 are activated. The character in the second column is “\$”, neither a blank nor a number, in line 20 of the input file. In this case, the line is shifted so that “\$” is in the first column.

F.4 Execution Command for MVPFAT

It is convenient for the user to use a shell script \$MVP_DIR/etc/Fatman for the execution of MVPFAT on UNIX. Fatman is a command for the easy use of the MVPFAT program. It has a function to transform vectorization directives of the FACOM format into those of the SX, CRAY, HITACHI formats.

<Usage>

Fatman *MVPFAT-file input-file output-file [post]*

MVPFAT-file : File including MVPFAT commands (SET, UNSET, etc.).
input-file : Input source file to be transformed.
output-file : Output source file after transformation.
post : Post-processing option.

The following options are available for “*post*”.

SX : Transform vectorization directives of the FACOM format into those of the SX format.
 CRAY : Transform vectorization directives of the FACOM format into those of the CRAY format.
 HITACHI : Transform vectorization directives of the FACOM format into those of the HITACHI format.

F.5 Parameters of SET and UNSET Commands

Parameter	Sub-parameter	Meaning
SYSTEM	LINUXELF	<Used system> Linux x86(ELF binary)
	FREEBSD	FreeBSD x86
	X86SOL2	Solaris2.x x86
	CYGWIN	Cygwin on Windows
	WIN32	Windows (32 bit)
UNIX		Use UNIX environment or compatible functions.
DYNAMIC		Dynamic memory allocation.
NOPOINTER		Use dynamic memory allocation but no POINTER statements are used.
ARGSAVE		Assume dummy arguments are saved in subroutines.

(Cont'd)

Parameter	Sub-parameter	Meaning
WORD	32	<Number of bits in 1 word> 32 bits.
	64	64 bits.
ADRSIZE	32	<Number of bits for memory addressing> 32 bits.
	64	64 bits.
ROUND OFF	TRUNCATE	<Round off treatment for floating-point arithmetic> Truncated value.
	NEAREST	Nearest value.
READONLY	DEC	<Open for a read-only file> Use the READONLY keyword (DEC form).
	ACTION	Use the ACTION keyword (IBM, F90 etc.).
	MODE	Use the MODE keyword (MS-FORTRAN etc.).
FILENAME	DOS	<Expression for file paths> MS-DOS form.
	MACOS	MacOS form.
ARGV		<How to get command line arguments in FORTRAN>
	GETARG	GETARG routine (UNIX standard).
FLOATHANDLER		<Treatment of floating-point exception>
	ONSTMT	ON statement (HP-FORTRAN).
RANDOM		Use a random number generator that does not generate either 0 or 1.
CHARARG		<Dummy arguments for a character type>
	LOOSE	Possible to pass non-character type real arguments to character type dummy arguments.
NOGETENV		Not use environmental variables (or impossible).
VECTORIZE		Use vectorization routines.
SSL2		Use the RANU2 routine in the SSL2 library.
PARA		<Multitasking mode>
	VPP	Multitasking with Fujitsu VPP-Fortran.
	SX	Multitasking with NEC SX series.
	CRAY	Multitasking with CRAY MP series.
	PVM	PVM (Parallel Virtual Machine).
	MPI	MPI (Message Passing Interface).

G. MVP Cross Section Libraries

G.1 Neutron Libraries

G.1.1 JENDL-4.0

The contents of the MVP library based on JENDL-4.0 is listed in Tables G.1 and G.2. Table G.1 shows cross section data based on the free gas model. Tables G.2 shows cross section data including thermal scattering data based on ENDF/B-VI.8.

The meaning of column labels for the following tables is listed as follows.

Z	Atomic number.
Nuclide	Nuclide represented with the element name and the atomic mass number. -Nat means the natural isotope mixture. -M means the metastable state of the nuclide.
ID in MVP	Nuclide ID used for the specification in MVP input data.
MAT	MAT number defined in the ENDF format.
MAT(tape) for $S(\alpha, \beta)$	MAT number and ENDF tape number for $S(\alpha, \beta)$.
C.S.P.	“Yes” denotes that cross section probability tables in the unresolved resonance region are included. “No” denotes that cross section probability tables in the unresolved resonance region are not included.
γ-prod.	“Yes” denotes that γ production data are included. “No” denotes that γ production data are not included.
KERMA factor	“Yes” denotes that the KERMA factor is included. “No” denotes that the KERMA factor is not included.
ELOW	Lowest energy of the cross section data (eV).
EHI	Highest energy of the cross section data (eV).
Temp.	Temperature at which the cross section data were processed (K).
Max-T	Maximum temperature up to which the cross section data can be generated (K). If no value is specified, there is no limitation for temperature.

Table G.1 MVP library based on JENDL-4.0 with the free gas model (1/8)

Z	Nuclide	ID in MVP	MAT	C.S.P. Tables	γ -prod. data	KERMA factor	ELOW (eV)	EHI (eV)	Temp. (K)	Max-T (K)
1	H - 1	H00010J40	125	No	No	Yes	1.00e-05	2.00e+07	293.15	
1	H - 2	H00020J40	128	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
2	HE- 3	HE0030J40	225	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
2	HE- 4	HE0040J40	228	No	No	Yes	1.00e-05	2.00e+07	293.15	
3	LI- 6	LI0060J40	325	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
3	LI- 7	LI0070J40	328	No	Yes	Yes	1.00e-05	2.00e+07	293.15	

(Cont'd)

Table G.1 MVP library based on JENDL-4.0 with the free gas model (2/8)

Z	Nuclide	ID in MVP	MAT	C.S.P. Tables	γ -prod. data	KERMA factor	ELOW (eV)	EH1 (eV)	Temp. (K)	Max-T (K)
4	BE- 9	BE0090J40	425	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
5	B - 10	B00100J40	525	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
5	B - 11	B00110J40	528	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
6	C -Nat.	C00000J40	600	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
7	N - 14	N00140J40	725	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
7	N - 15	N00150J40	728	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
8	O - 16	O00160J40	825	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
9	F - 19	F00190J40	925	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
11	NA- 23	NA0230J40	1125	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
12	MG- 24	MG0240J40	1225	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
12	MG- 25	MG0250J40	1228	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
12	MG- 26	MG0260J40	1231	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
13	AL- 27	AL0270J40	1325	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
14	SI- 28	SI0280J40	1425	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
14	SI- 29	SI0290J40	1428	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
14	SI- 30	SI0300J40	1431	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
15	P - 31	P00310J40	1525	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
16	S - 32	S00320J40	1625	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
16	S - 33	S00330J40	1628	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
16	S - 34	S00340J40	1631	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
16	S - 36	S00360J40	1637	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
17	CL- 35	CL0350J40	1725	No	No	Yes	1.00e-05	2.00e+07	293.15	
17	CL- 37	CL0370J40	1731	No	No	Yes	1.00e-05	2.00e+07	293.15	
18	AR- 40	AR0400J40	1837	No	No	Yes	1.00e-05	2.00e+07	293.15	
19	K - 39	K00390J40	1925	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
19	K - 40	K00400J40	1928	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
19	K - 41	K00410J40	1931	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
20	CA- 40	CA0400J40	2025	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
20	CA- 42	CA0420J40	2031	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
20	CA- 43	CA0430J40	2034	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
20	CA- 44	CA0440J40	2037	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
20	CA- 46	CA0460J40	2043	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
20	CA- 48	CA0480J40	2049	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
21	SC- 45	SC0450J40	2125	No	No	Yes	1.00e-05	2.00e+07	293.15	
22	TI- 46	TI0460J40	2225	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
22	TI- 47	TI0470J40	2228	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
22	TI- 48	TI0480J40	2231	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
22	TI- 49	TI0490J40	2234	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
22	TI- 50	TI0500J40	2237	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
23	V - 50	V00500J40	2325	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
23	V - 51	V00510J40	2328	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
24	CR- 50	CR0500J40	2425	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
24	CR- 52	CR0520J40	2431	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
24	CR- 53	CR0530J40	2434	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
24	CR- 54	CR0540J40	2437	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
25	MN- 55	MN0550J40	2525	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
26	FE- 54	FE0540J40	2625	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
26	FE- 56	FE0560J40	2631	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
26	FE- 57	FE0570J40	2634	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
26	FE- 58	FE0580J40	2637	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
26	FE- 59	FE0590J40	2640	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
27	CO- 59	CO0590J40	2725	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
28	NI- 58	NI0580J40	2825	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
28	NI- 59	NI0590J40	2828	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
28	NI- 60	NI0600J40	2831	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
28	NI- 61	NI0610J40	2834	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
28	NI- 62	NI0620J40	2837	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
28	NI- 64	NI0640J40	2843	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
29	CU- 63	CU0630J40	2925	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
29	CU- 65	CU0650J40	2931	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
30	ZN- 64	ZN0640J40	3025	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
30	ZN- 65	ZN0650J40	3028	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
30	ZN- 66	ZN0660J40	3031	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
30	ZN- 67	ZN0670J40	3034	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0

(Cont'd)

Table G.1 MVP library based on JENDL-4.0 with the free gas model (3/8)

Z	Nuclide	ID in MVP	MAT	C.S.P. Tables	γ -prod. data	KERMA factor	ELOW (eV)	EH1 (eV)	Temp. (K)	Max-T (K)
30	ZN- 68	ZN0680J40	3037	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
30	ZN- 70	ZN0700J40	3043	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
31	GA- 69	GA0690J40	3125	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
31	GA- 71	GA0710J40	3131	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
32	GE- 70	GE0700J40	3225	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
32	GE- 72	GE0720J40	3231	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
32	GE- 73	GE0730J40	3234	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
32	GE- 74	GE0740J40	3237	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
32	GE- 76	GE0760J40	3243	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
33	AS- 75	AS0750J40	3325	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
34	SE- 74	SE0740J40	3425	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
34	SE- 76	SE0760J40	3431	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
34	SE- 77	SE0770J40	3434	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
34	SE- 78	SE0780J40	3437	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
34	SE- 79	SE0790J40	3440	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
34	SE- 80	SE0800J40	3443	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
34	SE- 82	SE0820J40	3449	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
35	BR- 79	BR0790J40	3525	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
35	BR- 81	BR0810J40	3531	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
36	KR- 78	KR0780J40	3625	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
36	KR- 80	KR0800J40	3631	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
36	KR- 82	KR0820J40	3637	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
36	KR- 83	KR0830J40	3640	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
36	KR- 84	KR0840J40	3643	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
36	KR- 85	KR0850J40	3646	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
36	KR- 86	KR0860J40	3649	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
37	RB- 85	RB0850J40	3725	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
37	RB- 86	RB0860J40	3728	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
37	RB- 87	RB0870J40	3731	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
38	SR- 84	SR0840J40	3825	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
38	SR- 86	SR0860J40	3831	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
38	SR- 87	SR0870J40	3834	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
38	SR- 88	SR0880J40	3837	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
38	SR- 89	SR0890J40	3840	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
38	SR- 90	SR0900J40	3843	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
39	Y - 89	Y00890J40	3925	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
39	Y - 90	Y00900J40	3928	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
39	Y - 91	Y00910J40	3931	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
40	ZR- 90	ZR0900J40	4025	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
40	ZR- 91	ZR0910J40	4028	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
40	ZR- 92	ZR0920J40	4031	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
40	ZR- 93	ZR0930J40	4034	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
40	ZR- 94	ZR0940J40	4037	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
40	ZR- 95	ZR0950J40	4040	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
40	ZR- 96	ZR0960J40	4043	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
41	NB- 93	NB0930J40	4125	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
41	NB- 94	NB0940J40	4128	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
41	NB- 95	NB0950J40	4131	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
42	MO- 92	MO0920J40	4225	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
42	MO- 94	MO0940J40	4231	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
42	MO- 95	MO0950J40	4234	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
42	MO- 96	MO0960J40	4237	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
42	MO- 97	MO0970J40	4240	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
42	MO- 98	MO0980J40	4243	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
42	MO- 99	MO0990J40	4246	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
42	MO-100	MO1000J40	4249	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
43	TC- 99	TC0990J40	4331	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
44	RU- 96	RU0960J40	4425	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
44	RU- 98	RU0980J40	4431	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
44	RU- 99	RU0990J40	4434	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
44	RU-100	RU1000J40	4437	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
44	RU-101	RU1010J40	4440	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
44	RU-102	RU1020J40	4443	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
44	RU-103	RU1030J40	4446	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0

(Cont'd)

Table G.1 MVP library based on JENDL-4.0 with the free gas model (4/8)

Z	Nuclide	ID in MVP	MAT	C.S.P. Tables	γ -prod. data	KERMA factor	ELOW (eV)	EH1 (eV)	Temp. (K)	Max-T (K)
44	RU-104	RU1040J40	4449	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
44	RU-105	RU1050J40	4452	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
44	RU-106	RU1060J40	4455	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
45	RH-103	RH1030J40	4525	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
45	RH-105	RH1050J40	4531	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
46	PD-102	PD1020J40	4625	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
46	PD-104	PD1040J40	4631	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
46	PD-105	PD1050J40	4634	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
46	PD-106	PD1060J40	4637	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
46	PD-107	PD1070J40	4640	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
46	PD-108	PD1080J40	4643	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
46	PD-110	PD1100J40	4649	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
47	AG-107	AG1070J40	4725	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
47	AG-109	AG1090J40	4731	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
47	AG-110-M	AG1101J40	4735	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
47	AG-111	AG1110J40	4737	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
48	CD-106	CD1060J40	4825	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
48	CD-108	CD1080J40	4831	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
48	CD-110	CD1100J40	4837	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
48	CD-111	CD1110J40	4840	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
48	CD-112	CD1120J40	4843	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
48	CD-113	CD1130J40	4846	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
48	CD-114	CD1140J40	4849	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
48	CD-116	CD1160J40	4855	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
49	IN-113	IN1130J40	4925	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
49	IN-115	IN1150J40	4931	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
50	SN-112	SN1120J40	5025	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
50	SN-114	SN1140J40	5031	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
50	SN-115	SN1150J40	5034	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
50	SN-116	SN1160J40	5037	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
50	SN-117	SN1170J40	5040	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
50	SN-118	SN1180J40	5043	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
50	SN-119	SN1190J40	5046	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
50	SN-120	SN1200J40	5049	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
50	SN-122	SN1220J40	5055	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
50	SN-123	SN1230J40	5058	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
50	SN-124	SN1240J40	5061	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
50	SN-126	SN1260J40	5067	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
51	SB-121	SB1210J40	5125	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
51	SB-123	SB1230J40	5131	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
51	SB-124	SB1240J40	5134	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
51	SB-125	SB1250J40	5137	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
51	SB-126	SB1260J40	5140	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
52	TE-120	TE1200J40	5225	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
52	TE-122	TE1220J40	5231	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
52	TE-123	TE1230J40	5234	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
52	TE-124	TE1240J40	5237	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
52	TE-125	TE1250J40	5240	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
52	TE-126	TE1260J40	5243	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
52	TE-127-M	TE1271J40	5247	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
52	TE-128	TE1280J40	5249	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
52	TE-129-M	TE1291J40	5253	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
52	TE-130	TE1300J40	5255	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
52	TE-132	TE1320J40	5261	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
53	I -127	I01270J40	5325	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
53	I -129	I01290J40	5331	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
53	I -130	I01300J40	5334	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
53	I -131	I01310J40	5337	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
53	I -135	I01350J40	5349	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
54	XE-124	XE1240J40	5425	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
54	XE-126	XE1260J40	5431	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
54	XE-128	XE1280J40	5437	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
54	XE-129	XE1290J40	5440	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
54	XE-130	XE1300J40	5443	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0

(Cont'd)

Table G.1 MVP library based on JENDL-4.0 with the free gas model (5/8)

Z	Nuclide	ID in MVP	MAT	C.S.P. Tables	γ -prod. data	KERMA factor	ELOW (eV)	EH1 (eV)	Temp. (K)	Max-T (K)
54	XE-131	XE1310J40	5446	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
54	XE-132	XE1320J40	5449	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
54	XE-133	XE1330J40	5452	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
54	XE-134	XE1340J40	5455	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
54	XE-135	XE1350J40	5458	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
54	XE-136	XE1360J40	5461	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
55	CS-133	CS1330J40	5525	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
55	CS-134	CS1340J40	5528	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
55	CS-135	CS1350J40	5531	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
55	CS-136	CS1360J40	5534	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
55	CS-137	CS1370J40	5537	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
56	BA-130	BA1300J40	5625	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
56	BA-132	BA1320J40	5631	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
56	BA-134	BA1340J40	5637	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
56	BA-135	BA1350J40	5640	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
56	BA-136	BA1360J40	5643	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
56	BA-137	BA1370J40	5646	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
56	BA-138	BA1380J40	5649	No	No	Yes	1.00e-05	2.00e+07	293.15	
56	BA-140	BA1400J40	5655	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
57	LA-138	LA1380J40	5725	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
57	LA-139	LA1390J40	5728	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
57	LA-140	LA1400J40	5731	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
58	CE-140	CE1400J40	5837	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
58	CE-141	CE1410J40	5840	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
58	CE-142	CE1420J40	5843	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
58	CE-143	CE1430J40	5846	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
58	CE-144	CE1440J40	5849	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
59	PR-141	PR1410J40	5925	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
59	PR-143	PR1430J40	5931	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
60	ND-142	ND1420J40	6025	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
60	ND-143	ND1430J40	6028	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
60	ND-144	ND1440J40	6031	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
60	ND-145	ND1450J40	6034	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
60	ND-146	ND1460J40	6037	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
60	ND-147	ND1470J40	6040	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
60	ND-148	ND1480J40	6043	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
60	ND-150	ND1500J40	6049	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
61	PM-147	PM1470J40	6149	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
61	PM-148	PM1480J40	6152	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
61	PM-148-M	PM1481J40	6153	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
61	PM-149	PM1490J40	6155	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
61	PM-151	PM1510J40	6161	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
62	SM-144	SM1440J40	6225	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
62	SM-147	SM1470J40	6234	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
62	SM-148	SM1480J40	6237	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
62	SM-149	SM1490J40	6240	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
62	SM-150	SM1500J40	6243	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
62	SM-151	SM1510J40	6246	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
62	SM-152	SM1520J40	6249	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
62	SM-153	SM1530J40	6252	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
62	SM-154	SM1540J40	6255	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
63	EU-151	EU1510J40	6325	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
63	EU-152	EU1520J40	6328	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
63	EU-153	EU1530J40	6331	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
63	EU-154	EU1540J40	6334	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
63	EU-155	EU1550J40	6337	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
63	EU-156	EU1560J40	6340	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
63	EU-157	EU1570J40	6343	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
64	GD-152	GD1520J40	6425	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
64	GD-153	GD1530J40	6428	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
64	GD-154	GD1540J40	6431	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
64	GD-155	GD1550J40	6434	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
64	GD-156	GD1560J40	6437	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
64	GD-157	GD1570J40	6440	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0

(Cont'd)

Table G.1 MVP library based on JENDL-4.0 with the free gas model (6/8)

Z	Nuclide	ID in MVP	MAT	C.S.P. Tables	γ -prod. data	KERMA factor	ELOW (eV)	EH1 (eV)	Temp. (K)	Max-T (K)
64	GD-158	GD1580J40	6443	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
64	GD-160	GD1600J40	6449	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
65	TB-159	TB1590J40	6525	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
65	TB-160	TB1600J40	6528	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
66	DY-154	DY1540J40	6619	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
66	DY-156	DY1560J40	6625	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
66	DY-158	DY1580J40	6631	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
66	DY-159	DY1590J40	6634	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
66	DY-160	DY1600J40	6637	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
66	DY-161	DY1610J40	6640	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
66	DY-162	DY1620J40	6643	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
66	DY-163	DY1630J40	6646	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
66	DY-164	DY1640J40	6649	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
68	ER-162	ER1620J40	6825	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
68	ER-164	ER1640J40	6831	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
68	ER-166	ER1660J40	6837	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
68	ER-167	ER1670J40	6840	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
68	ER-168	ER1680J40	6843	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
68	ER-170	ER1700J40	6849	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
69	TM-169	TM1690J40	6925	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
70	YB-168	YB1680J40	7025	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
70	YB-170	YB1700J40	7031	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
70	YB-171	YB1710J40	7034	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
70	YB-172	YB1720J40	7037	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
70	YB-173	YB1730J40	7040	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
70	YB-174	YB1740J40	7043	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
70	YB-176	YB1760J40	7049	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
72	HF-174	HF1740J40	7225	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
72	HF-176	HF1760J40	7231	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
72	HF-177	HF1770J40	7234	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
72	HF-178	HF1780J40	7237	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
72	HF-179	HF1790J40	7240	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
72	HF-180	HF1800J40	7243	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
72	HF-181	HF1810J40	7246	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
72	HF-182	HF1820J40	7249	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
73	TA-181	TA1810J40	7328	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
74	W -180	W01800J40	7425	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
74	W -182	W01820J40	7431	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
74	W -183	W01830J40	7434	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
74	W -184	W01840J40	7437	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
74	W -186	W01860J40	7443	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
76	OS-184	OS1840J40	7625	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
76	OS-186	OS1860J40	7631	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
76	OS-187	OS1870J40	7634	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
76	OS-188	OS1880J40	7637	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
76	OS-189	OS1890J40	7640	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
76	OS-190	OS1900J40	7643	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
76	OS-192	OS1920J40	7649	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
79	AU-197	AU1970J40	7925	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
80	HG-196	HG1960J40	8025	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
80	HG-198	HG1980J40	8031	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
80	HG-199	HG1990J40	8034	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
80	HG-200	HG2000J40	8037	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
80	HG-201	HG2010J40	8040	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
80	HG-202	HG2020J40	8043	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
80	HG-204	HG2040J40	8049	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
82	PB-204	PB2040J40	8225	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
82	PB-206	PB2060J40	8231	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
82	PB-207	PB2070J40	8234	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
82	PB-208	PB2080J40	8237	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
83	BI-209	BI2090J40	8325	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
88	RA-223	RA2230J40	8825	No	No	Yes	1.00e-05	2.00e+07	293.15	
88	RA-224	RA2240J40	8828	No	No	Yes	1.00e-05	2.00e+07	293.15	
88	RA-225	RA2250J40	8831	No	No	Yes	1.00e-05	2.00e+07	293.15	

(Cont'd)

Table G.1 MVP library based on JENDL-4.0 with the free gas model (7/8)

Z	Nuclide	ID in MVP	MAT	C.S.P. Tables	γ -prod. data	KERMA factor	ELOW (eV)	EH1 (eV)	Temp. (K)	Max-T (K)
88	RA-226	RA2260J40	8834	No	No	Yes	1.00e-05	2.00e+07	293.15	
89	AC-225	AC2250J40	8925	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
89	AC-226	AC2260J40	8928	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
89	AC-227	AC2270J40	8931	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
90	TH-227	TH2270J40	9025	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
90	TH-228	TH2280J40	9028	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
90	TH-229	TH2290J40	9031	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
90	TH-230	TH2300J40	9034	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
90	TH-231	TH2310J40	9037	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
90	TH-232	TH2320J40	9040	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
90	TH-233	TH2330J40	9043	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
90	TH-234	TH2340J40	9046	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
91	PA-229	PA2290J40	9125	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
91	PA-230	PA2300J40	9128	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
91	PA-231	PA2310J40	9131	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
91	PA-232	PA2320J40	9134	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
91	PA-233	PA2330J40	9137	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
92	U -230	U02300J40	9213	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
92	U -231	U02310J40	9216	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
92	U -232	U02320J40	9219	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
92	U -233	U02330J40	9222	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
92	U -234	U02340J40	9225	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
92	U -235	U02350J40	9228	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
92	U -236	U02360J40	9231	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
92	U -237	U02370J40	9234	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
92	U -238	U02380J40	9237	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
93	NP-234	NP2340J40	9337	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
93	NP-235	NP2350J40	9340	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
93	NP-236	NP2360J40	9343	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
93	NP-237	NP2370J40	9346	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
93	NP-238	NP2380J40	9349	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
93	NP-239	NP2390J40	9352	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
94	PU-236	PU2360J40	9428	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
94	PU-237	PU2370J40	9431	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
94	PU-238	PU2380J40	9434	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
94	PU-239	PU2390J40	9437	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
94	PU-240	PU2400J40	9440	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
94	PU-241	PU2410J40	9443	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
94	PU-242	PU2420J40	9446	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
94	PU-244	PU2440J40	9452	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
94	PU-246	PU2460J40	9458	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
95	AM-240	AM2400J40	9540	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
95	AM-241	AM2410J40	9543	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
95	AM-242	AM2420J40	9546	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
95	AM-242-M	AM2421J40	9547	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
95	AM-243	AM2430J40	9549	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
95	AM-244	AM2440J40	9552	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
95	AM-244-M	AM2441J40	9553	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
96	CM-240	CM2400J40	9625	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
96	CM-241	CM2410J40	9628	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
96	CM-242	CM2420J40	9631	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
96	CM-243	CM2430J40	9634	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
96	CM-244	CM2440J40	9637	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
96	CM-245	CM2450J40	9640	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
96	CM-246	CM2460J40	9643	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
96	CM-247	CM2470J40	9646	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
96	CM-248	CM2480J40	9649	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
96	CM-249	CM2490J40	9652	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
96	CM-250	CM2500J40	9655	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
97	BK-245	BK2450J40	9740	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
97	BK-246	BK2460J40	9743	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
97	BK-247	BK2470J40	9746	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
97	BK-248	BK2480J40	9749	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
97	BK-249	BK2490J40	9752	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0

(Cont'd)

Table G.1 MVP library based on JENDL-4.0 with the free gas model (8/8)

Z	Nuclide	ID in MVP	MAT	C.S.P. Tables	γ -prod. data	KERMA factor	ELOW (eV)	EH1 (eV)	Temp. (K)	Max-T (K)
97	BK-250	BK2500J40	9755	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
98	CF-246	CF2460J40	9843	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
98	CF-248	CF2480J40	9849	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
98	CF-249	CF2490J40	9852	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
98	CF-250	CF2500J40	9855	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
98	CF-251	CF2510J40	9858	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
98	CF-252	CF2520J40	9861	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
98	CF-253	CF2530J40	9864	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
98	CF-254	CF2540J40	9867	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
99	ES-251	ES2510J40	9911	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
99	ES-252	ES2520J40	9912	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
99	ES-253	ES2530J40	9913	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
99	ES-254	ES2540J40	9914	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
99	ES-254-M	ES2541J40	9915	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
99	ES-255	ES2550J40	9916	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
100	FM-255	FM2550J40	9936	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
psd	ZZ0500	ZZ0500J40	500	No	No	Yes	1.00e-05	2.00e+07	293.15	
psd	ZZ0900	ZZ0900J40	1040	No	No	Yes	1.00e-05	2.00e+07	293.15	
psd	ZZ0990	ZZ1110J40	1110	No	No	Yes	1.00e-05	2.00e+07	293.15	
dmy	ZZ0000	ZZ0000DMY	4534	No	No	No	1.00e-05	2.00e+07	0.00	

psd : Pseudo fission product for burn-up calculation. These cross section were made based on the MOSRA library with 200-group structure.

dmy : Dummy cross section not effective to neutron transport calculation. This is an alternative cross section data in burnup calculation for the nuclides whose nuclear data is not evaluated and decay half-time is sufficiently short.

Table G.2 MVP library based on JENDL-4.0 with ENDF/B-VI.8 thermal scattering data

Z	Nuclide	ID in MVP	MAT	C.S.P. Tables	γ -prod. data	KERMA factor	ELOW (eV)	EH1 (eV)	Temp. (K)	Max-T (K)	Pair nuclides
1	H - 1 in H ₂ O	H0001HJ40	125	No	No	Yes	1.00e-05	2.00e+07	296.00	1000.0	O00160J40 (free)
1	H - 1 in CH ₂	H0001PJ40	125	No	No	Yes	1.00e-05	2.00e+07	296.00	350.0	O00160J40 (free)
1	H - 1 in C ₆ H ₆	H0001QJ40	125	No	No	Yes	1.00e-05	2.00e+07	296.00	1000.0	C0000QJ40
1	H - 1 in ZrH	H0001ZJ40	125	No	No	Yes	1.00e-05	2.00e+07	296.00	1200.0	ZR09J1-6JZJ40
1	H - 1 in lCH ₄	H0001LJ40CH4	125	No	No	Yes	1.00e-05	2.00e+07	100.00		C0000LJ40CH4
1	H - 1 in sCH ₄	H0001SJ40CH4	125	No	No	Yes	1.00e-05	2.00e+07	22.00		C0000SJ40CH4
1	Ortho H - 1	H0001TJ40ORT	125	No	No	Yes	1.00e-05	2.00e+07	20.00		
1	Para H - 1	H0001TJ40PAR	125	No	No	Yes	1.00e-05	2.00e+07	20.00		
1	H - 2 in D ₂ O	H0002DJ40	128	No	Yes	Yes	1.00e-05	2.00e+07	296.00	1000.0	O00160J40 (free)
1	Ortho H - 2	H0002TJ40ORT	128	No	Yes	Yes	1.00e-05	2.00e+07	19.00		
1	Para H - 2	H0002TJ40PAR	128	No	Yes	Yes	1.00e-05	2.00e+07	19.00		
4	Be- 9 Metal	BE009BJ40	425	No	Yes	Yes	1.00e-05	2.00e+07	296.00	1200.0	
4	Be- 9 in BeO	BE009EJ40	425	No	Yes	Yes	1.00e-05	2.00e+07	296.00	1200.0	O0016EJ40
6	Graphite	C0000CJ40	600	No	Yes	Yes	1.00e-05	2.00e+07	296.00	2000.0	
6	C -Nat. in C ₆ H ₆	C0000QJ40	600	No	Yes	Yes	1.00e-05	2.00e+07	296.00	1000.0	H0001QJ40
6	C -Nat. in lCH ₄	C0000LJ40CH4	600	No	Yes	Yes	1.00e-05	2.00e+07	100.00		H0001LJ40CH4
6	C -Nat. in sCH ₄	C0000SJ40CH4	600	No	Yes	Yes	1.00e-05	2.00e+07	22.00		H0001SJ40CH4
8	O - 16 in BeO	O0016EJ40	825	No	Yes	Yes	1.00e-05	2.00e+07	296.00	1200.0	BE009EJ40
40	Zr- 90 in ZrH	ZR090ZJ40	4025	Yes	Yes	Yes	1.00e-05	2.00e+07	296.00	1200.0	H0001ZJ40
40	Zr- 91 in ZrH	ZR091ZJ40	4028	Yes	Yes	Yes	1.00e-05	2.00e+07	296.00	1200.0	H0001ZJ40
40	Zr- 92 in ZrH	ZR092ZJ40	4031	Yes	Yes	Yes	1.00e-05	2.00e+07	296.00	1200.0	H0001ZJ40
40	Zr- 93 in ZrH	ZR093ZJ40	4034	Yes	Yes	Yes	1.00e-05	2.00e+07	296.00	1200.0	H0001ZJ40
40	Zr- 94 in ZrH	ZR094ZJ40	4037	Yes	Yes	Yes	1.00e-05	2.00e+07	296.00	1200.0	H0001ZJ40
40	Zr- 95 in ZrH	ZR095ZJ40	4040	Yes	Yes	Yes	1.00e-05	2.00e+07	296.00	1200.0	H0001ZJ40
40	Zr- 96 in ZrH	ZR096ZJ40	4043	Yes	Yes	Yes	1.00e-05	2.00e+07	296.00	1200.0	H0001ZJ40

The data above thermal energy region are the same as those of the free gas model.

G.1.2 JENDL-4.0u1

The contents of the MVP library based on JENDL-4.0u1 is listed in Table G.3. Table G.3 shows cross section data based on the free gas model. The meaning of column labels for the following table is the same as in Table G.1.

Table G.3 MVP library based on JENDL-4.0u1 with the free gas model

Z	Nuclide	ID in MVP	MAT	C.S.P. Tables	γ -prod. data	KERMA factor	ELOW (eV)	EH1 (eV)	Temp. (K)	Max-T (K)
1	H - 2	H00020J40U1	128	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
5	B - 10	B00100J40U1	525	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
19	K - 39	K00390J40U1	1925	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
22	Ti - 48	TI0480J40U1	2231	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
24	CR- 52	CR0520J40U1	2431	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
24	CR- 53	CR0530J40U1	2434	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
28	NI - 59	NI0590J40U1	2828	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
47	AG-109	AG1090J40U1	4731	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
52	TE-127-M	TE1271J40U1	5247	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
52	TE-129-M	TE1291J40U1	5253	Yes	No	Yes	1.00e-05	2.00e+07	293.15	3000.0
63	EU-156	EU1560J40U1	6340	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
79	AU-197	AU1970J40U1	7925	No	Yes	Yes	1.00e-05	2.00e+07	293.15	
80	HG-196	HG1960J40U1	8025	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
80	HG-202	HG2020J40U1	8043	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
80	HG-204	HG2040J40U1	8049	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
88	RA-224	RA2240J40U1	8828	No	No	Yes	1.00e-05	2.00e+07	293.15	
88	RA-225	RA2250J40U1	8831	No	No	Yes	1.00e-05	2.00e+07	293.15	
92	U -233	U02330J40U1	9222	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
92	U -234	U02340J40U1	9225	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
92	U -235	U02350J40U1	9228	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
92	U -238	U02380J40U1	9237	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
93	NP-237	NP2370J40U1	9346	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
94	PU-238	PU2380J40U1	9434	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
94	PU-239	PU2390J40U1	9437	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
94	PU-242	PU2420J40U1	9446	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
95	AM-241	AM2410J40U1	9543	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0
95	AM-243	AM2430J40U1	9549	Yes	Yes	Yes	1.00e-05	2.00e+07	293.15	3000.0

G.1.3 JENDL-4.0TZ

The contents of the MVP library based on JENDL-4.0TZ is listed in Table G.4. (“TZ” : Temperature = Zero Kelvin.) Table G.4 shows cross section data based on the free gas model. Each nuclear data file in this library includes the elastic scattering cross section at 0 K, which is used for the exact resonance scattering model (DOPPLER-SCATTERING (EXACT) or DOPPLER-SCATTERING (EXACT+)). The details for the exact resonance scattering model are described in Section 4.2.6.

The meaning of column labels for the following table is the same as in Table G.1.

Table G.4 MVP library based on JENDL-4.0TZ with the free gas model

Z	Nuclide	ID in MVP	MAT	C.S.P. Tables	γ -prod. data	KERMA factor	ELOW (eV)	EH1 (eV)	Temp. (K)	Max-T (K)
92	U -235	U02350J40TZ	9228	Yes	Yes	No	1.00e-05	2.00e+07	0.00	3000.0
92	U -238	U02380J40TZ	9237	Yes	Yes	No	1.00e-05	2.00e+07	0.00	3000.0
94	PU-238	PU2380J40TZ	9434	Yes	Yes	No	1.00e-05	2.00e+07	0.00	3000.0
94	PU-239	PU2390J40TZ	9437	Yes	Yes	No	1.00e-05	2.00e+07	0.00	3000.0
94	PU-240	PU2400J40TZ	9440	Yes	Yes	No	1.00e-05	2.00e+07	0.00	3000.0
94	PU-241	PU2410J40TZ	9443	Yes	Yes	No	1.00e-05	2.00e+07	0.00	3000.0
94	PU-242	PU2420J40TZ	9446	Yes	Yes	No	1.00e-05	2.00e+07	0.00	3000.0
95	AM-241	AM2410J40TZ	9543	Yes	Yes	No	1.00e-05	2.00e+07	0.00	3000.0

G.2 Photoatomic Data Library

G.2.1 MCPLIB

The contents of the MVP photoatomic data library converted from the MCNP photoatomic data file MCPLIB^{33,34)} are listed in Table G.5.

The meaning of column labels for the following table is listed as follows.

Z	Atomic number.
Element	Symbol of the element.
ID in MVP	Element ID used for the specification in MVP input data.
F.X. data	“Yes” denotes that fluorescence X-ray data are included. “No” denotes that fluorescence X-ray data are not included.
ELOW	Lowest energy of the cross section data (eV).
EHI	Highest energy of the cross section data (eV).

Table G.5 MVP photoatomic data library (1/2)

Z	Element	ID in MVP	F.X. data	ELOW (eV)	EHI (eV)
1	H	H0NPTNMC	No	1.00E+03	1.00E+08
2	HE	HENPTNMC	No	1.00E+03	1.00E+08
3	LI	LINPTNMC	No	1.00E+03	1.00E+08
4	BE	BENPTNMC	No	1.00E+03	1.00E+08
5	B	B0NPTNMC	No	1.00E+03	1.00E+08
6	C	C0NPTNMC	No	1.00E+03	1.00E+08
7	N	N0NPTNMC	No	1.00E+03	1.00E+08
8	O	O0NPTNMC	No	1.00E+03	1.00E+08
9	F	F0NPTNMC	No	1.00E+03	1.00E+08
10	NE	NENPTNMC	No	1.00E+03	1.00E+08
11	NA	NANPTNMC	No	1.00E+03	1.00E+08
12	MG	MGNPTNMC	Yes	1.00E+03	1.00E+08
13	AL	ALNPTNMC	Yes	1.00E+03	1.00E+08
14	SI	SINPTNMC	Yes	1.00E+03	1.00E+08
15	P	P0NPTNMC	Yes	1.00E+03	1.00E+08
16	S	S0NPTNMC	Yes	1.00E+03	1.00E+08
17	CL	CLNPTNMC	Yes	1.00E+03	1.00E+08
18	AR	ARNPTNMC	Yes	1.00E+03	1.00E+08
19	K	K0NPTNMC	Yes	1.00E+03	1.00E+08
20	CA	CANPTNMC	Yes	1.00E+03	1.00E+08
21	SC	SCNPTNMC	Yes	1.00E+03	1.00E+08
22	TI	TINPTNMC	Yes	1.00E+03	1.00E+08
23	V	V0NPTNMC	Yes	1.00E+03	1.00E+08
24	CR	CRNPTNMC	Yes	1.00E+03	1.00E+08
25	MN	MNNPTNMC	Yes	1.00E+03	1.00E+08
26	FE	FENPTNMC	Yes	1.00E+03	1.00E+08
27	CO	CONPTNMC	Yes	1.00E+03	1.00E+08
28	NI	NINPTNMC	Yes	1.00E+03	1.00E+08
29	CU	CUNPTNMC	Yes	1.00E+03	1.00E+08
30	ZN	ZNNPTNMC	Yes	1.00E+03	1.00E+08
31	GA	GANPTNMC	Yes	1.00E+03	1.00E+08
32	GE	GENPTNMC	Yes	1.00E+03	1.00E+08
33	AS	ASNPTNMC	Yes	1.00E+03	1.00E+08
34	SE	SENPTNMC	Yes	1.00E+03	1.00E+08
35	BR	BRNPTNMC	Yes	1.00E+03	1.00E+08
36	KR	KRNPTNMC	Yes	1.00E+03	1.00E+08
37	RB	RBNPTNMC	Yes	1.00E+03	1.00E+08
38	SR	SRNPTNMC	Yes	1.00E+03	1.00E+08
39	Y	Y0NPTNMC	Yes	1.00E+03	1.00E+08
40	ZR	ZRNPTNMC	Yes	1.00E+03	1.00E+08
41	NB	NBNPTNMC	Yes	1.00E+03	1.00E+08
42	MO	MONPTNMC	Yes	1.00E+03	1.00E+08
43	TC	TCNPTNMC	Yes	1.00E+03	1.00E+08
44	RU	RUNPTNMC	Yes	1.00E+03	1.00E+08
45	RH	RHNPTNMC	Yes	1.00E+03	1.00E+08

(Cont'd)

Table G.5 MVP photoatomic data library (2/2)

Z	Element	ID in MVP	F.X. data	ELOW (eV)	EHI (eV)
46	PD	PDNPTNMC	Yes	1.00E+03	1.00E+08
47	AG	AGNPTNMC	Yes	1.00E+03	1.00E+08
48	CD	CDNPTNMC	Yes	1.00E+03	1.00E+08
49	IN	INNPTNMC	Yes	1.00E+03	1.00E+08
50	SN	SNNPTNMC	Yes	1.00E+03	1.00E+08
51	SB	SBNPTNMC	Yes	1.00E+03	1.00E+08
52	TE	TENPTNMC	Yes	1.00E+03	1.00E+08
53	I	IONPTNMC	Yes	1.00E+03	1.00E+08
54	XE	XENPTNMC	Yes	1.00E+03	1.00E+08
55	CS	CSNPTNMC	Yes	1.00E+03	1.00E+08
56	BA	BANPTNMC	Yes	1.00E+03	1.00E+08
57	LA	LANPTNMC	Yes	1.00E+03	1.00E+08
58	CE	CENPTNMC	Yes	1.00E+03	1.00E+08
59	PR	PRNPTNMC	Yes	1.00E+03	1.00E+08
60	ND	NDNPTNMC	Yes	1.00E+03	1.00E+08
61	PM	PMNPTNMC	Yes	1.00E+03	1.00E+08
62	SM	SMNPTNMC	Yes	1.00E+03	1.00E+08
63	EU	EUNPTNMC	Yes	1.00E+03	1.00E+08
64	GD	GDNPTNMC	Yes	1.00E+03	1.00E+08
65	TB	TBNPTNMC	Yes	1.00E+03	1.00E+08
66	DY	DYNPTNMC	Yes	1.00E+03	1.00E+08
67	HO	HONPTNMC	Yes	1.00E+03	1.00E+08
68	ER	ERNPTNMC	Yes	1.00E+03	1.00E+08
69	TM	TMNPTNMC	Yes	1.00E+03	1.00E+08
70	YB	YBNPTNMC	Yes	1.00E+03	1.00E+08
71	LU	LUNPTNMC	Yes	1.00E+03	1.00E+08
72	HF	HFNPTNMC	Yes	1.00E+03	1.00E+08
73	TA	TANPTNMC	Yes	1.00E+03	1.00E+08
74	W	WONPTNMC	Yes	1.00E+03	1.00E+08
75	RE	RENPTNMC	Yes	1.00E+03	1.00E+08
76	OS	OSNPTNMC	Yes	1.00E+03	1.00E+08
77	IR	IRNPTNMC	Yes	1.00E+03	1.00E+08
78	PT	PTNPTNMC	Yes	1.00E+03	1.00E+08
79	AU	AUNPTNMC	Yes	1.00E+03	1.00E+08
80	HG	HGNPTNMC	Yes	1.00E+03	1.00E+08
81	TL	TLNPTNMC	Yes	1.00E+03	1.00E+08
82	PB	PBNPTNMC	Yes	1.00E+03	1.00E+08
83	BI	BINPTNMC	Yes	1.00E+03	1.00E+08
84	PO	PONPTNMC	Yes	1.00E+03	1.00E+08
85	AT	ATNPTNMC	Yes	1.00E+03	1.00E+08
86	RN	RNNPTNMC	Yes	1.00E+03	1.00E+08
87	FR	FRNPTNMC	Yes	1.00E+03	1.00E+08
88	RA	RANPTNMC	Yes	1.00E+03	1.00E+08
89	AC	ACNPTNMC	Yes	1.00E+03	1.00E+08
90	TH	THNPTNMC	Yes	1.00E+03	1.00E+08
91	PA	PANPTNMC	Yes	1.00E+03	1.00E+08
92	U	UONPTNMC	Yes	1.00E+03	1.00E+08
93	NP	NPNPTNMC	Yes	1.00E+03	1.00E+08
94	PU	PUNPTNMC	Yes	1.00E+03	1.00E+08

G.2.2 ENDF/B-VI.8

The contents of the MVP photoatomic data library based on ENDF/B-VI.8 are listed in Table G.6. The meaning of column labels for the following table is the same as in Table G.5. The fluorescence X-ray data for plutonium ($Z=94$) is used for the heavy elements ($Z>94$) as an ad-hoc treatment.

Table G.6 MVP photoatomic data library based on ENDF/B-VI.8 (1/2)

Z	Element	ID in MVP	F.X. data	ELOW (eV)	EHI (eV)
1	H	H0NPTNB6	No	1.00E+03	8.00E+09
2	HE	HENPTNB6	No	1.00E+03	8.00E+09
3	LI	LINPTNB6	No	1.00E+03	8.00E+09
4	BE	BENPTNB6	No	1.00E+03	8.00E+09
5	B	B0NPTNB6	No	1.00E+03	8.00E+09
6	C	C0NPTNB6	No	1.00E+03	8.00E+09
7	N	N0NPTNB6	No	1.00E+03	8.00E+09
8	O	O0NPTNB6	No	1.00E+03	8.00E+09
9	F	F0NPTNB6	No	1.00E+03	8.00E+09
10	NE	NENPTNB6	No	1.00E+03	8.00E+09
11	NA	NANPTNB6	No	1.00E+03	8.00E+09
12	MG	MGNPTNB6	Yes	1.00E+03	8.00E+09
13	AL	ALNPTNB6	Yes	1.00E+03	8.00E+09
14	SI	SINPTNB6	Yes	1.00E+03	8.00E+09
15	P	P0NPTNB6	Yes	1.00E+03	8.00E+09
16	S	S0NPTNB6	Yes	1.00E+03	8.00E+09
17	CL	CLNPTNB6	Yes	1.00E+03	8.00E+09
18	AR	ARNPTNB6	Yes	1.00E+03	8.00E+09
19	K	K0NPTNB6	Yes	1.00E+03	8.00E+09
20	CA	CANPTNB6	Yes	1.00E+03	8.00E+09
21	SC	SCNPTNB6	Yes	1.00E+03	8.00E+09
22	TI	TINPTNB6	Yes	1.00E+03	8.00E+09
23	V	V0NPTNB6	Yes	1.00E+03	8.00E+09
24	CR	CRNPTNB6	Yes	1.00E+03	8.00E+09
25	MN	MNNPTNB6	Yes	1.00E+03	8.00E+09
26	FE	FENPTNB6	Yes	1.00E+03	8.00E+09
27	CO	CONPTNB6	Yes	1.00E+03	8.00E+09
28	NI	NINPTNB6	Yes	1.00E+03	8.00E+09
29	CU	CUNPTNB6	Yes	1.00E+03	8.00E+09
30	ZN	ZNNPTNB6	Yes	1.00E+03	8.04E+09
31	GA	GANPTNB6	Yes	1.00E+03	8.22E+09
32	GE	GENPTNB6	Yes	1.00E+03	8.00E+09
33	AS	ASNPTNB6	Yes	1.00E+03	8.00E+09
34	SE	SENPTNB6	Yes	1.00E+03	8.00E+09
35	BR	BRNPTNB6	Yes	1.00E+03	8.00E+09
36	KR	KRNPTNB6	Yes	1.00E+03	8.00E+09
37	RB	RBNPTNB6	Yes	1.00E+03	8.00E+09
38	SR	SRNPTNB6	Yes	1.00E+03	8.00E+09
39	Y	Y0NPTNB6	Yes	1.00E+03	8.00E+09
40	ZR	ZRNPTNB6	Yes	1.00E+03	8.00E+09
41	NB	NBNPTNB6	Yes	1.00E+03	9.33E+09
42	MO	MONPTNB6	Yes	1.00E+03	8.00E+09
43	TC	TCNPTNB6	Yes	1.00E+03	8.00E+09
44	RU	RUNPTNB6	Yes	1.00E+03	8.00E+09
45	RH	RHNPTNB6	Yes	1.00E+03	8.00E+09
46	PD	PDNPTNB6	Yes	1.00E+03	8.00E+09
47	AG	AGNPTNB6	Yes	1.00E+03	8.00E+09
48	CD	CDNPTNB6	Yes	1.00E+03	8.00E+09
49	IN	INNPTNB6	Yes	1.00E+03	8.00E+09
50	SN	SNNPTNB6	Yes	1.00E+03	8.00E+09
51	SB	SBNPTNB6	Yes	1.00E+03	8.00E+09
52	TE	TENPTNB6	Yes	1.00E+03	8.00E+09
53	I	I0NPTNB6	Yes	1.00E+03	8.00E+09
54	XE	XENPTNB6	Yes	1.00E+03	8.00E+09
55	CS	CSNPTNB6	Yes	1.00E+03	8.00E+09
56	BA	BANPTNB6	Yes	1.00E+03	8.00E+09
57	LA	LANPTNB6	Yes	1.00E+03	8.00E+09
58	CE	CENPTNB6	Yes	1.00E+03	8.00E+09
59	PR	PRNPTNB6	Yes	1.00E+03	8.00E+09
60	ND	NDNPTNB6	Yes	1.00E+03	8.00E+09
61	PM	PMNPTNB6	Yes	1.00E+03	8.00E+09
62	SM	SMNPTNB6	Yes	1.00E+03	8.00E+09
63	EU	EUNPTNB6	Yes	1.00E+03	8.00E+09

(Cont'd)

Table G.6 MVP photoatomic data library based on ENDF/B-VI.8 (2/2)

Z	Element	ID in MVP	F.X. data	ELOW (eV)	EHI (eV)
64	GD	GDNPTNB6	Yes	1.00E+03	8.00E+09
65	TB	TBNPTNB6	Yes	1.00E+03	8.00E+09
66	DY	DYNPTNB6	Yes	1.00E+03	8.00E+09
67	HO	HONPTNB6	Yes	1.00E+03	8.00E+09
68	ER	ERNPTNB6	Yes	1.00E+03	8.00E+09
69	TM	TMNPTNB6	Yes	1.00E+03	8.00E+09
70	YB	YBNPTNB6	Yes	1.00E+03	8.00E+09
71	LU	LUNPTNB6	Yes	1.00E+03	9.33E+09
72	HF	HFNPTNB6	Yes	1.00E+03	8.00E+09
73	TA	TANPTNB6	Yes	1.00E+03	8.00E+09
74	W	WONPTNB6	Yes	1.00E+03	9.66E+09
75	RE	RENPTNB6	Yes	1.00E+03	8.00E+09
76	OS	OSNPTNB6	Yes	1.00E+03	8.00E+09
77	IR	IRNPTNB6	Yes	1.00E+03	8.00E+09
78	PT	PTNPTNB6	Yes	1.00E+03	8.00E+09
79	AU	AUNPTNB6	Yes	1.00E+03	8.00E+09
80	HG	HGNPTNB6	Yes	1.00E+03	8.00E+09
81	TL	TLNPTNB6	Yes	1.00E+03	8.00E+09
82	PB	PBNPTNB6	Yes	1.00E+03	8.00E+09
83	BI	BINPTNB6	Yes	1.00E+03	8.00E+09
84	PO	PONPTNB6	Yes	1.00E+03	8.00E+09
85	AT	ATNPTNB6	Yes	1.00E+03	8.00E+09
86	RN	RNNPTNB6	Yes	1.00E+03	8.00E+09
87	FR	FRNPTNB6	Yes	1.00E+03	8.00E+09
88	RA	RANPTNB6	Yes	1.00E+03	8.00E+09
89	AC	ACNPTNB6	Yes	1.00E+03	8.00E+09
90	TH	THNPTNB6	Yes	1.00E+03	8.00E+09
91	PA	PANPTNB6	Yes	1.00E+03	8.00E+09
92	U	UONPTNB6	Yes	1.00E+03	8.00E+09
93	NP	NPNPTNB6	Yes	1.00E+03	8.00E+09
94	PU	PUNPTNB6	Yes	1.00E+03	8.00E+09
95	AM	AMNPTNB6	Yes	1.00E+03	8.00E+09
96	CM	CMNPTNB6	Yes	1.00E+03	8.00E+09
97	BK	BKNPTNB6	Yes	1.00E+03	8.00E+09
98	CF	CFNPTNB6	Yes	1.00E+03	8.00E+09
99	ES	ESNPTNB6	Yes	1.00E+03	8.00E+09
100	FM	FMNPTNB6	Yes	1.00E+03	8.00E+09

G.3 Photonuclear Data Library

The contents of the MVP photonuclear data library are listed in Table G.7. The meaning of column labels for the following table is as follows.

Z	Atomic number.
Nuclide	Nuclide name.
ID in MVP	Nuclide ID used for the specification in MVP input data.
MAT	MAT number defined in the ENDF format.
ELOW	Lowest energy of the cross section data (eV).
EHI	Highest energy of the cross section data (eV).

Table G.7 MVP photonuclear data library (1/4)

Z	Nuclide	ID in MVP	MAT	ELOW (eV)	EH1 (eV)
1	H-2	H0002JEPN	128	2.22E+06	1.40E+08
2	HE-3	HE003JEPN	225	5.49E+06	1.50E+08
3	LI-6	LI006JEPN	325	3.70E+06	1.40E+08
3	LI-7	LI007JEPN	328	2.47E+06	1.40E+08
4	BE-9	BE009CNPN	409	1.00E+06	3.00E+07
4	BE-9	BE009JEPN	425	1.66E+06	1.40E+08
5	B-10	B0010JEPN	525	4.46E+06	1.40E+08
5	B-11	B0011JEPN	528	8.66E+06	1.40E+08
6	C-12	C0012JEPN	625	7.37E+06	1.40E+08
6	C-12	C0012KAPN	625	7.37E+06	1.40E+08
6	C-12	C0012LAPN	625	7.37E+06	1.50E+08
6	C-13	C0013KAPN	628	4.95E+06	1.40E+08
7	N-14	N0014JEPN	725	7.55E+06	1.50E+08
7	N-14	N0014KAPN	725	7.55E+06	1.40E+08
7	N-15	N0015KAPN	728	1.10E+07	1.40E+08
8	O-16	O0016JEPN	825	7.16E+06	1.50E+08
8	O-16	O0016KAPN	825	7.16E+06	1.40E+08
8	O-16	O0016LAPN	825	1.25E+07	1.50E+08
8	O-17	O0017KAPN	828	4.50E+06	1.40E+08
8	O-18	O0018KAPN	831	6.23E+06	1.40E+08
9	F-19	F0019JEPN	925	4.01E+06	1.40E+08
11	NA-23	NA023KAPN	1125	8.79E+06	1.40E+08
12	MG-24	MG024KAPN	1225	9.31E+06	1.40E+08
12	MG-25	MG025KAPN	1228	7.33E+06	1.40E+08
12	MG-26	MG026KAPN	1231	1.06E+07	1.40E+08
13	AL-27	AL027CNPN	2131	5.00E+06	3.00E+07
13	AL-27	AL027KAPN	1325	8.50E+06	1.40E+08
13	AL-27	AL027LAPN	1325	8.50E+06	1.50E+08
14	SI-27	SI027KAPN	1422	7.46E+06	1.40E+08
14	SI-28	SI028KAPN	1425	9.98E+06	1.40E+08
14	SI-28	SI028LAPN	1425	9.98E+06	1.50E+08
14	SI-29	SI029JEPN	1428	8.47E+06	3.57E+08
14	SI-29	SI029KAPN	1428	8.47E+06	1.40E+08
14	SI-30	SI030JEPN	1431	1.06E+07	3.25E+08
14	SI-30	SI030KAPN	1431	1.06E+07	1.40E+08
15	P-31	P0031JEPN	1525	7.30E+06	1.40E+08
16	S-32	S0032KAPN	1625	6.95E+06	1.40E+08
16	S-33	S0033KAPN	1628	7.12E+06	1.40E+08
16	S-34	S0034KAPN	1631	7.92E+06	1.40E+08
16	S-36	S0036KAPN	1637	9.01E+06	1.40E+08
17	CL-35	CL035KAPN	1725	6.37E+06	1.40E+08
17	CL-37	CL037KAPN	1731	7.85E+06	1.40E+08
18	AR-36	AR036KAPN	1825	6.64E+06	1.40E+08
18	AR-38	AR038KAPN	1831	7.21E+06	1.40E+08
18	AR-40	AR040KAPN	1837	6.80E+06	1.40E+08
19	K-39	K0039KAPN	1925	6.38E+06	1.40E+08
19	K-40	K0040KAPN	1928	6.44E+06	1.40E+08
19	K-41	K0041KAPN	1931	6.22E+06	1.40E+08
20	CA-40	CA040JEPN	2025	7.04E+06	1.50E+08
20	CA-40	CA040KAPN	2025	9.00E+06	1.40E+08
20	CA-40	CA040LAPN	2025	7.04E+06	1.50E+08
20	CA-42	CA042KAPN	2031	6.26E+06	1.40E+08
20	CA-43	CA043KAPN	2034	7.59E+06	1.40E+08
20	CA-44	CA044KAPN	2037	8.86E+06	1.40E+08
20	CA-46	CA046KAPN	2043	1.04E+07	1.40E+08
20	CA-48	CA048KAPN	2049	9.94E+06	1.40E+08
22	TI-46	TI046KAPN	2225	8.00E+06	1.40E+08
22	TI-47	TI047KAPN	2228	8.88E+06	1.40E+08
22	TI-48	TI048KAPN	2231	9.44E+06	1.40E+08
22	TI-49	TI049KAPN	2234	8.14E+06	1.40E+08
22	TI-50	TI050KAPN	2237	1.07E+07	1.40E+08
23	V-51	V0051CNPN	2231	5.00E+06	3.00E+07
23	V-51	V0051JEPN	2325	1.11E+07	4.94E+08

(Cont'd)

Table G.7 MVP photonuclear data library (2/4)

Z	Nuclide	ID in MVP	MAT	ELOW (eV)	EH1 (eV)
24	CR-50	CR050CNPN	2241	1.00E+06	3.00E+07
24	CR-50	CR050KAPN	2425	8.56E+06	1.40E+08
24	CR-52	CR052CNPN	2242	1.00E+06	3.00E+07
24	CR-52	CR052KAPN	2431	9.35E+06	1.40E+08
24	CR-53	CR053CNPN	2243	1.00E+06	3.00E+07
24	CR-53	CR053KAPN	2434	7.94E+06	1.40E+08
24	CR-54	CR054CNPN	2244	1.00E+06	3.00E+07
24	CR-54	CR054KAPN	2437	7.93E+06	1.40E+08
25	MN-55	MN055KAPN	2525	7.93E+06	1.40E+08
26	FE-54	FE054CNPN	2261	5.00E+06	3.00E+07
26	FE-54	FE054JEPN	2625	1.31E+07	5.53E+08
26	FE-54	FE054KAPN	2625	8.42E+06	1.40E+08
26	FE-56	FE056CNPN	2262	5.00E+06	3.00E+07
26	FE-56	FE056JEPN	2628	1.12E+07	5.05E+08
26	FE-56	FE056KAPN	2631	7.61E+06	1.40E+08
26	FE-56	FE056LAPN	2631	7.61E+06	1.50E+08
26	FE-57	FE057CNPN	2263	5.00E+06	3.00E+07
26	FE-57	FE057KAPN	2634	7.32E+06	1.40E+08
26	FE-58	FE058CNPN	2264	5.00E+06	3.00E+07
26	FE-58	FE058KAPN	2637	7.65E+06	1.40E+08
27	CO-59	CO059KAPN	2725	6.94E+06	1.40E+08
28	NI-58	NI058JEPN	2825	1.22E+07	5.40E+08
28	NI-58	NI058KAPN	2825	6.40E+06	1.40E+08
28	NI-60	NI060JEPN	2831	1.14E+07	1.50E+08
28	NI-60	NI060KAPN	2831	6.29E+06	1.40E+08
28	NI-61	NI061KAPN	2834	6.47E+06	1.40E+08
28	NI-62	NI062KAPN	2837	7.02E+06	1.40E+08
28	NI-64	NI064KAPN	2843	8.12E+06	1.40E+08
29	CU-63	CU063CNPN	2291	5.00E+06	3.00E+07
29	CU-63	CU063JEPN	2925	1.10E+07	4.67E+08
29	CU-63	CU063KAPN	2925	5.78E+06	1.40E+08
29	CU-63	CU063LAPN	2925	5.78E+06	1.50E+08
29	CU-65	CU065CNPN	2292	5.00E+06	3.00E+07
29	CU-65	CU065JEPN	2928	9.91E+06	4.30E+08
29	CU-65	CU065KAPN	2931	6.79E+06	1.40E+08
30	ZN-64	ZN064JEPN	3025	1.19E+07	4.85E+08
30	ZN-64	ZN064KAPN	3025	3.96E+06	1.40E+08
30	ZN-66	ZN066KAPN	3031	4.58E+06	1.40E+08
30	ZN-67	ZN067KAPN	3034	4.79E+06	1.40E+08
30	ZN-68	ZN068KAPN	3037	5.33E+06	1.40E+08
30	ZN-70	ZN070KAPN	3043	5.96E+06	1.40E+08
32	GE-70	GE070KAPN	3225	4.09E+06	1.40E+08
32	GE-72	GE072KAPN	3231	5.00E+06	1.40E+08
32	GE-73	GE073KAPN	3234	5.30E+06	1.40E+08
32	GE-74	GE074KAPN	3237	6.29E+06	1.40E+08
32	GE-76	GE076KAPN	3243	7.51E+06	1.40E+08
38	SR-84	SR084KAPN	3825	5.17E+06	1.40E+08
38	SR-86	SR086KAPN	3831	6.35E+06	1.40E+08
38	SR-87	SR087KAPN	3834	7.32E+06	1.40E+08
38	SR-88	SR088KAPN	3837	7.91E+06	1.40E+08
38	SR-90	SR090KAPN	3843	5.11E+06	1.40E+08
40	ZR-90	ZR090CNPN	2401	5.00E+06	3.00E+07
40	ZR-90	ZR090JEPN	4025	1.20E+07	3.93E+08
40	ZR-90	ZR090KAPN	4025	6.68E+06	1.40E+08
40	ZR-91	ZR091CNPN	2402	5.00E+06	3.00E+07
40	ZR-91	ZR091KAPN	4028	5.44E+06	1.40E+08
40	ZR-92	ZR092CNPN	2403	2.97E+06	3.00E+07
40	ZR-92	ZR092KAPN	4031	2.97E+06	1.40E+08
40	ZR-93	ZR093KAPN	4034	3.33E+06	1.40E+08
40	ZR-94	ZR094CNPN	2404	3.00E+06	3.00E+07
40	ZR-94	ZR094KAPN	4037	3.75E+06	1.40E+08
40	ZR-96	ZR096CNPN	2405	4.94E+06	3.00E+07
40	ZR-96	ZR096KAPN	4043	4.94E+06	1.40E+08
41	NB-93	NB093JEPN	4125	8.83E+06	3.79E+08

(Cont'd)

Table G.7 MVP photonuclear data library (3/4)

Z	Nuclide	ID in MVP	MAT	ELOW (eV)	EH1 (eV)
41	NB-93	NB093KAPN	4125	1.93E+06	1.40E+08
41	NB-94	NB094KAPN	4128	2.30E+06	1.40E+08
42	MO-100	MO100KAPN	4249	3.17E+06	1.30E+08
42	MO-92	MO092KAPN	4225	5.61E+06	1.40E+08
42	MO-94	MO094KAPN	4231	2.07E+06	1.40E+08
42	MO-95	MO095KAPN	4234	2.24E+06	1.40E+08
42	MO-96	MO096KAPN	4237	2.76E+06	1.40E+08
42	MO-97	MO097KAPN	4240	2.85E+06	1.40E+08
42	MO-98	MO098KAPN	4243	3.27E+06	1.35E+08
46	PD-102	PD102KAPN	4625	2.12E+06	1.40E+08
46	PD-104	PD104KAPN	4631	2.60E+06	1.40E+08
46	PD-105	PD105KAPN	4634	2.89E+06	1.40E+08
46	PD-106	PD106KAPN	4637	3.23E+06	1.40E+08
46	PD-107	PD107KAPN	4640	3.54E+06	1.40E+08
46	PD-108	PD108KAPN	4643	3.85E+06	1.40E+08
46	PD-110	PD110KAPN	4649	4.44E+06	1.40E+08
47	AG-107	AG107KAPN	4725	2.81E+06	1.40E+08
47	AG-108	AG108KAPN	4728	3.08E+06	1.40E+08
47	AG-109	AG109KAPN	4731	3.30E+06	1.40E+08
48	CD-106	CD106KAPN	4825	1.64E+06	1.40E+08
48	CD-108	CD108KAPN	4831	2.28E+06	1.40E+08
48	CD-110	CD110KAPN	4837	2.87E+06	1.40E+08
48	CD-111	CD111KAPN	4840	3.30E+06	1.40E+08
48	CD-112	CD112KAPN	4843	3.48E+06	1.40E+08
48	CD-113	CD113KAPN	4846	3.87E+06	1.40E+08
48	CD-114	CD114KAPN	4849	4.10E+06	1.40E+08
48	CD-116	CD116KAPN	4855	4.81E+06	1.40E+08
50	SN-112	SN112KAPN	5025	1.83E+06	1.40E+08
50	SN-114	SN114KAPN	5031	2.63E+06	1.40E+08
50	SN-115	SN115KAPN	5034	3.20E+06	1.40E+08
50	SN-116	SN116KAPN	5037	3.37E+06	1.15E+08
50	SN-117	SN117KAPN	5040	3.77E+06	1.10E+08
50	SN-118	SN118KAPN	5043	4.06E+06	1.15E+08
50	SN-119	SN119KAPN	5046	4.40E+06	1.10E+08
50	SN-120	SN120KAPN	5049	4.81E+06	1.10E+08
50	SN-122	SN122KAPN	5055	5.66E+06	1.40E+08
50	SN-124	SN124KAPN	5061	6.69E+06	1.30E+08
51	SB-121	SB121KAPN	5125	3.07E+06	1.40E+08
51	SB-123	SB123KAPN	5131	3.92E+06	1.40E+08
52	TE-120	TE120KAPN	5225	2.85E+06	1.40E+08
52	TE-122	TE122KAPN	5231	1.08E+06	1.40E+08
52	TE-123	TE123KAPN	5234	1.53E+06	1.40E+08
52	TE-124	TE124KAPN	5237	1.85E+06	1.15E+08
52	TE-125	TE125KAPN	5240	2.25E+06	1.40E+08
52	TE-126	TE126KAPN	5243	2.55E+06	1.20E+08
52	TE-128	TE128KAPN	5249	3.18E+06	1.20E+08
52	TE-130	TE130KAPN	5255	3.75E+06	1.30E+08
53	I-127	I0127KAPN	5325	2.18E+06	1.40E+08
53	I-129	I0129KAPN	5331	2.67E+06	1.40E+08
55	CS-133	CS133KAPN	5525	2.00E+06	1.40E+08
55	CS-135	CS135KAPN	5531	2.63E+06	1.40E+08
55	CS-137	CS137KAPN	5537	3.09E+06	1.40E+08
59	PR-141	PR141KAPN	5925	3.00E+06	1.40E+08
62	SM-144	SM144KAPN	6225	8.00E+06	1.40E+08
62	SM-147	SM147KAPN	6234	6.00E+06	1.40E+08
62	SM-148	SM148KAPN	6237	8.00E+06	1.40E+08
62	SM-149	SM149KAPN	6240	5.50E+06	1.40E+08
62	SM-150	SM150KAPN	6243	7.50E+06	1.40E+08
62	SM-151	SM151KAPN	6246	5.50E+06	1.40E+08
62	SM-152	SM152KAPN	6249	8.00E+06	1.40E+08
62	SM-154	SM154KAPN	6255	1.20E+06	1.40E+08
64	GD-152	GD152JEPN	6425	8.59E+06	1.50E+08
64	GD-154	GD152JEPN	6431	8.89E+06	1.50E+08
64	GD-155	GD155JEPN	6434	6.44E+06	1.50E+08

(Cont'd)

Table G.7 MVP photonuclear data library (4/4)

Z	Nuclide	ID in MVP	MAT	ELOW (eV)	EHI (eV)
64	GD-156	GD156JEPN	6437	8.54E+06	1.50E+08
64	GD-157	GD157JEPN	6440	6.36E+06	1.50E+08
64	GD-158	GD158JEPN	6443	7.94E+06	1.50E+08
64	GD-160	GD160JEPN	6449	7.45E+06	1.50E+08
65	TB-158	TB158KAPN	6522	1.57E+05	1.40E+08
65	TB-159	TB159KAPN	6525	1.38E+05	1.40E+08
67	HO-165	HO165KAPN	6725	4.00E+06	1.40E+08
73	TA-181	TA181JEPN	7328	7.58E+06	1.60E+08
73	TA-181	TA181LAPN	7328	7.50E+06	1.50E+08
74	W-180	W0180CNPN	2741	5.00E+06	3.00E+07
74	W-182	W0182CNPN	2742	5.00E+06	3.00E+07
74	W-182	W0182JEPN	7431	8.20E+06	1.40E+08
74	W-183	W0183CNPN	2743	4.00E+06	3.00E+07
74	W-184	W0184CNPN	2744	5.00E+06	3.00E+07
74	W-184	W0184JEPN	7437	7.41E+06	1.40E+08
74	W-184	W0184LAPN	7437	7.50E+06	1.50E+08
74	W-186	W0186CNPN	2746	5.00E+06	3.00E+07
74	W-186	W0186JEPN	7443	7.20E+06	1.40E+08
79	AU-197	AU197JEPN	7925	5.78E+06	1.40E+08
79	AU-197	AU197KAPN	7925	8.00E+06	1.40E+08
80	HG-196	HG196JEPN	8025	8.85E+06	1.50E+08
80	HG-198	HG198JEPN	8031	8.49E+06	1.50E+08
80	HG-199	HG199JEPN	8034	6.66E+06	1.50E+08
80	HG-200	HG200JEPN	8037	8.03E+06	1.50E+08
80	HG-201	HG201JEPN	8040	6.23E+06	1.50E+08
80	HG-202	HG202JEPN	8043	7.75E+06	1.50E+08
80	HG-204	HG204JEPN	8049	7.50E+06	1.50E+08
82	PB-206	PB206JEPN	8225	8.09E+06	1.40E+08
82	PB-206	PB206LAPN	8231	8.00E+06	1.50E+08
82	PB-207	PB207JEPN	8228	6.74E+06	1.40E+08
82	PB-207	PB207LAPN	8234	6.50E+06	1.50E+08
82	PB-208	PB208JEPN	8231	7.37E+06	1.40E+08
82	PB-208	PB208LAPN	8237	7.50E+06	1.50E+08
83	BI-209	BI209CNPN	2831	1.00E+06	3.00E+07
83	BI-209	BI209JEPN	8325	7.40E+06	1.50E+08
83	BI-209	BI209KAPN	8325	7.00E+06	1.40E+08
90	TH-232	TH232BOPN	9032	4.80E+06	2.00E+07
92	U-233	U0233BOPN	9233	4.80E+06	2.00E+07
92	U-234	U0234BOPN	9234	5.06E+06	2.00E+07
92	U-235	U0235BOPN	9235	3.25E+06	2.00E+07
92	U-235	U0235JEPN	9228	5.30E+06	1.40E+08
92	U-236	U0236BOPN	9236	5.90E+06	2.00E+07
92	U-238	U0238BOPN	9238	3.42E+06	2.00E+07
92	U-238	U0238JEPN	9237	5.00E+06	1.40E+08
93	NP-237	NP237JEPN	9346	5.30E+06	1.40E+08
94	PU-238	PU238BOPN	9438	5.00E+06	2.00E+07
94	PU-239	PU239BOPN	9439	5.34E+06	2.00E+07
94	PU-241	PU241BOPN	9441	4.00E+06	2.00E+07

G.4 Electron Library

The contents of the MVP electron library is listed in Table G.8. The MVP electron library is generated from the MCNP electron cross section data file EL1.³⁴⁾ The user does not have to specify element ID's to use these data.

The meaning of column labels for the following table is listed as follows.

Z	Atomic number.
Element	Symbol of the element.

ID in MVP	Element ID used for the specification in MVP input data.
ELOW	Lowest energy of the cross section data (eV).
EHI	Highest energy of the cross section data (eV).

Table G.8 MVP electron library (1/2)

Z	Element	ID in MVP	ELOW (eV)	EHI (eV)
1	H	H0NELEL1	1.00E+03	1.00E+08
2	HE	HENELEL1	1.00E+03	1.00E+08
3	LI	L1NELEL1	1.00E+03	1.00E+08
4	BE	BENELEL1	1.00E+03	1.00E+08
5	B	B0NELEL1	1.00E+03	1.00E+08
6	C	C0NELEL1	1.00E+03	1.00E+08
7	N	N0NELEL1	1.00E+03	1.00E+08
8	O	O0NELEL1	1.00E+03	1.00E+08
9	F	F0NELEL1	1.00E+03	1.00E+08
10	NE	NENELEL1	1.00E+03	1.00E+08
11	NA	NANELEL1	1.00E+03	1.00E+08
12	MG	MGNELEL1	1.00E+03	1.00E+08
13	AL	ALNELEL1	1.00E+03	1.00E+08
14	SI	SINELEL1	1.00E+03	1.00E+08
15	P	P0NELEL1	1.00E+03	1.00E+08
16	S	S0NELEL1	1.00E+03	1.00E+08
17	CL	CLNELEL1	1.00E+03	1.00E+08
18	AR	ARNELEL1	1.00E+03	1.00E+08
19	K	K0NELEL1	1.00E+03	1.00E+08
20	CA	CANELEL1	1.00E+03	1.00E+08
21	SC	SCNELEL1	1.00E+03	1.00E+08
22	TI	TINELEL1	1.00E+03	1.00E+08
23	V	V0NELEL1	1.00E+03	1.00E+08
24	CR	CRNELEL1	1.00E+03	1.00E+08
25	MN	MNNELEL1	1.00E+03	1.00E+08
26	FE	FENELEL1	1.00E+03	1.00E+08
27	CO	CONELEL1	1.00E+03	1.00E+08
28	NI	NINELEL1	1.00E+03	1.00E+08
29	CU	CUNELEL1	1.00E+03	1.00E+08
30	ZN	ZNNELEL1	1.00E+03	1.00E+08
31	GA	GANELEL1	1.00E+03	1.00E+08
32	GE	GENELEL1	1.00E+03	1.00E+08
33	AS	ASNELEL1	1.00E+03	1.00E+08
34	SE	SENELEL1	1.00E+03	1.00E+08
35	BR	BRNELEL1	1.00E+03	1.00E+08
36	KR	KRNELEL1	1.00E+03	1.00E+08
37	RB	RBNELEL1	1.00E+03	1.00E+08
38	SR	SRNELEL1	1.00E+03	1.00E+08
39	Y	Y0NELEL1	1.00E+03	1.00E+08
40	ZR	ZRNELEL1	1.00E+03	1.00E+08
41	NB	NBNELEL1	1.00E+03	1.00E+08
42	MO	MONELEL1	1.00E+03	1.00E+08
43	TC	TCNELEL1	1.00E+03	1.00E+08
44	RU	RUNELEL1	1.00E+03	1.00E+08
45	RH	RHNELEL1	1.00E+03	1.00E+08
46	PD	PDNELEL1	1.00E+03	1.00E+08
47	AG	AGNELEL1	1.00E+03	1.00E+08
48	CD	CDNELEL1	1.00E+03	1.00E+08
49	IN	INNELEL1	1.00E+03	1.00E+08
50	SN	SNNELEL1	1.00E+03	1.00E+08
51	SB	SBNELEL1	1.00E+03	1.00E+08
52	TE	TENELEL1	1.00E+03	1.00E+08
53	I	I0NELEL1	1.00E+03	1.00E+08
54	XE	XENELEL1	1.00E+03	1.00E+08
55	CS	CSNELEL1	1.00E+03	1.00E+08
56	BA	BANELEL1	1.00E+03	1.00E+08

(Cont'd)

Table G.8 MVP electron library (2/2)

Z	Element	ID in MVP	ELOW (eV)	EHI (eV)
57	LA	LANELEM1	1.00E+03	1.00E+08
58	CE	CENELEM1	1.00E+03	1.00E+08
59	PR	PRNELEM1	1.00E+03	1.00E+08
60	ND	NDNELEM1	1.00E+03	1.00E+08
61	PM	PMNELEM1	1.00E+03	1.00E+08
62	SM	SMNELEM1	1.00E+03	1.00E+08
63	EU	EUNELEM1	1.00E+03	1.00E+08
64	GD	GDNELEM1	1.00E+03	1.00E+08
65	TB	TBNELEM1	1.00E+03	1.00E+08
66	DY	DYNELEM1	1.00E+03	1.00E+08
67	HO	HONELEM1	1.00E+03	1.00E+08
68	ER	ERNELEM1	1.00E+03	1.00E+08
69	TM	TMNELEM1	1.00E+03	1.00E+08
70	YB	YBNELEM1	1.00E+03	1.00E+08
71	LU	LUNELEM1	1.00E+03	1.00E+08
72	HF	HFNELEM1	1.00E+03	1.00E+08
73	TA	TANELEM1	1.00E+03	1.00E+08
74	W	WONELEM1	1.00E+03	1.00E+08
75	RE	RENELEM1	1.00E+03	1.00E+08
76	OS	OSNELEM1	1.00E+03	1.00E+08
77	IR	IRNELEM1	1.00E+03	1.00E+08
78	PT	PTNELEM1	1.00E+03	1.00E+08
79	AU	AUNELEM1	1.00E+03	1.00E+08
80	HG	HGNELEM1	1.00E+03	1.00E+08
81	TL	TLNELEM1	1.00E+03	1.00E+08
82	PB	PBNELEM1	1.00E+03	1.00E+08
83	BI	BINELEM1	1.00E+03	1.00E+08
84	PO	PONELEM1	1.00E+03	1.00E+08
85	AT	ATNELEM1	1.00E+03	1.00E+08
86	RN	RNNELEM1	1.00E+03	1.00E+08
87	FR	FRNELEM1	1.00E+03	1.00E+08
88	RA	RANELEM1	1.00E+03	1.00E+08
89	AC	ACNELEM1	1.00E+03	1.00E+08
90	TH	THNELEM1	1.00E+03	1.00E+08
91	PA	PANELEM1	1.00E+03	1.00E+08
92	U	UONELEM1	1.00E+03	1.00E+08
93	NP	NPNELEM1	1.00E+03	1.00E+08
94	PU	PUNELEM1	1.00E+03	1.00E+08

G.5 Dosimetry Library

The MVP dosimetry library is the same as the FSXDOS-J3 library⁶⁶⁾ for the MCNP code based on the JENDL dosimetry file. The contents of the MVP dosimetry library is listed in Table G.9.

The meaning of column labels for the following table is listed as follows.

ZAID	Nuclide identifier used in the MCNP code. The form of a ZAID is ZZZAAA.nnX, where ZZZ is the atomic number, AAA is the atomic mass number, nn is the unique evaluation identifier and X is the class of data.
MT	MT number defined in the ENDF format.
Target	Target nuclide.
Reaction	Type of the nuclear reaction.
Product	Product from the nuclear reaction.

EMIN	Lowest energy of the cross section data (MeV).
EMAX	Highest energy of the cross section data (MeV).
NES	Number of energy points at which cross sections are given.
MAT	MAT number defined in the ENDF format.

Table G.9 MVP dosimetry library (1/2)

ZAID	MT	Target	Reaction	Product	EMIN (MeV)	EMAX (MeV)	NES	MAT
3006.03Y	105	Li-6	(n,t)	He-4	1.00E-11	2.00E+01	403	331
3006.03Y	207	Li-6	a production		1.00E-11	2.00E+01	422	331
3007.03Y	205	Li-7	t production		2.82E+00	2.00E+01	18	332
5010.03Y	107	B-10	(n,a)	Li-7	1.00E-11	2.00E+01	440	531
5010.03Y	207	B-10	a production		1.00E-11	2.00E+01	423	531
9019.03Y	16	F-19	(n,2n)	F-18	1.10E+01	2.00E+01	19	931
11023.03Y	16	Na-23	(n,2n)	Na-22	1.30E+01	2.00E+01	13	1131
11023.03Y	102	Na-23	(n,g)	Na-24	1.00E-11	2.00E+01	1228	1131
12024.03Y	103	Mg-24	(n,p)	Na-24	4.93E+00	2.00E+01	41	1231
13027.03Y	103	Al-27	(n,p)	Mg-27	1.90E+00	2.00E+01	38	1331
13027.03Y	107	Al-27	(n,a)	Na-24	3.25E+00	2.00E+01	72	1331
15031.03Y	103	P-31	(n,p)	Si-31	7.32E-01	2.00E+01	22	1531
16032.03Y	103	S-32	(n,p)	P-32	9.57E-01	2.00E+01	31	1631
21045.03Y	102	Sc-45	(n,g)	Sc-46	1.00E-11	2.00E+01	9572	2131
22000.03Y	210	Ti-nat	(n,x)	Sc-46	1.62E+00	2.00E+01	32	2230
22000.03Y	211	Ti-nat	(n,x)	Sc-47	1.00E-11	2.00E+01	38	2230
22000.03Y	212	Ti-nat	(n,x)	Sc-48	3.28E+00	2.00E+01	27	2230
22046.03Y	103	Ti-46	(n,p)	Sc-46	1.62E+00	2.00E+01	31	2231
22047.03Y	28	Ti-47	(n,np)	Sc-46	1.07E+01	2.00E+01	13	2232
22047.03Y	103	Ti-47	(n,p)	Sc-47	1.00E-11	2.00E+01	30	2232
22048.03Y	28	Ti-48	(n,np)	Sc-47	1.17E+01	2.00E+01	17	2233
22048.03Y	103	Ti-48	(n,p)	Sc-48	3.28E+00	2.00E+01	26	2233
22049.03Y	28	Ti-49	(n,np)	Sc-48	1.25E+00	2.00E+01	11	2234
25055.03Y	16	Mn-55	(n,2n)	Mn-54	1.04E+01	2.00E+01	12	2531
25055.03Y	102	Mn-55	(n,g)	Mn-56	1.00E-11	2.00E+01	7355	2531
26054.03Y	103	Fe-54	(n,p)	Mn-54	1.00E-11	2.00E+01	35	2631
26056.03Y	103	Fe-56	(n,p)	Mn-56	2.97E+00	2.00E+01	32	2632
26057.03Y	28	Fe-57	(n,np)	Mn-56	1.08E+01	2.00E+01	10	2633
26058.03Y	102	Fe-58	(n,g)	Fe-59	1.00E-11	2.00E+01	4731	2634
27059.03Y	16	Co-59	(n,2n)	Co-58	1.06E+01	2.00E+01	14	2731
27059.03Y	102	Co-59	(n,g)	Co-60	1.00E-11	2.00E+01	10701	2731
27059.03Y	107	Co-59	(n,a)	Mn-56	1.00E-11	2.00E+01	16	2731
28058.03Y	16	Ni-58	(n,2n)	Ni-57	1.24E+01	2.00E+01	16	2831
28058.03Y	103	Ni-58	(n,p)	Co-58	1.00E-11	2.00E+01	35	2831
28060.03Y	103	Ni-60	(n,p)	Co-60	2.08E+00	2.00E+01	29	2832
29063.03Y	16	Cu-63	(n,2n)	Cu-62	1.10E+01	2.00E+01	10	2931
29063.03Y	102	Cu-63	(n,g)	Cu-64	1.00E-11	2.00E+01	15081	2931
29063.03Y	107	Cu-63	(n,a)	Co-60	1.00E-11	2.00E+01	23	2931
29065.03Y	16	Cu-65	(n,2n)	Cu-64	1.01E+01	2.00E+01	11	2932
30064.03Y	103	Zn-64	(n,p)	Cu-64	1.00E-11	2.00E+01	25	3031
40090.03Y	16	Zr-90	(n,2n)	Zr-89	1.21E+01	2.00E+01	10	4031
41093.03Y	16	Nb-93	(n,2n)	Nb-92m	9.05E+00	2.00E+01	12	4131
41093.03Y	51	Nb-93	(n,n')	Nb-93m	3.07E-02	2.00E+01	58	4131
45103.03Y	51	Rh-103	(n,n')	Rh-103m	4.00E-02	2.00E+01	70	4531
49115.03Y	51	In-115	(n,n')	In-115m	3.20E-01	2.00E+01	388	4931
49115.03Y	102	In-115	(n,g)	In-116m	1.00E-11	2.00E+01	13392	4931
53127.03Y	16	I-127	(n,2n)	I-126	9.22E+00	2.00E+01	16	5331
63151.03Y	102	Eu-151	(n,g)	Eu-152	1.00E-11	2.00E+01	4016	6331
73181.03Y	102	Ta-181	(n,g)	Ta-182	1.00E-11	2.00E+01	22138	7331
74186.03Y	102	W-186	(n,g)	W-187	1.00E-11	2.00E+01	8865	7431
79197.03Y	16	Au-197	(n,2n)	Au-196	8.11E+00	2.00E+01	14	7931
79197.03Y	102	Au-197	(n,g)	Au-198	1.00E-11	2.00E+01	10941	7931
80199.03Y	57	Hg-199	(n,n')	Hg-199m	5.34E-01	2.00E+01	49	8031
90232.03Y	18	Th-232	fission	-	1.00E-11	2.00E+01	92	9031

(Cont'd)

Table G.9 MVP dosimetry library (2/2)

ZAID	MT	Target	Reaction	Product	EMIN (MeV)	EMAX (MeV)	NES	MAT
90232.03Y	102	Th-232	(n,g)	Th-233	1.00E-11	2.00E+01	18519	9031
92235.03Y	18	U-235	fission	-	1.00E-11	2.00E+01	3987	9231
92238.03Y	18	U-238	fission	-	1.00E-11	2.00E+01	3330	9232
92238.03Y	102	U-238	(n,g)	U-239	1.00E-11	2.00E+01	32318	9232
93237.03Y	18	Np-237	fission	-	1.00E-11	2.00E+01	6230	9331
94239.03Y	18	Pu-239	fission	-	1.00E-11	2.00E+01	9832	9431
95241.03Y	18	Am-241	fission	-	1.00E-11	2.00E+01	7328	9531

H. Nuclear Reactions in MVP libraries

H.1 Neutron Reactions

Table H.1 lists the reaction IDs defined in MVP neutron libraries. The IDs are the same as the MT number in the ENDF format. The ID numbers are assigned up to 120. MTPAR is a pair reaction ID.

Table H.1 Neutron reactions in MVP neutron libraries

ID/MT	MTPAR	comments	ID/MT	MTPAR	comments	ID/MT	MTPAR	comments
1	0	total	41	-1		81	81	(n,n')31th ex.
2	2	elastic	42	-1		82	82	(n,n')32th ex.
3	0	nonelastic	43	-1		83	83	(n,n')33th ex.
4	0	total inel.	44	-1		84	84	(n,n')34th ex.
5	-1	—	45	-1		85	85	(n,n')35th ex.
6	46	(n,2n) 1st ex.	46	46	(n,2n) 1th ex.	86	86	(n,n')36th ex.
7	47	(n,2n) 2nd ex.	47	47	(n,2n) 2th ex.	87	87	(n,n')37th ex.
8	48	(n,2n) 3rd ex.	48	48	(n,2n) 3th ex.	88	88	(n,n')38th ex.
9	49	(n,2n) 4th ex.	49	49	(n,2n) 4th ex.	89	89	(n,n')39th ex.
10	-1	—	50	-1		90	90	(n,n')40th ex.
11	-1	—	51	51	(n,n') 1st ex.	91	91	(n,n') continuum
12	-1	—	52	52	(n,n') 2nd ex.	92	92	thermal inela.
13	-1	—	53	53	(n,n') 3rd ex.	93	93	thermal elastic
14	-1	—	54	54	(n,n') 4th ex.	94	0	total (n,2n)
15	-1	—	55	55	(n,n') 5th ex.	95	0	total (n,3n)
16	16	(n,2n) direct	56	56	(n,n') 6th ex.	96	0	total (n,4n)
17	17	(n,3n)	57	57	(n,n') 7th ex.	97	0	total threshold (n,Nn')
18	18	total fission	58	58	(n,n') 8th ex.	98	18	delayed neutron
19	-1	(n,f)	59	59	(n,n') 9th ex.	99	-1	
20	-1	(n,n'f)	60	60	(n,n')10th ex.	100	0	kerma factor
21	-1	(n,2n'f)	61	61	(n,n')11th ex.	101	0	capture
22	22	(n,n'a)	62	62	(n,n')12th ex.	102	0	(n,g)
23	23	(n,n'3a)	63	63	(n,n')13th ex.	103	0	(n,p)
24	24	(n,2na)	64	64	(n,n')14th ex.	104	0	(n,d)
25	25	(n,3na)	65	65	(n,n')15th ex.	105	0	(n,t)
26	-1	(n,2n)isomeric	66	66	(n,n')16th ex.	106	0	(n,he-3)
27	-1	absorption	67	67	(n,n')17th ex.	107	0	(n,a)
28	28	(n,n'p)	68	68	(n,n')18th ex.	108	0	(n,2a)
29	29	(n,n'2a)	69	69	(n,n')19th ex.	109	0	(n,3a)
30	30	(n,2n2a)	70	70	(n,n')20th ex.	110	-1	—
31	-1	—	71	71	(n,n')21th ex.	111	0	(n,2p)
32	32	(n,n'd)	72	72	(n,n')22th ex.	112	0	(n,pa)
33	33	(n,n't)	73	73	(n,n')23th ex.	113	0	(n,t2a)
34	34	(n,n'he 3)	74	74	(n,n')24th ex.	114	0	(n,d2a)
35	35	(n,n'd2a)	75	75	(n,n')25th ex.	115	0	gamma data
36	36	(n,n't2a)	76	76	(n,n')26th ex.	116	-1	—
37	37	(n,4n)	77	77	(n,n')27th ex.	117	-1	—
38	-1	(n,3nf)	78	78	(n,n')28th ex.	118	-1	—
39	-1	—	79	79	(n,n')29th ex.	119	-1	—
40	-1	—	80	80	(n,n')30th ex.	120	-1	—

(n,2n) ⇒ MT=6,7,8,9,16,24,30 : (n,3n) ⇒ MT=17,25 : (n,4n) ⇒ MT=37 : (n,inelastic) ⇒ not MT=4, but cross section emitting 1 neutron except elastic capture ⇒ MT=102 – 116 not including MT=117

H.2 Photoatomic Reactions

Table H.2 lists the reaction IDs defined in MVP photoatomic data libraries. The ID numbers are reserved up to 60, which include unassigned reactions.

Table H.2 Photoatomic reactions in MVP photoatomic data libraries

ID	MT	comments
1	501	total photon interaction
2	502	photon coherent scattering
3	0	incoherent scattering by Klein-Nishina
4	504	photon incoherent scattering
5	515	pair production, electron field
6	516	pair production
7	517	pair production, nuclear field
8	522	photoelectric absorption
9 – 52	0	(not assigned)
53	0	averaged heating number
54 – 60	0	(not assigned)

H.3 Photonuclear Reactions

Table H.3 lists the reaction IDs defined in MVP photonuclear data libraries. The ID numbers are reserved up to 135, which include unassigned reactions.

Table H.3 Photonuclear reactions in MVP photonuclear data libraries

ID	MT	comments	ID	MT	comments	ID	MT	comments
1	0	total	51	51	(g,n') 1th ex.	101	-1	
2	-1		52	52	(g,n') 2th ex.	102	0	(g,g')
3	0	total reaction	53	53	(g,n') 3th ex.	103	103	(g,p)
4	4	total (g,ln)	54	54	(g,n') 4th ex.	104	104	(g,d)
5	5	anything	55	55	(g,n') 5th ex.	105	105	(g,t)
6	-1		56	56	(g,n') 6th ex.	106	106	(g,He-3)
7	-1		57	57	(g,n') 7th ex.	107	107	(g,a)
8	-1		58	58	(g,n') 8th ex.	108	108	(g,2a)
9	-1		59	59	(g,n') 9th ex.	109	109	(g,3a)
10	-1		60	60	(g,n')10th ex.	110	-1	
11	-1		61	61	(g,n')11th ex.	111	111	(g,2p)
12	-1		62	62	(g,n')12th ex.	112	-1	
13	-1		63	63	(g,n')13th ex.	113	-1	
14	-1		64	64	(g,n')14th ex.	114	-1	
15	-1		65	65	(g,n')15th ex.	115	-1	
16	16	(g,2n)	66	66	(g,n')16th ex.	116	-1	
17	17	(g,3n)	67	67	(g,n')17th ex.	117	-1	
18	18	total fission	68	68	(g,n')18th ex.	118	201	total neutron
19	-1		69	69	(g,n')19th ex.	119	202	total gamma-ray
20	-1		70	70	(g,n')20th ex.	120	203	total proton
21	-1		71	71	(g,n')21th ex.	121	204	total deuteron
22	22	(g,na)	72	72	(g,n')22th ex.	122	205	total triton
23	-1		73	73	(g,n')23th ex.	123	206	total He3
24	-1		74	74	(g,n')24th ex.	124	207	total alpha
25	-1		75	75	(g,n')25th ex.	125	208	total pi+
26	-1		76	76	(g,n')26th ex.	126	209	total pi0
27	-1		77	77	(g,n')27th ex.	127	210	total pi-
28	28	(g,np)	78	78	(g,n')28th ex.	128	211	total mu+
29	-1		79	79	(g,n')29th ex.	129	212	total mu-
30	-1		80	80	(g,n')30th ex.	130	213	total k+
31	-1		81	81	(g,n')31th ex.	131	214	total k0(long)
32	-1		82	82	(g,n')32th ex.	132	215	total k0(short)
33	-1		83	83	(g,n')33th ex.	133	216	total k-
34	-1		84	84	(g,n')34th ex.	134	217	total anti-proton
35	-1		85	85	(g,n')35th ex.	135	218	total anti-neutron
36	-1		86	86	(g,n')36th ex.			
37	-1		87	87	(g,n')37th ex.			
38	-1		88	88	(g,n')38th ex.			
39	-1		89	89	(g,n')39th ex.			
40	-1		90	90	(g,n')40th ex.			
41	-1		91	91	(g,n')continuum			
42	-1		92	-1				
43	-1		93	600	(g,p')ground			
44	-1		94	650	(g,d')ground			
45	-1		95	700	(g,t')ground			
46	-1		96	750	(g,He-3')ground			
47	-1		97	800	(g,a')ground			
48	-1		98	-1	delayed neutron			
49	-1		99	-1				
50	50	(g,n')ground	100	-1	KERMA			

Index

■ Symbols ■

$S(\alpha, \beta)$, 37

#CELL

data sub-block, 146, 176

#CIRCLE – sampling function, 206

#COSINE – sampling function, 202

#DISC – sampling function, 205

#EVAPORATION – sampling function, 204

#FEYNMAN – sampling function, 208

#FISSION – sampling function, 205

#FISSIONFILE – sampling function, 207

#GAUSS – sampling function, 203

#ISOTROPIC – sampling function, 206

#MAXWELL – sampling function, 203

#NEWAXIS – data-transform function, 213

#NORMALIZE – data-transform function, 213

#POWER – sampling function, 204

#RESIZE – data-transform function, 212

#ROTATE2D – data-transform function, 211

#ROTATE3D – data-transform function, 212

#SPHERE – sampling function, 206

#SUBFRAME

data sub-block, 184

#SWAP – data-transform function, 211

#TABLE – sampling function, 199

#TABLEXY – sampling function, 201

#TALLY REGION

data sub-block, 146, 191

#TYPE n – sampling function, 208

#UNIFORM – sampling function, 202

#USER n – sampling function, 209

#WATT – sampling function, 203

■ A ■

ADAPTIVE-MICRO-CALCULATION option, 86, 93

ADJOINT option, 86, 93

AMLIM – control data, 124

ANALOG – tally data, 246

analog absorption, 12, 34, 124

analog estimator, 40

ANALOG-FISSION option, 86, 94, 231, 246

ANGLE – tally data, 225

angular bin, 225, 226

angular flux, 4

ANISN, 134

ANISN-type multigroup cross sections, 140, 275

ARB – BODY, 153

ASSIM, 330

file, 311

AUTO-RESTART option, 86, 94

■ B ■

B – CGVIEW command, 336

BACK – CGVIEW command, 336

background job, 305

BANKP – control data, 125

batch, 17, 115, 116

BBC – BODY, 162

BDY – CGVIEW command, 336

BEFF

perturbation data, 265

BETA-EFFECTIVE option, 87, 94

binary output

file, 359

of calculation results, 284

BODIEs, 8, 49

ARB, 153

BBC, 162

BODY-by-BODY combination, 162

BOX, 152, 176

CYL, 149

cylinder with specified radial direction, 156

cylinder with the central axis parallel to the z -axis, 149

definition of —, 146

ELL, 159

elliptical torus, 160

ELT, 160, 225

GEL, 160

general ellipsoid, 160

general hexagonal prism, 155

general quadratic curved surface, 161

general right circular cylinder, 150

GQS, 161

HAF, 154

half space, 154

HEX, 155, 176

- parallelepiped, 152
 - polyhedron, 153
 - RAW, 152
 - RCC, 150
 - RCL, 156, 176
 - RHP, 155, 176
 - right hexagonal prism with the central line parallel to the z -axis, 155
 - right parallelepiped, 149
 - RPP, 149, 176
 - SPH, 151
 - sphere, 151
 - spheroid, 159
 - TEC, 158
 - TRC, 157
 - truncated elliptic cone, 158
 - truncated right cone, 157
 - WED, 152
 - wedge, 152
 - BODY, 49
 - BODY – CGVIEW command, 336
 - BODY ID, 146, 164
 - BODY-by-BODY combination, 50
 - Boltzmann equation, 4
 - BOUNDARY – CGVIEW command, 336
 - boundary condition, 10, 54, 163
 - isotropic reflective, 11
 - perfect reflective, 11, 54
 - periodic, 11, 55
 - vacuum, 11
 - boundary of energy groups, 224
 - bounded sphere approach, 15
 - BOX – BODY, 152, 176
 - BREMSSTRAHLUNG option, 87, 95
 - Bremsstrahlung photon, 47
- C ■**
- CELL, 8, 146
 - CGVIEW command, 337
 - definition of —, 55
 - definition of —s, 176
 - geometry data, 187
 - type of —s
 - BOX, 177
 - HEXA, 177
 - MBASE, 177
 - STGP, 177, 188
 - CELL-arrays coordinates, 167
 - CELLDIRS – geometry data, 170
 - CELLIDS – geometry data, 170
 - CELLSYMS – geometry data, 170
 - CENTER
 - CGVIEW command, 337
 - central limit theorem, 16
 - CGV_PRBASE – environment variable, 344
 - CGV_PRINT – environment variable, 344
 - CGVIEW, 25, 331
 - -SLICE, 331
 - environment variables
 - CGV_PRBASE, 344
 - CGV_PRINT, 344
 - input data for —, 332
 - interactive mode, 353
 - CGVIEW commands, 336
 - B, 336
 - BACK, 336
 - BDY, 336
 - BODY, 336
 - BOUNDARY, 336
 - CELL, 337
 - CENTER, 337
 - CHECK, 337
 - DOVERLAP, 337
 - DOVL, 337
 - EXIT, 345
 - F, 338
 - FL, 338
 - FLOOD, 338
 - FORGET, 338
 - FORWARD, 338
 - GRANGE, 339
 - H, 339
 - HELP, 339
 - IZC, 339
 - IZCOLOR, 339
 - IZCR, 339
 - LEVEL, 340
 - LINES, 340
 - M, 341
 - MA, 341
 - MAT, 341
 - MC, 341
 - MCOLOR, 341
 - MCR, 341
 - MOVE, 341
 - MOVEA, 341

MX, 341	ZS, 352
MXA, 341	ZSEARCH, 352
MY, 341	ZX, 350
MYA, 341	ZY, 350
MZ, 341	ZZ, 351
MZA, 341	CHECK – CGVIEW command, 337
NOSUBF, 349	CKLATT – geometry data, 170
OVERLAP, 342	CKSLAT – geometry data, 170
OVL, 342	cluster environment, 26
PAINT, 342	CNAMES – geometry data, 185
PAPER, 342	coherent scattering, 46
PNT, 342	collision estimator, 13, 17, 230
PRINT, 343	COLLISION-LESS option, 87, 95
QUIT, 345	combinatorial geometry, 8, 49, 164
RC, 345	OR, 164
RCOLOR, 345	command search path, 304
RCR, 345	command-line parameter, 271
RD, 345	comment, 75
REDRAW, 345	COMPOSITION – GMVP cross section data, 144
ROT, 346	composition data, 127
ROTX, 346	Compton scattering, 45
ROTY, 346	continuous-energy method, 11
SCAN, 347	control data
SIZE or XMAX, 348	common to MVP and GMVP
SPTYP, 349	DEPS, 121
STYLE, 348	DINF, 121, 149–151, 155, 156
SUBF, 349	ELOOP, 122
TITLE, 349	IRAND, 119
WAITOFF, 349	MXPGEN, 121
WAITON, 349	NBANK, 116
WHAT, 349	NBPINT, 118
WOFF, 349	NFBANK, 119
WON, 349	NGP1, 117
XMAX, 348	NGP2, 117
XY, 350	NGROUP(outside of the data block), 116
XZ, 350	NGROUP.N, 117
YX, 350	NGROUP.N[EUTRON], 117
YZ, 350	NGROUP.P, 117
Z, 350	NGROUP.P[HOTON], 117
ZB, 351	NHIST, 101, 115, 119
ZC, 350	NMEMO, 120
ZCOLOR, 350	NMEMOP, 120
ZCR, 350	NMEMS, 120
ZM, 350	NMENO, 165
ZONE, 350	NPART, 115
ZOOM, 350	NRSINT, 118
ZOOMBODY, 351	NRSKIP, 120
ZOOMZONE, 351	NSKIP, 119

- NTIME(outside of the TALLY data block), 117
- NTMINT, 118
- TCPU, 119
- TCUT, 121
- WLLIM, 122
- MVP only
 - AMLIM, 124
 - BANKP, 125
 - EBOT.P, 123
 - EBOT[.N], 123
 - ETHMAX, 124
 - ETOP.P, 123
 - ETOP[.N], 123
 - EW CUT, 124
 - MB, 125
- PICTURE option
 - NPICT, 125
 - PAPER, 126
- COPYNNND – geometry data, 187
- COSINE – tally data, 226
- cosine distribution, 202
- CPU time, 119
- CRESP – tally data, 228
- cross section data
 - GMVP
 - COMPOSITION, 144
 - FKAI, 144
 - IDMAT, 144
 - IHT, 140
 - INCELM, 141
 - INGP, 140
 - INPUT, 143
 - ISGG, 140
 - ITBL, 139
 - MUS, 143
 - NDS1, 138
 - NDS2, 139
 - NGP1, 138
 - NGP2, 138
 - NPL, 141
 - NSCT, 141
 - OUTPUT, 143
 - PLFORM, 136
 - PRFLAG, 142
 - TITLE, 135
 - TYPE, 136
 - XLAST, 144
 - MVP
 - IDMAT, 127
 - JPNLCP, 132
 - JVPOP, 130, 131
 - JXPOOL, 130, 132
 - TEMPMT, 133, 134
 - TPRECS, 130, 131
 - CROSS SECTION data block, 205
 - GMVP, 135
 - MVP, 127
 - cross section index file, 273
 - cross section library
 - arbitrary-temperature neutron —, 29, 127, 130
 - photon, 30
 - CSPACE – geometry data, 185
 - current
 - surface crossing estimator, 15, 230
 - CYGIN, 330
 - CYL – BODY, 149
- D ■
 - data block, 77
 - CROSS SECTION, 127, 135, 205
 - GEOMETRY, 146, 166
 - SOURCE, 194
 - TALLY, 223
 - XSEC, 127
 - DELAYED-NEUTRON option, 87, 95
 - DEPS, 54
 - DEPS – control data, 121
 - detailed model, 45
 - differential-form transport equation, 4
 - DIMENSION – tally data, 231
 - DINF – BODY, 156
 - DINF – control data, 121, 149–151, 155
 - DIRSYMS – geometry data, 170
 - discrete conditional sampling method, 37
 - DOPPLER-SCATTERING option, 87, 96
 - dosimetry data file, 228
 - double-differential cross section, 275
 - DOVERLAP – CGVIEW command, 337
 - DOVL – CGVIEW command, 337
 - DTF-IV format, 135
 - dynamic memory allocation, 311
 - DYNAMIC-MEMORY option, 87, 97
- E ■
 - EBOT.P – control data, 123
 - EBOT[.N] – control data, 123

- EDIT-BY-COLLISION option, 87, 98
 - EDIT-BY-TRACK-LENGTH option, 87, 98
 - EDIT-MACROSCOPIC-DATA option, 87, 99
 - EDIT-MICROSCOPIC-DATA option, 88, 100
 - EIGEN-VALUE option, 88, 90, 100
 - eigenvalue problems, 22
 - elastic scattering, 35
 - electron mass, 45
 - electron pair production, 45
 - element ID, 129
 - ELL – BODY, 159
 - ELOOP – control data, 122
 - ELT – BODY, 160, 225
 - energy groups, 223
 - ENGYB.N – tally data, 223
 - ENGYB.P – tally data, 223
 - environment variables
 - MVP_ARTPOOL, 132
 - MVP_DIR, 304, 319, 320
 - MVPHOST, 304, 319–321
 - MVPLIB_DIR, 304, 319, 320
 - estimation of eigenvalue
 - neutron balance, 40
 - production, 40
 - estimation of real variance, 23, 40
 - estimator, 13
 - ETHMAX – control data, 124
 - ETOP.P – control data, 123
 - ETOP[.N] – control data, 123
 - evaporation model, 36
 - evaporation spectrum, 204
 - event, 17
 - EVENT – perturbation data, 265
 - EVENT – tally data, 230
 - event stack, 26
 - event-based algorithm, 25
 - EWCUT – control data, 35, 124
 - EXIT – CGVIEW command, 345
 - exponential transform, 20
- F ■**
- F=*BODY-ID*, 176
 - f2c, 330
 - FAT, 372
 - Feynman's Y value, 42
 - Feynman- α method, 42
 - FEYNMAN-Y – tally data, 247
 - fictitious material, 10, 54, 163
 - isotropic reflective, 11
 - mirror reflective, 11
 - perfect reflective, 11
 - white reflective, 11
 - FIDO format, 134, 136
 - fission neutron, 34
 - fission neutron bank, 119
 - FISSION option, 88, 100
 - fission source file, 207, 301, 308
 - fission spectrum, 34
 - FISSION-MULTIPLICITY option, 88, 101
 - FIXED-SOURCE option, 88, 100, 102
 - fixed-source problem, 22
 - FKAI
 - GMVP cross section data, 144
 - secondary particle data, 258
 - FL – CGVIEW command, 338
 - FLOOD – CGVIEW command, 338
 - fluorescent X-ray, 45
 - flux
 - surface crossing estimator, 230
 - FLUX-PRINT option, 88, 102
 - FNND1 – geometry data, 187
 - FNND2 – geometry data, 187
 - FNND3 – geometry data, 187
 - FO – CGVIEW command, 338
 - FORGET – CGVIEW command, 338
 - FORWARD – CGVIEW command, 338
 - FRACTIONAL-CHANGE
 - perturbation data, 267
 - FRAME ZONE, 102, 174, 184
 - definition of —, 56
 - name, 184
 - FRAME-DEPENDENT-TALLY option, 91, 184
 - FRAME=*BODY-ID*, 176
 - free gas model, 36, 124
 - free monatomic gas model, 36
 - FREE-LATTICE-FRAME option, 88, 102, 174, 176, 180
 - FreeBSD, 330
- G ■**
- Gauss distribution, 203
 - GEL – BODY, 160
 - GENERATION – tally data, 238
 - geometry data
 - CELL
 - CNAMES, 185

CSPACE, 185
 ID, 177
 NAMES, 184
 SPACE, 185
 TYPE, 177
 LATTICE
 CELLDIRS, 170
 CELLIDS, 170
 CELLSYMS, 170
 CKLATT, 170
 CKSLAT, 170
 DIRSYMS, 170
 IDLAT, 167, 174
 KLATT, 63, 168
 KSLAT, 168
 KSLATT, 63
 LTYP, 167
 NVLAT, 167
 RCELL, 168
 SZHEX, 167
 SZLAT, 167
 STGM region
 CELL, 187
 COPYNND, 187
 FNND1, 187
 FNND2, 187
 FNND3, 187
 IDLAT, 186
 LTYP, 186
 MBASE, 187
 PCELL, 187
 PF, 187
 WNND1, 187
 WNND2, 187
 WNND3, 187
 GEOMETRY data block, 146, 166
 getsystem, 304
 gmview3, 352
 GMVPLBCV, 371
 GQS – BODY, 161
 GRANGE – CGVIEW command, 339
 group-wise response function, 228

■ H ■

H – CGVIEW command, 339
 HAF – BODY, 154
 HELP – CGVIEW command, 339
 HEX – BODY, 155, 176

hexagonal prism LATTICE, 9
 history, 17, 115
 history-based algorithm, 25

■ I ■

I/O unit, 271
 IANGLE – tally data, 236
 IANGLE2 – tally data, 237
 ID
 geometry data, 177
 perturbation data, 264
 source data, 195
 tally data, 230
 IDLAT – geometry data, 167, 174, 186
 IDMAT – GMVP cross section data, 144
 IDMAT – MVP cross section data, 127
 IENERGY – tally data, 234
 IENERGY2 – tally data, 235
 IGENE2 – tally data, 238
 IHT – GMVP cross section data, 140
 ImageMagick, 345
 import, 345
 IMAGINARY-PARTICLE option, 89
 IMPORTANCE option, 89, 103, 114, 116, 252
 INCELM – GMVP cross section data, 141
 incoherent scattering, 45
 index file, 305
 infinite mass model, 36
 INGP – GMVP cross section data, 140
 initial distribution, 22
 initial random number, 119
 INPUT – GMVP cross section data, 143
 INPUT-ZONE, 52, 164
 IRAND – control data, 119
 IRESP – tally data, 243
 ISGG – GMVP cross section data, 140
 ITBL – GMVP cross section data, 139
 ITIME – tally data, 235
 ITIME2 – tally data, 236
 IZC – CGVIEW command, 339
 IZCOLOR – CGVIEW command, 339
 IZCR – CGVIEW command, 339

■ J ■

JPNLCP – MVP cross section data, 132
 JVPOP – MVP cross section data, 130, 131
 JXPOOL – MVP cross section data, 130
 JXPOOL — MVP cross section data, 132

■ K ■

KEFF
 perturbation data, 265
 KENO, 4, 12
 KLATT – geometry data, 168
 Klein-Nishina differential cross section, 45
 Klein-Nishina formula, 46
 KSLAT – geometry data, 168

■ L ■

L=*LATTICE-ID*, 163
 LABEL – perturbation data, 265
 LABEL – tally data, 230
 LAT=*LATTICE-ID*, 163
 LATTICE
 definition of —, 56
 LATTICE coordinates, 167
 LATTICE FRAME, 8
 definition of —, 56
 LATTICE geometry, 8, 208
 input data for —, 166
 LATTICE ID, 164, 167, 174
 LATTICE option, 89, 104, 146, 166
 LEVEL – CGVIEW command, 340
 LICEM, 29
 linear-linear interpolation, 29
 LINES – CGVIEW command, 340
 Linux, 330
 log-linear interpolation, 29
 lost particle, 54, 277
 LTYP – geometry data, 167, 186

■ M ■

M – CGVIEW command, 341
 MA – CGVIEW command, 341
 MACRO – tally data (neutron), 243
 MACRO – tally data (photon), 244
 macroscopic cross section bank, 243, 244
 MAIL, 134, 136
 makemake, 321
 MARKER-REGION – tally data, 239
 MAT – CGVIEW command, 341
 material composition data, 144
 material ID, 54, 163
 mathematical expression
 in the input data, 78
 Maxwell distribution, 203
 MB – control data, 125
 MBASE – geometry data, 187

MC – CGVIEW command, 341
 MCNP, 4, 12
 MCOLOR – CGVIEW command, 341
 MCR – CGVIEW command, 341
 METHOD
 perturbation data, 266
 method of maximum likelihood, 40
 MICRO – tally data (neutron), 244
 MICRO – tally data (photon), 245
 MICROPN – tally data, 245
 microscopic cross sections
 calculation of —, 93
 saving calculation time of —, 93
 MONITOR option, 89, 104
 MONPNWGT — secondary particle data, 262
 MORSE, 4, 12, 51
 MOVE – CGVIEW command, 341
 MOVEA – CGVIEW command, 341
 MPI, 27
 MS Windows, 330
 MULTI-TASK option, 89, 104
 multigroup method, 11
 multiply defined ZONE, 53
 MUS – GMVP cross section data, 143
 mview3, 352
 MVP_ARTPOOL – environment variable, 132
 MVPFAT, 372
 mvpmake, 321, 323
 MX – CGVIEW command, 341
 MXA – CGVIEW command, 341
 MXPGEN – control data, 121
 MY – CGVIEW command, 341
 MYA – CGVIEW command, 341
 MZ – CGVIEW command, 341
 MZA – CGVIEW command, 341

■ N ■

NAMES – geometry data, 184
 NBANK – control data, 116
 NBPINT – control data, 118
 NDS1 – GMVP cross section data, 138
 NDS2 – GMVP cross section data, 139
 nearest neighbor distribution, 68, 187
 nested LATTICE geometry, 56
 netpbm, 344
 NEUTRON
 perturbation data, 265
 source data, 195

tally data, 231
 NEUTRON option, 89, 105
 NFBANK – control data, 119
 NFS, 318
 NGP1
 control data, 117
 GMVP cross section data, 138
 NGP2
 control data, 117
 GMVP cross section data, 138
 NGROUP – control data, 116
 NGROUP.N
 control data, 117
 tally data, 223
 NGROUP.N[EUTRON] – control data, 117
 NGROUP.P
 control data, 117
 tally data, 223
 NGROUP.P[HOTON] – control data, 117
 NHIST – control data, 23, 101, 115, 119
 NLB2TXT, 368, 369
 NMEMO – control data, 120, 165
 NMEMOP – control data, 120
 NMEMS – control data, 120
 NND, 68, 187
 NND1, 187
 NND2, 187
 NND3, 187
 non-analog absorption, 12, 34
 NOSUBF – CGVIEW command, 349
 NPART – control data, 115
 NPDET – tally data, 226
 NPICT – control data, 125
 NPL – GMVP cross section data, 141
 NRESP – tally data, 228
 NRSINT – control data, 118
 NRSKIP – control data, 120
 NSCT – GMVP cross section data, 141
 NSKIP – control data, 23, 119
 NSTAL – tally data, 228
 NTIME
 control data, 117
 tally data, 224
 NTMINT – control data, 118
 NTXT2LB, 368
 nuclide ID, 128, 129, 205
 NVLAT – geometry data, 167
 NWORK – tally data, 229

■ O ■

OBJ.\$MVPHOST, 330
 optical path length, 5
 optical thickness, 5
 option
 BETA-EFFECTIVE, 87
 DELAYED-NEUTRON, 87, 95
 PERTURBATION, 89, 106
 PHOTO-NUCLEAR, 107
 SCATTERING-MATRIX, 91, 112
 SCATTERING-MUBAR, 91, 112
 option block, 76
 options
 ADAPTIVE-MICRO-CALCULATION, 86, 93
 ADJOINT, 86, 93
 ANALOG-FISSION, 86, 94, 231, 246
 AUTO-RESTART, 86, 94
 BETA-EFFECTIVE, 94
 BREMSSTRAHLUNG, 87, 95
 COLLISION-LESS, 87, 95
 conflict between —, 91
 DOPPLER-SCATTERING, 87, 96
 DYNAMIC-MEMORY, 87, 97, 311
 EDIT-BY-COLLISION, 87, 98
 EDIT-BY-TRACK-LENGTH, 87, 98
 EDIT-MACROSCOPIC-DATA, 87, 99
 EDIT-MICROSCOPIC-DATA, 88, 100
 EIGEN-VALUE, 88, 90, 100
 FISSION, 88, 100
 FISSION-MULTIPLICITY, 88, 101
 FIXED-SOURCE, 88, 100, 102
 FLUX-PRINT, 88, 102
 FRAME-DEPENDENT-TALLY, 91, 113, 184
 FRAME-DEPENDENT-TALLY option, 113
 FREE-LATTICE-FRAME, 66, 88, 102, 174, 176, 180
 IMAGINARY-PARTICLE, 89
 IMPORTANCE, 89, 103, 114, 116, 252
 LATTICE, 89, 104, 146, 166
 MONITOR, 89, 104
 MULTI-TASK, 89, 104
 NEUTRON, 89, 105
 PATH-STRETCH, 89, 105, 254
 PERIODIC-TIME, 89, 106
 PHOTO-NUCLEAR, 90
 PHOTON, 89, 95, 106
 PICTURE, 90, 108, 125
 PRINT-SUPPRESS, 90, 108

- RELATIVE-WEIGHT, 90, 109, 122
 - REPEATED-GEOMETRY, 89, 104
 - RESPONSE, 90, 109
 - RESTART, 109
 - RESTART-FILE, 90, 110
 - RUN-MODE, 90, 111
 - RUSSIAN-ROULETTE, 91, 111, 114, 224, 253
 - SOURCE-OUTPUT, 91, 112, 207
 - TALLY-LATTICE, 82, 91, 113, 184
 - TIME-DEPENDENT, 91, 106, 113, 121
 - WEIGHT-WINDOW, 91, 104, 111, 113, 116, 224, 253
 - options for variance reduction, 22
 - outer iteration, 23
 - OUTPUT – GMVP cross section data, 143
 - OVERLAP – CGVIEW command, 342
 - OVL – CGVIEW command, 342
- P ■**
- PAINT – CGVIEW command, 342
 - PAPER
 - CGVIEW command, 342
 - control data, 126
 - parallel calculation, 26
 - PARAMETER
 - perturbation data, 266
 - PARTICLE – tally data, 231
 - particle bank, 26, 116, 125, 258
 - particle source, 7
 - PATH line, 274
 - path stretching, 20, 105
 - PATH-STRETCH option, 89, 105, 254
 - PCELL – geometry data, 187
 - perfect absorber, 163
 - periodic boundary condition, 165
 - PERIODIC-TIME option, 89, 106
 - perturbation
 - calculation, 264
 - calculation results, 282
 - input for — data
 - BEFF, 265
 - EVENT, 265
 - FRACTIONAL-CHANGE, 267
 - ID, 264
 - KEFF, 265
 - LABEL, 265
 - METHOD, 266
 - NEUTRON, 265
 - PARAMETER, 266
 - PERTURBED-REGION, 267
 - PERTURBED-SOURCE, 267
 - option, 89, 106
 - PERTURBATION Data Block, 264
 - specification of perturbed nuclide, 267
 - PERTURBED-REGION
 - perturbation data, 267
 - PERTURBED-SOURCE
 - perturbation data, 267
 - PF – geometry data, 187
 - PFCN – variance reduction data, 255
 - PHOTO-NUCLEAR option, 90, 107
 - photoatomic reaction, 44
 - cross section, 30
 - photoelectric effect, 45
 - PHOTON
 - source data, 195
 - tally data, 231
 - PHOTON option, 89, 95, 106
 - photonuclear reaction, 44
 - cross section, 31
 - analysis of —, 47
 - input of — cross section data, 129
 - input of secondary particle data
 - MONPNWGT, 262
 - WGTPN, 260
 - input of tally data
 - DIMENSION, 232
 - MICROPN, 245
 - PNPRODUCE, 229
 - PRODUCE-NUCLIDE, 240
 - PRODUCE-REACTION, 242
 - PRODUCE-REGION, 240
 - input of variance reduction data
 - PFCN, 255
 - PMT, 256
 - PPNBR, 255
 - input option for cross section data, 132
 - option, 90, 107
 - PICTURE option, 90, 108, 125
 - PLFORM – GMVP cross section data, 136
 - PMT — variance reduction data, 256
 - PNPRODUCE – tally data, 229
 - PNT – CGVIEW command, 342
 - POINT – tally data, 234
 - point detector, 226, 227, 230, 232
 - estimator, 14

position of —, 226
 power distribution, 204
 PPNBR – variance reduction data, 255
 PRESP – tally data, 243
 PRFLAG – GMVP cross section data, 142
 primary particle, 138, 258
 PRINT – CGVIEW command, 343
 PRINT-SUPPRESS option, 90, 108
 probability table method, 29
 processing element, 27
 PRODUCE-NUCLIDE – tally data, 240
 PRODUCE-REACTION – tally data, 242
 PRODUCE-REGION – tally data, 240
 PSALP – variance reduction data, 21, 254
 PSXYZ – variance reduction data, 21, 254

■ Q ■

Q-value, 36
 QUIT – CGVIEW command, 345

■ R ■

random walk, 7, 8, 12, 13, 17
 RATIO – source data, 195
 RAW – BODY, 152
 RC – CGVIEW command, 345
 RCC – BODY, 150
 RCELL – geometry data, 168
 RCL – BODY, 156, 176
 RCOLOR – CGVIEW command, 345
 RCR – CGVIEW command, 345
 RD – CGVIEW command, 345
 reactor noise analysis, 42
 specification for each tally, 246
 real variance
 definition of —, 41
 rectangular parallelepiped LATTICE, 9
 REDRAW – CGVIEW command, 345
 reflective material, 163
 REGION, 49, 233
 name, 163, 185
 number, 82
 tally data, 233
 relation between the current and the flux, 15
 RELATIVE-WEIGHT option, 90, 109, 122
 REPEATED-GEOMETRY option, 89, 104
 resolved resonance region, 29
 resonance elastic scattering, 37
 asymptotic model, 37, 96

asymptotic slowing-down scattering model, 37, 96
 constant cross section model, 38, 96
 exact model, 38, 96
 free gas model with constant cross section, 38
 free gas model with resonance cross section, 38
 free-monatomic-gas model with constant cross section, 96
 free-monatomic-gas model with resonance cross section, 96
 RESP.N – tally data, 228
 RESP.P – tally data, 228
 response function
 continuous energy, 229
 RESPONSE option, 90, 109
 restart, 94
 restart calculation, 115, 116
 restart file, 110, 118, 301, 305, 308–310
 RESTART option, 109
 RESTART-FILE option, 90, 110
 RHP – BODY, 155, 176
 ROOT-CELL, 59
 ROOT-SUBFRAME, 59
 ROT – CGVIEW command, 346
 rotation of a vector, 212
 ROTX – CGVIEW command, 346
 ROTY – CGVIEW command, 346
 RPP – BODY, 149, 176
 RUN-MODE option, 90, 111
 rungmvp, 309, 328
 runmvp, 305, 306, 326
 Russian roulette, 19, 258
 — with importance, 20
 — with weight, 20
 RUSSIAN-ROULETTE option, 91, 111, 114, 224, 253
 RVOL – volume data, 250

■ S ■

sampling specification line, 195
 SCAN – CGVIEW command, 347
 scattering cosine tally, 112
 calculation result, 281
 option, 112
 scattering matrix
 — moments, 112
 — tally, 112

- calculation results, 280
- group-to-group scattering rate, 112
- option, 112
- SCATTERING-MATRIX option, 91, 112
- SCATTERING-MUBAR option, 91, 112
- secondary particle, 125, 139, 258
- secondary particle data
 - FKAI
 - outside of data blocks, 258
 - MONPNWGT, 262
 - WGTF, 22, 23, 34, 101, 119, 224, 258
 - WGTP, 224, 260
 - WGTPN, 260
- simple model, 45
- SIZE – CGVIEW command, 348
- skipped batch, 23
- source, 7
- SOURCE – tally data, 239
- source data
 - ACCEPT, 214
 - biased sampling, 216
 - data-transform function, 197, 198, 211
 - #NEWAXIS, 213
 - #NORMALIZE, 213
 - #RESIZE, 212
 - #ROTATE2D, 211
 - #ROTATE3D, 212
 - #SWAP, 211
 - END-ACCEPT, 214
 - ID, 195
 - IN-BODY, 216
 - IN-ZONE, 216
 - NEUTRON, 195
 - NOT-IN-BODY, 216
 - NOT-IN-ZONE, 216
 - PHOTON, 195
 - RATIO, 195
 - sampling function, 197–199
 - #CIRCLE, 206
 - #COSINE, 202
 - #DISC, 205
 - #EVAPORATION, 204
 - #FEYNMAN, 208
 - #FISSION, 205
 - #FISSIONFILE, 207
 - #GAUSS, 203
 - #ISOTROPIC, 206
 - #MAXWELL, 203
 - #POWER, 204
 - #SPHERE, 206
 - #TABLE, 199
 - #TABLEXY, 201
 - #TYPE_n, 208
 - #UNIFORM, 202
 - #USER_n, 209
 - #WATT, 203
 - source information set, 194
 - user-defined sampling function, 209
 - WHEN, 214
- SOURCE data block, 194
- source set, 120
- SOURCE-OUTPUT option, 91, 112, 207, 301
- SOURCE-REGION – tally data, 239
- SPACE – geometry data, 185
- SPDET – tally data, 227
- special tally, 361
- speed of neutrons, 39
- speed of photons, 48
- SPH – BODY, 151
- splitting, 19, 20, 115, 116
- SPTYP – CGVIEW command, 349
- stack, 26
- STAL – tally data, 228
- statistical geometry model, 9, 68, 176, 185
 - allocation of STGM regions, 188
 - matrix medium in the STGM region, 186
 - MBASE, 186
 - packing fraction, 187
 - STG particle, 185, 188
 - STGM region, 69, 185, 186
 - MATRIX, 189
 - SUBFRAME name, 188
 - STGP CELL, 186
- statistical independence, 17
- statistical uncertainty, 15
- STYLE – CGVIEW command, 348
- SUBF – CGVIEW command, 349
- SUBFRAME
 - definition of —, 56
- SUBFRAME name, 184
- surface crossing estimator, 225
 - current, 15, 230
 - flux, 230
- symbolic parameter, 78
- SZHEX – geometry data, 167
- SZLAT – geometry data, 167

■ T ■

tally, 13

tally data

ANALOG, 246

ANGLE, 225

COSINE, 226

CRESP, 228

DIMENSION, 231

ENGYB.N, 223

ENGYB.P, 223

EVENT, 230

FEYNMAN-Y, 247

GENERATION, 238

IANGLE, 236

IANGLE2, 237

ID, 230

IENERGY, 234

IENERGY2, 235

IGENE2, 238

IRESP, 243

ITIME, 235

ITIME2, 236

LABEL, 230

MACRO – neutron reaction, 243

MACRO – photoatomic reaction, 244

MARKER-REGION, 239

MICRO

SPREAD, 244, 245

MICRO – neutron reaction, 244

MICRO – photoatomic reaction, 245

MICROPN, 245

NEUTRON, 231

NGROUP.N, 223

NGROUP.P, 223

NPDET, 226

NRESP

in the TALLY data block, 228

NSTAL, 228

NTIME

in the TALLY data block, 224

NWORK, 229

PARTICLE, 231

PHOTON, 231

PNPRODUCE, 229

POINT, 234

PRESP, 243

PRODUCE-NUCLIDE, 240

PRODUCE-REACTION, 242

PRODUCE-REGION, 240

REGION, 233

RESP.N, 228

RESP.P, 228

SOURCE, 239

SOURCE-REGION, 239

SPDET, 227

STAL, 228

TALLY-SURFACE, 225

TIMEB, 121, 224

WTIME

in the TALLY data block, 224

XPDET, 226

TALLY data block, 223, 361

TALLY REGION, 71, 191, 233

ADD type, 192

DEFINE type, 192

TALLY-LATTICE option, 91, 113, 184

TALLY-SURFACE – tally data, 225

task, 25

TCPU – control data, 119

TCUT – control data, 106, 121

TEC – BODY, 158

temperature

specification of — for each nuclide, 134

template directory, 77

TEMPMT – MVP cross section data, 133, 134

Terrell's distribution, 42

thermal neutron scattering, 36, 124

thermal scattering cross section data, 36

thick target approximation, 31, 47

Thomson scattering, 46

time bin, 224

TIME-DEPENDENT option, 91, 106, 113, 121

TIMEB – tally data, 121, 224

TITLE

CGVIEW commands, 349

title

for input data, 75

TITLE – GMVP cross section data, 135

TPRECS – MVP cross section data, 130, 131

track length estimator, 13, 230

TRC – BODY, 157

TRVOL – volume data, 250

TYPE

geometry data, 177

GMVP cross section data, 136

■ U ■

unbiased estimator, 16
 undefined region, 54
 unresolved resonance region, 29

■ V ■

variance, 16, 17
 variance reduction data
 PFCN, 255
 PMT, 256
 PPNBR, 255
 PSALP, 254
 PSXYZ, 254
 WKIL, 224, 253
 WKIL.N, 253
 WKIL.P, 253
 WSRV, 224, 253
 WSRV.N, 253
 WSRV.P, 253
 XIMP, 252
 XIMP.N, 252
 XIMP.P, 252
 variance reduction parameter, 252
 variance reduction technique, 17
 variance reduction techniques, 252
 variance-to-mean ratio, 42
 vector calculation, 25, 116
 VIM, 4, 12
 VOL – volume data, 249
 volume
 REGION, 249
 TALLY REGION, 249
 ZONE, 249

■ W ■

WAITOFF – CGVIEW command, 349
 WAITON – CGVIEW command, 349
 Watt's fission spectrum, 203
 WED – BODY, 152
 weight, 12–14, 18
 weight window, 22
 WEIGHT-WINDOW option, 91, 104, 111, 113, 116, 224, 253
 WGTF – secondary particle data, 101, 119, 224, 258
 WGTP – secondary particle data, 224, 260
 WGTPN – secondary particle data, 260
 WHAT – CGVIEW command, 349
 wild card character, 85, 205, 369, 370

WKIL – variance reduction data, 224, 253
 WKIL.N – variance reduction data, 253
 WKIL.P – variance reduction data, 253
 WLLIM – control data, 122
 WNND1 – geometry data, 187
 WNND2 – geometry data, 187
 WNND3 – geometry data, 187
 WOFF – CGVIEW command, 349
 WON – CGVIEW command, 349
 WSRV – variance reduction data, 224, 253
 WSRV.N – variance reduction data, 253
 WSRV.P – variance reduction data, 253
 WTIME – tally data, 224

■ X ■

XIMP – variance reduction data, 252
 XIMP.N – variance reduction data, 252
 XIMP.P – variance reduction data, 252
 XLAST – GMVP cross section data, 144
 XMAX – CGVIEW command, 348
 XPDET – tally data, 226
 XSEC data block
 MVP, 127
 xwd, 344
 XY – CGVIEW command, 350
 XZ – CGVIEW command, 350

■ Y ■

YX – CGVIEW command, 350
 YZ – CGVIEW command, 350

■ Z ■

Z – CGVIEW command, 350
 ZB – CGVIEW command, 351
 ZC – CGVIEW command, 350
 ZCOLOR – CGVIEW command, 350
 ZCR – CGVIEW command, 350
 ZM – CGVIEW command, 350
 ZONE, 8, 49
 input data for —s, 162
 name, 163
 search, 8, 120, 165
 ZONE – CGVIEW command, 350
 ZOOM – CGVIEW command, 350
 ZOOMBODY – CGVIEW command, 351
 ZOOMZONE – CGVIEW command, 351
 ZS – CGVIEW command, 352
 ZSEARCH – CGVIEW command, 352
 ZX – CGVIEW command, 350

ZY – CGVIEW command, 350

ZZ – CGVIEW command, 351

Source Code of MVP/GMVP and Auxiliary Programs

- Source files of MVP

aaaloc.f	acaloc.f	action.f	actmpp.f	actsgl.f	actsmp.f
actvpp.f	artlib.f	beff.f	blkiso.f	bprogv.f	cadenz.f
center.f	check.f	chkenv.f	chkstk.f	chsymb.f	ckpool.f
clrstk.f	ctxsin.f	cxsint.f	delayed.f	dentb2.f	dumpbk.f
dumpst.f	dumpxx.f	editfl.f	egychk.f	elgen.f	elmntr.f
faloc.f	faloce.f	falocf.f	falocp.f	falocpn.f	fisgen.f
fismin.f	fisset.f	fissrc.f	fissrcpt.f	flagin.f	fldist.f
flight.f	flione.f	fregas.f	getmac.f	getmic.f	gmacnx.f
gmacpt.f	gmicn1.f	gmicnx.f	gmicp1.f	gmicpn1.f	gmicpt.f
gtdate.c	gtmacp.f	gtmacpn.f	gtmicp.f	gtmicpn.f	gtsctl.f
gtvval.f	igtflg.f	inpidx.f	intmpn.f	intro.f	intro2.f
isnds.f	keff.f	keff0.f	keffpt.f	kpinit.f	kpsymb.f
main.f	mmchk.f	mntcpu.f	monpnwinp.f	mtdata.f	mtdatp.f
mtsums.f	mttask.f	natomz.f	neesel.f	neutr.f	nid2nn.f
nthscf.f	nucfis.f	nuclst.f	nxphr.f	nxphr2.f	nxphr3.f
nxphr4.f	nxttee.f	nxtfli.f	nxtmvt.f	nxtnr.f	nxtnr2.f
nxtnr3.f	nxtnr4.f	nxtths.f	option.f	panlgf.f	pfcninp.f
pfcnnuc.f	photnuc.f	photr.f	phtgen.f	pmtinp.f	pn4lct2.f
pnbrinp.f	pnkinm.f	pnuncl.f	popara.f	prdate.c	preinp.f
prmntr.f	prnspc.f	prsize.f	prsour.f	qppl.f	rangee.f
resti0.f	restin.f	resto.f	resto0.f	resto1.f	restot.f
restrt.f	seaone.f	search.f	sef5n.f	sef5n0.f	sef5n2.f
sef5n3.f	sef5n4.f	sef6n3.f	select.f	selone.f	source.f
splitb.f	srcold.f	srcul.f	srcu2.f	staln1.f	staln3.f
staln7.f	staln9.f	stkidt.f	suncl.f	talkp.f	tally.f
tallyo.f	tallyp.f	talprt.f	talsme.f	talsmf5.f	talsmf6.f
talspt.f	talsum.f	thsct.f	ttbr.f	verstr.f	vmntr.f
wgtfdinp.f	wgtpninp.f	wrkary.f	wwdprs.f	xsec.f	xsecd.f
xsece.f	xsecp.f	xsecpn.f	xsecpp.f	xsfile.f	zzbank.f
zzcustom.f	zzinit.f	zztalo.f	zztals.f		

- Include files of MVP

_ATPOOL	_CXSEC	_FBANK2	_FLAGS	_IOUNIT2	_ISOTOP
_KPIDS	_KPMASS	_KPSYMS	_NGPS	_NMON	_PERT
_PMNTR	_PROGV	_PSOUR	_PTALY	_PTALY2	_PXSEC
_REACC	_SBANK	_SIZES2	_STACK	_WKC2A	_WKCOL
_WKFL1	_WKFLA	_WKFSS	_WKLAT	_WKMIR	_WKNXT
_WKPHT	_WKSE1	_WKSEA	_WKSOU	_WKTLO	_WKTLS
_XBANK	_XWORK				

- Source files of GMVP

aaaloc.f	acaloc.f	action.f	actmpp.f	actsgl.f	actsmp.f
actvpp.f	adjrev.f	angles.f	apmatx.f	badmom.f	bprogv.f
cadenz.f	center.f	check.f	chkenv.f	chknum.f	chsymb.f
colisn.f	dumpbk.f	dumpst.f	dumpxx.f	ffread.f	find.f
fisgen.f	fisset.f	fissrc.f	flagin.f	flight.f	flione.f
gamgen.f	getmus.f	gtdate.c	gtndsk.f	gtvval.f	gtxsec.f
igtflg.f	intro.f	intro2.f	jinput.f	keff0.f	kpinit.f
kpsymb.f	legend.f	macro1.f	macro2.f	macro3.f	macro4.f
macro5.f	main.f	mament.f	mntcpu.f	mtsums.f	mttask.f
neesel.f	neesum.f	ngvel.f	nid2nn.f	nucfis.f	nuclst.f
nxtec.f	nxtee.f	nxtfli.f	option.f	pgvel.f	popara.f
prdate.c	preinp.f	prmntr.f	prnspc.f	prsize.f	prsour.f
q.f	readsg.f	rearag.f	resti0.f	restin.f	resto.f
resto0.f	resto1.f	restor.f	restot.f	restrt.f	seaone.f
search.f	select.f	selone.f	source.f	srcold.f	srcul.f
srcu2.f	staln1.f	staln3.f	staln7.f	staln9.f	store.f
suncl.f	tallyo.f	talsum.f	verstr.f	vmntr.f	wrkary.f
xseccn.f	xsecin.f	xsgmvp.f	xstape.f	zzbank.f	zzcustom.f
zzinit.f	zztalo.f	zztals.f			

- Include files of GMVP

_FBANK2	_FLAGS	_IOUNIT2	_KPIDS	_KPSYMS	_NGPS
_PMNTR	_PROGV	_PSOUR	_PTALY	_PTALY2	_PXSEC
_SIZES2	_STACK	_WKC2A	_WKC0L	_WKFL1	_WKFLA
_WKFSS	_WKLAT	_WKMIR	_WKNXT	_WKSE1	_WKSEA
_WKSOU	_WKTLO	_WKTLS	_XBANK	_XWORK	

- Source files common to MVP and GMVP

abs2zn.f	acta2z.f	angset.f	arrayin.f	aryii4.f	aryinv.f
aryir4.f	aryir8.f	arysum.f	ascti4.f	atrans.f	atrns2.f
bmcmk.f	bmcmk0.f	bmcmk1.f	bodarb.f	bodbbc.f	bodbox.f
bodcyl.f	bodell.f	bodelt.f	bodgel.f	bodgqs.f	bodhaf.f
bodhex.f	bodrcc.f	bodrcl.f	bodrhpf.f	bodrpp.f	bodsph.f
bodtec.f	bodtrc.f	bodwed.f	bsdec1.f	bsdec3.f	bsincl.f
bsinc3.f	bsvdec.f	bsvinc.f	catstr.f	cbsinc.f	ccomp.f
cel2ab.f	celtr.f	celtyp.f	chkdup.f	chkngp.f	cizon.f
ckalp.f	ckbats.f	cknul0.f	ckodr4.f	ckval4.f	ckval8.f
ckwght.f	cmpchi.f	cnterr.f	cnvttox.f	crsinp.f	crvol.f
crvoll.f	crvolt.f	dframe.f	dotmax.f	feynmf.f	flsurf.f
fnssc2.f	fnssoc.f	fssamp.f	geomin.f	grconv.f	gtsour.f
hextr0.f	i4prnt.f	ibtile.f	iiiset.f	inplst.f	intplc.f
iorder.f	iounud.f	istrcm.f	itrjsp.f	judge.f	keep.f
keep2.f	keepc.f	kgpset.f	kpsort.f	kzrstr.f	latdw2.f
latdwn.f	lattice.f	lattin.f	lattr.f	lattyf.f	latup2.f
latupw.f	lbzcnt.f	lframe.f	lkeep.f	lsiz.f	lthrgy.f
memalc.f	memerr.f	memmem.f	memuse.f	mid2nm.f	mirror.f
mrzone.f	mvpcommu.f	mvpssync.f	namlst.f	neefli.f	neesea.f
nestck.f	ngpchk.f	nndcpy.f	oldsrc.f	opensc.f	pablo.f
packet.f	pcuml2.f	pesum.f	picaso.f	pkkai.f	pnorm.f
prein0.f	presrc.f	prlat0.f	prlatt.f	prnspd.f	prnuwe.f
prpara.f	prstop.f	prtxyz.f	pt2zon.f	ptinck.f	ptinp.f
putv.f	r4prnt.f	r8prnt.f	rdsubp.f	realvr.f	refzon.f
regdat.f	reglst.f	regnam.f	regnm0.f	report.f	reqhst.f
rgfind.f	rskip.f	rstbod.f	rvlatt.f	rxpdet.f	scfinp.f
scfout.f	seqstr.f	sinklt.f	sizchk.f	smpfyn.f	smpinf.f
smplx1.f	smplx2.f	smplx3.f	smpstg.f	spcchk.f	spear1.f
spnums.f	spreg.f	sqsol1.f	srcinf.f	srcinp.f	srecin.f
srecot.f	stamp.f	stlout.f	stlprx.f	stlsum.f	stprn1.f
stprn2.f	ststfn.f	sumon.f	sumupx.f	syssrc.f	talin1.f
talin2.f	talin3.f	talinp.f	talsrf.f	tblxy.f	tlereg.f
tlstio.f	tregnm.f	tskctl.f	tskmpi.f	tskpvm.f	tsksm.f
typbod.f	usrsrc.f	value0.f	varreg.f	vdentk.f	walias.f
wdbl.f	wkaa2z.f	wsizof.f	zlbz.f	zonein.f	

- Include files common to MVP and GMVP

_ARRAY	_COUNTS	_FRDATA	_IDXOFF	_IUNIT	_LINEAR
_LISTOFF	_LISTON	_LNAM	_PGEOM	_PMLATT	_PTALY0
_PTLSP	_PVMPARA	_PVRED	_RAND	_SFLATT	_SIZES
_STALY	_TASKDT	_TIMEDT	_VPPARA	_WKAB2Z	_WORDL

- Source files of CGVIEW

aaalloc.f	acalloc.f	action.f	astoeb.c	calplot.c	canvs2.f
center.f	cgparam.f	cgvinit.f	chkenv.f	ckdefsrc.f	cnum.f
dmblock.f	dmsour.f	dmxsec.f	dumpbk.f	dumpst.f	flagin.f
flichk.f	flione.f	flyend.f	fxcheck.f	gpbody.f	gpcell.f
gpzone.f	gr2paper.f	graint.f	graphd.f	graphe.f	graphs.f
gtvval.f	helppg.f	igtflg.f	inptl.f	intro.f	inum.f
jnumber.f	jsymbol.f	kpinit.f	kpsymb.f	lostch.f	main.f
mcncso.f	mcnnid.f	mcnout.f	mcnsck.f	newp.f	nucfis.f
option.f	ovlpmsg.f	pablo1.f	paper.f	pifext.c	preinp.f
prgmck.f	prsize.f	psdraw.c	pssetattr.c	pssymchar.c	seaone.f
selone.f	showbd.f	showmt.f	showzn.f	source.f	texdraw.c
texsymchar.c	vector.f	verstr.f	vmntr.f	wrkary.f	xdraw.c
xsdummy.f	xsetattr.c	xsymchar.c	xyplot.f		

- Include files of CGVIEW

_FLAGS	_KPIDS	_KPSYMS	_MVGEOM	_MVIEW	_NGPS
_PROGV	_PTLSP	_STACK	_SYMBOL	_WKFL1	_WKGRA
_WKLAT	_WKSEL	_WKSOU	_XBANK	_XWORK	

- Source files of MVP-ART

cnterr.f	getinp.f	gtcmdl.f	gtvval.f	inpidx.f	logoma.f
main000.f	preina.f	prstop.f			

- Source files common to MVP and MVP-ART

bmcmkd.f	broadh.f	broadl.f	broadn.f	broads.f	cardo.f
clea.f	copout.f	dclea.f	dimchk.f	dmain.f	errokc.f
errokt.f	fend.f	fil5te.f	file3.f	file5t.f	fillup.f
fistpr.f	iexpin.f	intcoh.f	intdbf.f	inteff.f	intrpl.f
intth3.f	intth4.f	intth6.f	intth7.f	intths.f	lastpr.f
libout.f	listh6.f	lisths.f	lisu3r.f	mend.f	normx.f
nxtseq.f	oute.f	pointv.f	prbtab.f	readin.f	sctolc.f
sendo.f	setcon.f	sroadh.f	sroads.f	tend.f	terpl.f
terpo.f	therm6.f	thermj.f	thinit.f	update.f	unresr.f
unrint.f	wot6.f				

- Include files common to MVP and MVP-ART

_CONST1	_EXPPRM	_MAINIO	_PARAM
---------	---------	---------	--------

- Source files of MVP-BURN

autobn.f	averg0.f	averg1.f	averg2.f	averg3.f	averg4.f
avgin0.f	burn0.f	burnc4.f	burncl.f	burncx.f	burnin.f
burnip.f	burnpr.f	burnr4.f	burnrm.f	burnrv.f	burnrx.f
burnsm.f	burntm.f	burnup.f	burnvd.f	catstr.f	ccomp.f
checkp.f	chkchn.f	chkenv.f	chkmid.f	chkwrđ.f	ckbnup.f
ckstep.f	ckstp2.f	clea.f	clrcom.f	cmpchi.f	cnterr.f
copyht.f	defcnv.f	dtlist.f	fnbate.f	fncalc.f	getden.f
getln2.f	getmic.f	getmid.f	gettmp.f	gettrg.f	getvol.f
gtvval.f	holpos.f	lnchai.f	logopr.f	main.f	memove.f
mvpbrn.f	mvpdel.f	mvpin0.f	mvpinp.f	mvprun.f	mvpsym.f
namrst.f	packet.f	pdsfil.f	pdsio.f	pdsln.f	pdsnm.f
pdsrd.f	pdsrm.f	pdssr.f	pdswt.f	prstop.f	redmic.f
redmvp.f	rwchan.f	rwcom1.f	rwcom2.f	rwcom3.f	rwmatd.f
rwrest.f	sctolc.f	setmid.f	smryin.f	souchk.f	srtag.f
srtag2.f	stepnm.f	stlrd2.f	sumry0.f	sumry1.f	uđate.f
wdbl.f	zzmmm.f				

- Include files of MVP-BURN

_AVERG	_BURNP	_CBURNC1	_CBURNC2	_CBURNC3	_IOUNIT
_MEMRY	_MVPBRN	_WORDL			

- Source files of utility programs

adtoidx.c	cctimdat.c	cdumy.f	cplin.f	cputime.c	cputime2.c
cputm.f	cputm2.f	dentak.f	envexp.f	envget.f	erf.f
error.f	flushstd.c	fread.f	fsigusr.c	ftgetenv.c	ftsystē.c
funbuf.c	getlin.f	header.f	hizuke.f	iclen.f	iclen2.f
imatch.f	int8.f	jfopen.f	jikan.f	label.f	lfmalloc.c
match.f	msort.f	mtxinv.f	mvctoi.f	mvitoc.f	noblnc.f
openf.f	para.f	ranu2.f	rwind.f	tokei.f	wallcloc.c

- Source files of NTXT2LB, NLB2TXT, GMVPLBCV and MVPFAT

bio2.f	cat80.f	chuck.c	dtk.f	fat.f	fbcheck.c
fsfreader.f	gmvpłbcv.f	incinc.c	insu.f	mcrđf.f	mvpđump.f
myfsplit.f	nlb2txt.f	ntxt2łb.f	permu.f	rstchk.f	varlist.f

- Source files for multitasking

mtask.f

This is a blank page.

国際単位系（SI）

表 1. SI 基本単位

基本量	SI 基本単位	
	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량	モル	mol
光度	カンデラ	cd

表 2. 基本単位を用いて表されるSI組立単位の例

組立量	SI 組立単位	
	名称	記号
面積	平方メートル	m ²
体積	立方メートル	m ³
速度	メートル毎秒	m/s
加速度	メートル毎秒毎秒	m/s ²
波数	毎メートル	m ⁻¹
密度, 質量密度	キログラム毎立方メートル	kg/m ³
面積密度	キログラム毎平方メートル	kg/m ²
比体積	立方メートル毎キログラム	m ³ /kg
電流密度	アンペア毎平方メートル	A/m ²
磁界の強さ	アンペア毎メートル	A/m
量濃度 ^(a) , 濃度	モル毎立方メートル	mol/m ³
質量濃度	キログラム毎立方メートル	kg/m ³
輝度	カンデラ毎平方メートル	cd/m ²
屈折率 ^(b)	(数字の) 1	1
比透磁率 ^(b)	(数字の) 1	1

(a) 量濃度 (amount concentration) は臨床化学の分野では物質濃度 (substance concentration) ともよばれる。
(b) これらは無次元量あるいは次元 1 をもつ量であるが、そのことを表す単位記号である数字の 1 は通常は表記しない。

表 3. 固有の名称と記号で表されるSI組立単位

組立量	SI 組立単位			
	名称	記号	他のSI単位による表し方	SI基本単位による表し方
平面角	ラジアン ^(b)	rad	1 ^(b)	m/m
立体角	ステラジアン ^(b)	sr ^(c)	1 ^(b)	m ² /m ²
周波数	ヘルツ ^(d)	Hz		s ⁻¹
力	ニュートン	N		m kg s ⁻²
圧力, 応力	パスカル	Pa	N/m ²	m ⁻¹ kg s ⁻²
エネルギー, 仕事, 熱量	ジュール	J	N m	m ² kg s ⁻²
仕事率, 工率, 放射束	ワット	W	J/s	m ² kg s ⁻³
電荷, 電気量	クーロン	C		s A
電位差 (電圧), 起電力	ボルト	V	W/A	m ² kg s ⁻³ A ⁻¹
静電容量	ファラド	F	C/V	m ⁻² kg ⁻¹ s ⁴ A ²
電気抵抗	オーム	Ω	V/A	m ² kg s ⁻³ A ⁻²
コンダクタンス	ジーメンズ	S	A/V	m ⁻² kg ⁻¹ s ³ A ²
磁束	ウェーバ	Wb	Vs	m ² kg s ⁻² A ⁻¹
磁束密度	テスラ	T	Wb/m ²	kg s ⁻² A ⁻¹
インダクタンス	ヘンリー	H	Wb/A	m ² kg s ⁻² A ⁻²
セルシウス温度	セルシウス度 ^(e)	°C		K
光束度	ルーメン	lm	cd sr ^(c)	cd
照射度	ルクス	lx	lm/m ²	m ⁻² cd
放射性核種の放射能 ^(f)	ベクレル ^(d)	Bq		s ⁻¹
吸収線量, 比エネルギー分与, カーマ	グレイ	Gy	J/kg	m ² s ⁻²
線量当量, 周辺線量当量, 方向性線量当量, 個人線量当量	シーベルト ^(g)	Sv	J/kg	m ² s ⁻²
酸素活性化	カタール	kat		s ⁻¹ mol

(a) SI接頭語は固有の名称と記号を持つ組立単位と組み合わせても使用できる。しかし接頭語を付した単位はもはやコヒーレントではない。
(b) ラジアンとステラジアンは数字の 1 に対する単位の特別な名称で、量についての情報をつたえるために使われる。実際には、使用する時には記号rad及びsrが用いられるが、習慣として組立単位としての記号である数字の 1 は明示されない。
(c) 測光学ではステラジアンという名称と記号srを単位の表し方の中に、そのまま維持している。
(d) ヘルツは周期現象についてののみ、ベクレルは放射性核種の統計的過程についてののみ使用される。
(e) セルシウス度はケルビンの特別な名称で、セルシウス温度を表すために使用される。セルシウス度とケルビンの単位の大きさは同一である。したがって、温度差や温度間隔を表す数値はどちらの単位で表しても同じである。
(f) 放射性核種の放射能 (activity referred to a radionuclide) は、しばしば誤った用語で"radioactivity"と記される。
(g) 単位シーベルト (PV, 2002, 70, 205) についてはCIPM勧告2 (CI-2002) を参照。

表 4. 単位の中に固有の名称と記号を含むSI組立単位の例

組立量	SI 組立単位		
	名称	記号	SI 基本単位による表し方
粘着力のモーメント	パスカル秒	Pa s	m ⁻¹ kg s ⁻¹
表面張力	ニュートンメートル	N m	m ² kg s ⁻²
角速度	ニュートン毎メートル	N/m	kg s ⁻²
角加速度	ラジアン毎秒	rad/s	m m ⁻¹ s ⁻¹ =s ⁻¹
熱流密度, 放射照度	ラジアン毎秒毎秒	rad/s ²	m m ⁻¹ s ⁻² =s ⁻²
熱容量, エントロピー	ワット毎平方メートル	W/m ²	kg s ⁻³
比熱容量, 比エントロピー	ジュール毎ケルビン	J/K	m ² kg s ⁻² K ⁻¹
比エネルギー	ジュール毎キログラム毎ケルビン	J/(kg K)	m ² s ⁻² K ⁻¹
熱伝導率	ジュール毎キログラム	J/kg	m ² s ⁻²
体積エネルギー	ワット毎メートル毎ケルビン	W/(m K)	m kg s ⁻³ K ⁻¹
電界の強さ	ジュール毎立方メートル	J/m ³	m ⁻¹ kg s ⁻²
電荷密度	ジュール毎立方メートル	V/m	m kg s ⁻³ A ⁻¹
表面電荷密度	クーロン毎立方メートル	C/m ³	m ⁻³ s A
電束密度, 電気変位	クーロン毎平方メートル	C/m ²	m ⁻² s A
誘電率	クーロン毎平方メートル	C/m ²	m ² s A
透磁率	ファラド毎メートル	F/m	m ³ kg ⁻¹ s ⁴ A ²
モルエネルギー	ヘンリー毎メートル	H/m	m kg s ⁻² A ⁻²
モルエントロピー, モル熱容量	ジュール毎モル	J/mol	m ² kg s ⁻² mol ⁻¹
照射線量 (X線及びγ線)	ジュール毎モル毎ケルビン	J/(mol K)	m ² kg s ⁻² K ⁻¹ mol ⁻¹
吸収線量率	クーロン毎キログラム	C/kg	kg ⁻¹ s A
放射線強度	グレイ毎秒	Gy/s	m ² s ⁻³
放射輝度	ワット毎ステラジアン	W/sr	m ⁴ m ⁻² kg s ⁻³ =m ² kg s ⁻³
酵素活性濃度	ワット毎平方メートル毎ステラジアン	W/(m ² sr)	m ² m ⁻² kg s ⁻³ =kg s ⁻³
	カタール毎立方メートル	kat/m ³	m ⁻³ s ⁻¹ mol

表 5. SI 接頭語

乗数	名称	記号	乗数	名称	記号
10 ²⁴	ヨタ	Y	10 ⁻¹	デシ	d
10 ²¹	ゼタ	Z	10 ⁻²	センチ	c
10 ¹⁸	エクサ	E	10 ⁻³	ミリ	m
10 ¹⁵	ペタ	P	10 ⁻⁶	マイクロ	μ
10 ¹²	テラ	T	10 ⁻⁹	ナノ	n
10 ⁹	ギガ	G	10 ⁻¹²	ピコ	p
10 ⁶	メガ	M	10 ⁻¹⁵	フェムト	f
10 ³	キロ	k	10 ⁻¹⁸	アト	a
10 ²	ヘクト	h	10 ⁻²¹	ゼプト	z
10 ¹	デカ	da	10 ⁻²⁴	ヨクト	y

表 6. SIに属さないが、SIと併用される単位

名称	記号	SI 単位による値
分	min	1 min=60 s
時	h	1 h =60 min=3600 s
日	d	1 d=24 h=86 400 s
度	°	1°=(π/180) rad
分	′	1′=(1/60)°=(π/10 800) rad
秒	″	1″=(1/60)′=(π/648 000) rad
ヘクタール	ha	1 ha=1 hm ² =10 ⁴ m ²
リットル	L, l	1 L=1 l=1 dm ³ =10 ³ cm ³ =10 ⁻³ m ³
トン	t	1 t=10 ³ kg

表 7. SIに属さないが、SIと併用される単位で、SI単位で表される数値が実験的に得られるもの

名称	記号	SI 単位で表される数値
電子ボルト	eV	1 eV=1.602 176 53(14)×10 ⁻¹⁹ J
ダルトン	Da	1 Da=1.660 538 86(28)×10 ⁻²⁷ kg
統一原子質量単位	u	1 u=1 Da
天文単位	ua	1 ua=1.495 978 706 91(6)×10 ¹¹ m

表 8. SIに属さないが、SIと併用されるその他の単位

名称	記号	SI 単位で表される数値
バール	bar	1 bar=0.1 MPa=100 kPa=10 ⁵ Pa
水銀柱ミリメートル	mmHg	1 mmHg=133.322 Pa
オングストローム	Å	1 Å=0.1 nm=100 pm=10 ⁻¹⁰ m
海里	M	1 M=1852 m
バイン	b	1 b=100 fm ² =(10 ¹² cm) ² =10 ⁻²⁸ m ²
ノット	kn	1 kn=(1852/3600) m/s
ネーパ	Np	SI単位との数値的な関係は、 対数量の定義に依存。
ベレル	B	
デシベル	dB	

表 9. 固有の名称をもつCGS組立単位

名称	記号	SI 単位で表される数値
エルグ	erg	1 erg=10 ⁻⁷ J
ダイン	dyn	1 dyn=10 ⁻⁵ N
ポアズ	P	1 P=1 dyn s cm ⁻² =0.1 Pa s
ストークス	St	1 St=1 cm ² s ⁻¹ =10 ⁻⁴ m ² s ⁻¹
スチルブ	sb	1 sb=1 cd cm ⁻² =10 ⁴ cd m ⁻²
フオット	ph	1 ph=1 cd sr cm ⁻² =10 ⁴ lx
ガリ	Gal	1 Gal=1 cm s ⁻² =10 ⁻² ms ⁻²
マクスウェル	Mx	1 Mx=1 G cm ² =10 ⁻⁸ Wb
ガウス	G	1 G=1 Mx cm ⁻² =10 ⁻⁴ T
エルステッド ^(a)	Oe	1 Oe Δ (10 ³ /4 π) A m ⁻¹

(a) 3 元系のCGS単位系とSIでは直接比較できないため、等号「 Δ 」は対応関係を示すものである。

表 10. SIに属さないその他の単位の例

名称	記号	SI 単位で表される数値
キュリー	Ci	1 Ci=3.7×10 ¹⁰ Bq
レントゲン	R	1 R = 2.58×10 ⁻⁴ C/kg
ラド	rad	1 rad=1 cGy=10 ⁻² Gy
レム	rem	1 rem=1 cSv=10 ⁻² Sv
ガンマ	γ	1 γ=1 nT=10 ⁻⁹ T
フェルミ	f	1 フェルミ=1 fm=10 ⁻¹⁵ m
メートル系カラット		1 メートル系カラット=0.2 g = 2×10 ⁻⁴ kg
トル	Torr	1 Torr = (101 325/760) Pa
標準大気圧	atm	1 atm = 101 325 Pa
カロリ	cal	1 cal=4.1858 J (「15℃」カロリ), 4.1868 J (「IT」カロリ), 4.184 J (「熱化学」カロリ)
ミクロン	μ	1 μ =1 μm=10 ⁻⁶ m

