PHITS

Ver. 2.88

User's Manual

目 次

1 はじめに **1**
1.1 最近の改良点 1
1.2 開発者 11
1.3 PHITS の参考文献 12

2 インストール、コンパイル及び使用方法 **13**
2.1 動作環境 13
2.2 Windows でのインストール及び実行方法 13
2.3 Mac でのインストール及び実行方法 14
2.4 Makefile を利用したコンパイル 17
2.5 Microsoft Visual Studio と Intel Fortran を利用したコンパイル (Windows 用) 18
2.6 ANGEL のコンパイル 18
2.7 実行シェル 18
2.8 実行の途中中断 19
2.9 配列の大きさの変更 20

3 入力ファイルの書式 **21**
3.1 セクションの種類 21
3.2 読み込みコントロール 22
3.3 ファイルの挿入 23
3.4 ユーザー定義定数 24
3.5 数式の利用 24
3.6 粒子の表式 25

4 セクション書式 **27**
4.1 [Title] セクション 27
4.2 [Parameters] セクション 28
4.2.1 計算モード 28
4.2.2 ヒストリー数、バンク配列の大きさ 29
4.2.3 計算打切エネルギー、切り替えエネルギー 31
4.2.4 時間カット、ウエイトカット、ウエイトウインドウ 34
4.2.5 計算モデルオプション (1) 35
4.2.6 計算モデルオプション (2) 36
4.2.7 計算モデルオプション (3) 37
4.2.8 計算モデルオプション (4) 38
4.2.9 計算モデルオプション (5) 39
4.2.10 出力オプション (1) 40
4.2.11 出力オプション (2) 43
4.2.12 出力オプション (3) 44
4.2.13 出力オプション (4) 45
4.2.14 出力オプション (5) 46
4.2.15 幾何形状のエラー関係 47
4.2.16 入出力ファイル名 48
4.2.17 その他 49
4.2.18 低エネルギー中性子の物理パラメータ 49

4.2.19 光子・電子輸送オリジナルモデルの物理パラメータ 50
4.2.20 EGS5 用パラメータ 51
4.2.21 Dumpall オプション 54
4.2.22 Event Generator Mode 59
4.3 [Source] セクション 61
4.3.1 <Source> : マルチソース 62
4.3.2 共通パラメータ 63
4.3.3 円柱分布ソース 65
4.3.4 角柱分布ソース 65
4.3.5 ガウス分布ソース (x, y, z 独立) 66
4.3.6 一般パラボラ分布ソース (x, y, z 独立) 66
4.3.7 ガウス分布ソース (xy 平面) 67
4.3.8 一般パラボラ分布ソース (xy 平面) 67
4.3.9 球及び球殻分布ソース 68
4.3.10 s-type = 11 ソース 68
4.3.11 s-type = 12 ソース 69
4.3.12 円錐形状分布ソース 70
4.3.13 三角柱形状分布ソース 71
4.3.14 dump データソース 72
4.3.15 ユーザー定義ソース 75
4.3.16 エネルギー分布の定義 78
4.3.17 角分布の定義 86
4.3.18 時間分布の定義 88
4.3.19 マルチソースの例題 91
4.3.20 ダクトソースオプション 95
4.4 [Material] セクション 98
4.4.1 書式 98
4.4.2 核種の定義 99
4.4.3 密度の定義 99
4.4.4 物質パラメータ 99
4.4.5 $S(\alpha, \beta)$ の指定 100
4.4.6 例題 100
4.5 [Surface] セクション 102
4.5.1 書式 102
4.5.2 マクロボディー 104
4.5.3 マクロボディーの面定義 105
4.6 [Cell] セクション 106
4.6.1 書式 106
4.6.2 セルの記述方法 107
4.6.3 Universe 構造 110
4.6.4 Lattice 構造 113
4.6.5 繰り返し幾何形状 116
4.7 [Transform] セクション 123
4.7.1 書式 123
4.7.2 座標変換の定義 123
4.8 [Temperature] セクション 126

4.9 [Mat Time Change] セクション 127
4.10 [Magnetic Field] セクション 128
4.10.1 荷電粒子 128
4.10.2 中性子 129
4.11 [Electro Magnetic Field] セクション 131
4.12 [Delta Ray] セクション 132
4.13 [Super Mirror] セクション 133
4.14 [Elastic Option] セクション 134
4.15 [Data Max] セクション 135
4.16 [Frag Data] セクション 136
4.17 [Importance] セクション 139
4.18 [Weight Window] セクション 141
4.19 [Forced Collisions] セクション 142
4.20 [Brems Bias] セクション 143
4.21 [Photon Weight] セクション 144
4.22 [Volume] セクション 145
4.23 [Multiplier] セクション 146
4.24 [Mat Name Color] セクション 148
4.25 [Reg Name] セクション 150
4.26 [Counter] セクション 151
4.27 [Timer] セクション 152

5 タリー共通パラメータの書式 153
5.1 形状メッシュ 153
5.1.1 領域メッシュ 153
5.1.2 階層構造の領域と体積の定義 154
5.1.3 r-z メッシュ 156
5.1.4 xyz メッシュ 157
5.2 エネルギーメッシュ 157
5.3 LET メッシュ 157
5.4 時間メッシュ 158
5.5 角度メッシュ 158
5.6 メッシュ定義文 158
5.6.1 メッシュタイプ 158
5.6.2 e-type = 1 の場合 158
5.6.3 e-type = 2, 3 の場合 159
5.6.4 e-type = 4 の場合 159
5.6.5 e-type = 5 の場合 160
5.7 他のタリー定義文 160
5.7.1 粒子定義文 160
5.7.2 axis 定義文 161
5.7.3 file 定義文 162
5.7.4 resfile 定義文 162
5.7.5 unit 定義文 162
5.7.6 factor 定義文 162
5.7.7 output 定義文 163

5.7.8 info 定義文 163
5.7.9 title 定義文 163
5.7.10 ANGEL パラメータ定義文 163
5.7.11 2d-type 定義文 164
5.7.12 gshow 定義文 164
5.7.13 rshow 定義文 165
5.7.14 x-txt, y-txt, z-txt 定義文 166
5.7.15 volmat 定義文 166
5.7.16 epsout 定義文 166
5.7.17 カウンター定義文 166
5.7.18 resol 分解能、width 線太さ定義文 166
5.7.19 trcl 座標変換 167
5.7.20 dump 定義文 167
5.8 複数のタリー結果の統合機能 169

6 タリー入力書式 171
6.1 [T-Track] セクション 171
6.2 [T-Cross] セクション 176
6.3 [T-Point] セクション 182
6.4 [T-Heat] セクション 186
6.5 [T-Deposit] セクション 189
6.6 [T-Deposit2] セクション 193
6.7 [T-Yield] セクション 195
6.8 [T-Product] セクション 198
6.9 [T-DPA] セクション 202
6.10 [T-LET] セクション 205
6.11 [T-SED] セクション 208
6.12 [T-Time] セクション 211
6.13 [T-Star] セクション 214
6.14 [T-Dchain] セクション 217
6.15 [T-WWG] セクション 221
6.16 [T-Userdefined] セクション 224
6.17 [T-Gshow] セクション 229
6.18 [T-Rshow] セクション 231
6.19 [T-3Dshow] セクション 233
6.19.1 box の定義 236
6.19.2 3dshow の例題 237

7 タリーを用いた体積、面積計算 240

8 dump ファイルの処理 242

9 出力中性子、光子データフォーマット 247

10 領域エラーチェック 249

11 並列版のための指定方法 **251**
11.1 メモリ分散型並列 251
11.1.1 実行方法 251
11.1.2 maxcas, maxbch の指定方法 251
11.1.3 異常終了の処理 252
11.1.4 PHITS での ncut, gcut, pcut, dumpall ファイルの指定 252
11.1.5 PHITS での読み込みファイルの指定 252
11.2 メモリ共有型並列 253
11.2.1 実行方法 253
11.2.2 メモリ共有型並列計算の注意点 253

12 FAQ **255**
12.1 パラメータ設定関連 255
12.2 エラー、コンパイル関連 256
12.3 タリー関連 257
12.4 線源設定関連 258

索引 **259**

1　はじめに

PHITS コードは、日本原子力研究所 (以下、「原研」と記す。現在の組織名は、日本原子力研究開発機構であり、以下、「原子力機構」と記す。) が開発した高エネルギー核反応モデル組込み核子中間子輸送コード NMTC/JAM ver.2 に、高度情報科学技術研究機構 (RIST)、東北大、原研／原子力機構の協力のもとに、重イオンの輸送計算機能を組み込んだ粒子、重イオン輸送統合コードシステム (Particle and Heavy Ion Transport code System; *PHITS*) です。

NMTC/JAM ver.2 では、低エネルギーの中性子、光子、電子の輸送を含むことにより、高エネルギーから低エネルギーまでの輸送を NMTC/JAM コードだけで計算することが可能です。低エネルギーの輸送部分に ACE 形式の断面積データを用いた場合、MCNP4C と同じ結果を与えます。従って、今までの NMTC/JAM と MCNP4C のつなぎ計算の結果も同じ結果を与え、しかも、タリーの結果は両者を統合したものが得られます。

これらの NMTC/JAM の成果に、重イオン核反応と物質中の重イオン輸送の機能を加え、MCNPX を超える機能を持つ粒子、重イオン輸送統合コードに発展しました。

このマニュアルは、JAERI-Data/Code 2001-007 のマニュアルの日本語訳抜粋を含んでいます。また、上記の英語のマニュアル完成以後のコードの改良に伴う変更も記述してあります。例えば、GG 幾何形状の導入、並列化、DPA タリーの導入、低エネルギーの中性子、光子、電子の輸送などです。しかしながら、コードの概要、物理モデルなどの解説は、コンパクトにするために省いています。

今後も、新しい機能が入り次第改訂していきますので、よろしくお願いいたします。皆さんに使ってもらって、バグ潰しと改良を加えたいと思います。できるだけ多くの利用をお願いいたします。また、バグ、不都合の報告、改良の希望など、お待ちしています。

1.1　最近の改良点

以下に、*PHITS* バージョン 2.24 以降の主な改良点を記載します。

バージョン 2.88 では主に次の改良を行いました。

- `[t-dchain]` でも sumtally 機能が使えるようになりました。
- ユーザー指定断面積の読み込み機能を拡張し、データが与えられていないエネルギー・角度領域に外挿を行って補間するオプションと微分断面積を与えない場合に核反応モデルにより核反応イベントを計算するオプションを追加しました。
- 計算打切エネルギーにより輸送が止められた中性子が全て崩壊してしまうバグを修正しました。このバグはバージョン 2.83 以降で発生していましたのでご注意ください。

バージョン 2.87 では主に次の改良を行いました。

- `[t-3dshow]` で xyz 軸を出力可能としました。
- 連続四面体形状の読込方法を改良し，計算を高速化しました。
- ジオメトリエラー出力ファイル名を “*.err” から “*_geo.out” に変更しました。
- 物質の定義で MTx（$S(\alpha,\beta)$ テーブル）を使用した際のバグ（バージョン 2.86 のみで発生）を修正しました。

バージョン 2.86 では主に次の改良を行いました。

- 新しいタリー `[t-wwg]` を導入しました。このタリーを使えば，Weight Window 機能が効果的に動作するパラメータ設定を自動で得ることができます。詳しくは、6.15 節をご参照下さい。
- 3 次元可視化ソフトウェア ParaView 用の出力機能を作成しました。この機能を使えば，PHITS の計算結果を ParaView 上に 3 次元的に可視化することが可能です。使い方は，utility フォルダにある ParaView をご参照下さい。また、2 次元のタリー計算結果やジオメトリを Bitmap 形式で出力する機能を加えました。これらの改良は，原子力機構原子力センシング研究グループの古高和禎さんおよび（株）V.I.C. の協力により実施いたしました。
- 荷電粒子の阻止能を全て ATIMA で計算するモード（`ndedx = 3`) を新たに加えデフォルト設定としました。
- 現在のバッチ情報を出力するファイルの名前を `batch.now` から `batch.out` に変更し、[parameters] セクションにおいて `file(22)` により指定できるようにしました。
- RI 線源発生機能を追加しました。この機能を使えば，放射性核種とその放射能を指定することにより，放射性核種の崩壊に伴って放出される粒子の線スペクトルを自動的に定義することができます。ただし、崩壊ガンマ線のみ扱えます。崩壊ガンマ線の線スペクトル計算には，放射性核種崩壊データベース DECDC [1] を使用します。このデータベースは ICRP107 と同等です。詳しくは表 55 をご参照ください。なお，本改良は，原子力機構原子力基礎工学研究センターの遠藤章氏にご協力いただきました。
- [material] セクションで質量数を定義せずに元素を指定した際，天然同位体元素比に従って自動で各同位体に展開する機能を作成しました。ただし，JENDL-4.0 に含まれない核種に対しては展開されませんのでご注意下さい。本改良は、原子力機構・システム計算科学センターにご協力いただきました。
- マテリアルの核種ごとに核データライブラリー利用の上限エネルギーを設定できる [data max] セクションを追加しました。詳しくは、4.15 節をご参照下さい。本改良は、原子力機構・システム計算科学センターにご協力いただきました。
- ミューオン核反応モデルを改良しました。詳しくは，文献[2] をご参照下さい。
- 重陽子の全反応断面積を精度良く記述する新規の模型を導入しました。`[parameters]` セクションにおいて `icrdm=1` とすることで利用できます。詳細は文献[3] をご覧ください。
- パイオンの全反応断面積を改訂し、それをデフォルトで使用するようにしました。従来の幾何学的な断面積と比べて、入射エネルギー依存性を考慮しており、デルタ共鳴によるピーク構造を再現できるようになりました。従来の幾何学的断面積を使用する場合は、[parameters] セクションにおいて、`icxspi=0` としてください。
- 1 つのインプットファイルで複数の sumtally サブセクションを定義できるようにしました。また，sumtally に関するいくつかのバグを修正しました。

バージョン 2.85 では主に次の改良を行いました。

- 3GeV/u 以上の重イオン核反応で使われる JAMQMD モデルを改良し，宇宙線輸送計算などの精度及び安定性を高めました。従来モデルでは，まれにメモリ違反で強制終了されてしまう可能性がありました。

---

[1] A. Endo, Y. Yamaguchi and K.F. Eckerman, Nuclear decay data for dosimetry calculation - Revised data of ICRP Publication 38, JAERI 1347 (2005).

[2] S. Abe and T. Sato, Implementation of muon interaction models in PHITS, J. Nucl. Sci. Technol. (2016) [http://www.tandfonline.com/doi/abs/10.1080/00223131.2016.1210043]

[3] K. Minomo, K. Washiyama, and K. Ogata, J. Nucl. Sci. Tech. DOI:10.1080/00223131.2016.1213672

- 阻止能計算モデル ATIMA のアルゴリズムを改良し、高速化しました。この改良により、精度の高い ATIMA を用いた PHITS シミュレーションが、SPAR を用いた場合とほぼ同程度の計算時間で実行できるようになりました。本改良は、原子力機構・システム計算科学センター・原子力コード高速化作業の一環として、高度情報科学技術研究機構（RIST）の和田暁雄さんに実施していただきました。
- 飛跡を計算する最小&最大エネルギーを指定するパラメータ `esmin` と `esmax` の単位を MeV から MeV/u に変更しました。
- EGS5 を使って高エネルギー光子（約 10MeV 以上）を輸送する際，ジオメトリを細かく区切ると光子の飛程が長くなってしまうバグを修正しました。
- 標的中に ^{1}H が存在する際，負ミューオンが全て ^{1}H に捕獲されてしまうバグを修正しました。

バージョン 2.84 では、幾つかの微小なバグを修正しました。

バージョン 2.83 では主に次の改良を行いました。

- 平均寿命 886.7 秒で中性子が陽子，電子，反電子ニュートリノに崩壊する反応を考慮できるようにしました。ただし，デフォルトでは時間カットオフ（1 秒）で粒子輸送がストップしてしまうため，中性子の崩壊が問題になるような巨大な体系では，中性子のみならず，その崩壊により生成される陽子や電子などの `tmax` を大きく設定して下さい。
- EGS5 を用いて付与エネルギーを計算する場合に，ドップラー効果によるエネルギーの微妙なゆらぎを光子によるエネルギー付与とカウントしてしまうバグを修正しました。このゆらぎが問題となる低エネルギー光子（主に 100keV 以下）によるエネルギー付与計算や，`[t-deposit]` で `part = photon` としていた場合に結果が変化しますので，ご注意下さい。
- `[t-point]` タリーよりも後ろに `mesh = reg` を含むタリーが書かれていると，そのタリーの結果が壊れてしまうバグを修正しました。
- `infl` を含むインプットファイルにおいて `sumtally` 機能が使えないバグを修正しました。

バージョン 2.82 では主に次の改良を行いました。

- ある点やリング状の線分上における中性子・光子フルエンスを効率的に計算するポイントタリー `[t-point]` を導入しました（6.3 節及び `\utility\tpoint` 参照）。
- `[t-yield]` タリーに，弾性散乱により反跳された標的核情報を出力する `elastic` オプションを加えました。
- `[t-star]` タリーに，標的核が核変換を起こした場合の star density を出力するオプション `output = transmut` を加えました。
- `[counter]` セクションに，核分裂のオプション `fiss` を加えました。これにより，核分裂を経由して生成する粒子、特に核分裂の世代（回数）毎の情報を選択的にタリー可能となりました。
- `[source]` セクションに，文献[4] で計算される自発核分裂からの中性子線源を模擬するオプションを加えました。この改良は，株式会社ナイスの Liem Peng Hong 氏のご協力を得て行いました（4.3.2 節参照）。
- `[source]` セクションに，三角柱内に分布する線源を表現するオプションを加えました（4.3.13 節参照）。

[4] J. M. Verbeke, C. Hagmnn, and D. Wright, “Simulation of Neutron and Gamma Ray Emission from Fission and Photofission”, UCRL-AR-228518 (2014).

- [source] セクションで，任意の時間分布を指定可能としました（4.3.18 節参照）。
- [parameters] セクションに，中性子増倍効果を制御する NONU パラメータを追加しました。この機能により，臨界解析コード等により別途評価した同体系における中性子増倍効率を加味した中性子線源を用いて，実質的な中性子透過問題の評価が行えるようになりました。
- [t-track] タリーで mesh = r-z の場合に，円面の角度 θ も定義可能としました。この機能により，扇形柱体における粒子フルエンスをタリー可能となりました。
- EGS5 に関連するいくつかのバグ（[t-track] を使ったときに電子のエネルギーが増えてしまうバグ，物質として水を指定したときにホルミウム（元素記号 Ho）の値を使って密度効果などを補正してしまうバグなど）を修正しました。
- 核共鳴蛍光散乱において入射光子の偏光効果を考慮可能としました。
- [t-dpa] を使った再開始計算を可能としました。
- Sumtally 機能を拡張し，[t-dchain] 以外の全てのタリーに対応しました。（バージョン 2.88 より、全てのタリーで本機能が使えるようになりました。）
- [t-deposit] タリーで output=deposit として part を指定した場合，各粒子の寄与を正しく計算可能としました。
- ミューオン核反応，光核反応，JQMD-2.0 に関する軽微なバグを修正しました。
- utility フォルダに，連続四面体 (TetraGEOM), ポイントタリー (tpoint)，ユーザー定義タリー (usrtally) の使用方法の解説を加えました。

バージョン 2.81 では主に次の改良を行いました。

- makefile で依存関係を正しく記述するようにしました。この改良により，make -j オプションが使えるようになり，コンパイルが速くなりました。なお，makefile を使ってコンパイルした際に作られる実行ファイル名が変更されていますのでご注意下さい。この改良は，原子力機構 原子力センシング研究グループの古高和禎さんの協力の下，実施いたしました。
- EGS5 モードを使用したときに使える物質制限をなくしました。ただし，数 100 以上の物質を定義した場合は，メモリ不足により計算が実行できない場合があります。また，EGS5 の仕様により，1 つの物質に含まれる元素数の最大値は 20 となります。
- EGS5 モードで [t-track] を使ったときに希に発生するバグを修正しました。
- EGS5 モードで [t-deposit] タリー，output = deposit のときに全吸収ピークエネルギーが多少低くなってしまうバグを修正しました。
- [t-dchain] で継続行を正しく認識するようにしましたまた，[t-dchain] で指定できる領域の最大数を 100 から 500 に拡張しました
- [t-deposit] にて mesh=reg,output=deposit で計算した際，デルタ線を生成した粒子のエネルギー付与量計算の誤りを修正しました。
- [t-heat] において mesh=r-z とした場合に起こるバグを修正しました。

バージョン 2.80 では主に次の改良を行いました。

- 連続四面体形状（ポリゴンの一種）の読み込み機能を組み込みました（4.6.5.4 節参照）本改良は，韓国 Hanyang 大学の HUREL 研究所の協力の下，実施しました。
- ミューオンによる制動放射及び対生成を再現できるようにしました。また，ミューオン捕獲反応におけるバグを修正しました。これにより，ミューオンが引き起こすほぼ全ての反応を精度よく再現可能となりました。
- 核共鳴散乱（NRF）計算機能を組み込みました。これにより，数 MeV 程度の光子照射による原子核の励起や核異性体の生成が再現できるようになりました。核共鳴蛍光散乱モデルは `[parameters]` セクションで `ipnint=2` とすることにより起動します。
- Sumtally 機能を拡充し，`[t-dpa]`，`[t-dchain]` を除く全てのタリー結果を足し合わせることが可能となりました。（バージョン 2.88 より、全てのタリーで本機能が使えるようになりました。）
- ユーザー指定断面積の読み込み機能を追加しました（4.16 節参照）。
- 荷電粒子のエネルギー分散計算方法を修正しました。この改良により，陽子や重イオンによるブラッグピーク付近の吸収線量がより精度よく再現可能となりました。
- 飛程に関するパラメータ `deltc`，`deltm` を物質の密度で割ることにより，気体中の計算時間を短縮しました。
- Dump 線源を使った繋ぎ計算時の統計誤差計算方法を改良しました（4.3.14 節参照）。
- `pnimul` パラメータを導入し，光核反応を優先して引き起こす機能を追加しました。
- `[t-yield]` の誤差計算方法のバグを修正しました
- EGS5 モードに関する改良点
  - `ipegs` パラメータを導入し，PEGS5 のみ実行したり，PEGS5 は実行せずに PHI TS のみ実行したりすることを可能としました。
  - `imsegs` パラメータを導入し，EGS5 を使った場合，物質が変わるたびに多重散乱を詳細に模擬するようにしました。これにより，`chard` パラメータを変更することなく，薄膜による電子の散乱を模擬可能としました（PHITS オリジナルオプション）。
  - EGS5 を使った場合に，電子の飛程が短くなるバグ（v2.77 のみ）を修正しました。
  - EGS5 で使用するメモリの一部を動的配列化することにより，EGS5 を使った場合の物質制限数をなくしました。本改良は，原子力機構・システム計算科学センター・原子力コード高速化作業の一環として高度情報科学技術研究機構 (RIST) の足立将晶さんに実施していただきました。

バージョン 2.77 では主に次の改良を行いました。

- EGS5 を使用した際、`axis=eng` としたときに、タリー結果が不自然な飛び飛びの値を取るという不具合を修正しました。
- ミューオン捕獲反応を記述するモデルを改良しました。
- 重陽子やアルファ粒子などの軽イオンが標的となる核反応を、逆運動学によって計算するようにしました。すなわち、これらの反応が起こった際、重イオンが入射粒子の場合でも、デフォルトの設定では INCL が核反応モデルとして使用されます。

バージョン 2.76 では主に次の改良を行いました。

- ミューオンが仮想光子を介して引き起こす核反応モデルを組み込みました。また、負ミューオンが物質内で止まったときに起こるミューオン捕獲反応を考慮できるようになり、この反応で生成されるミューオニックアトムからの特性 X 線の放出、及びその後に起こる核吸収反応も再現できるようになりました。
- `nspred = 2` として荷電粒子の角度分散を考慮する際、その分散の大きさを調整するパラメータを導入しました。
- Intel Fortran コンパイラの最新版 (2015) で生じるバグを修正しました。

バージョン 2.75 では、複数のタリーセクションが書かれたインプットファイルで sumtally 機能が動作しないバグを修正しました。また、`e-mode=2` を選択した場合に発生するバグを修正しました。

バージョン 2.74 では主に次の改良を行いました。

- PHITS パッケージに含まれる DCHAIN-SP のバージョンを DCHAIN-SP2001 (dchain264.exe) から DCHAIN-SP2014(dchain274.exe) に変更しました。DCHAIN-SP2001 と比べた DCHAIN-SP2014 の主な改良点は以下の通りです。
  (1) 入力ファイル形式の変更
  (2) 放射化計算断面積ライブラリの中性子エネルギー群数を 175 から 1968 に変更
  (3) PHITS 入力形式での [source] セクション出力オプションの追加
  (4) ANGEL 入力形式での残留放射能出力オプションの追加
- EGS5 部分もスレッド並列化しました。これにより、`negs=1` としたときにスレッド並列で実行できなくなる制限がなくなりました。また、これに関連して、EGS5 部分のバグをいくつか修正しました。本並列化の開発は、原子力機構・システム計算科学センター・原子力コード高速化作業の一環として高度情報科学技術研究機構 (RIST) の足立将晶さんに実施していただきました。
- 複数のタリー結果を足し合わせる新しい機能 “sumtally” を追加しました。（バージョン 2.88 より、全てのタリーで本機能が使えるようになりました。）詳しくは、5.8 節をご参照下さい。本機能の開発は、原子力機構・システム計算科学センター・スーパーコンピュータ利用に係るプログラミング支援作業の一環として RIST の三浦孝充さんに実施していただきました。
- Kurotama モデルで、5GeV/u 以上の断面積を出力できるようにしました。詳しくは文献[5] をご参照下さい。
- γ 脱励起に関するデータベース `trxcrd.dat` をソースファイルに組み込みました。この改良により、`e-mode≥1` や `igamma≥1` のときに `file(14)` を指定する必要がなくなりました。
- JAM 及び JAMQMD などに関連するバグをいくつか修正しました。

バージョン 2.73 では、核反応モデルによる計算結果として di-neutron 等の異常な原子核が生成されるバグを修正しました。また、Windows OS については、メモリ共有型並列計算用の実行ファイルとして 64bit 版をインストールするようにしました。シングルコアによる計算は 32bit 版と 64bit 版の両方の Windows で動作しますが、メモリ共有型並列計算は 32bit 版では動作しなくなりますのでご注意ください。

バージョン 2.72 では、`igamma=2` を選択した際に発生するバグと GEM の計算結果として di-neutron（2 つの中性子で構成される原子核）が生成されるバグを修正しました。また、[source] セクションにおいて

[5] L. Sihver *et al.*, Nucl. Instr. & Meth. B 334, 34-39 (2014).

a-type により角分布を定義する際、角度を degree 単位で与えていた場合のバグを修正しました。cos の値に変換して内挿を実行すべきところを degree 単位のまま行っており、指定したビン幅で一様な分布を取るべきところがバイアスのかかった分布となっておりましたのでご注意ください。加えて、[source] の a-type サブセクションにおいて、na や nn を負の値で与えられないようにしました。

バージョン 2.71 では、EGS5 を使用した際の電子対消滅の取り扱いに関するバグを修正しました。

バージョン 2.70 では主に次の改良を行いました。

- 電磁カスケード計算コード EGS5 [6] を組み込み、光子・電子・陽電子の輸送に使用することが可能となりました。EGS5 は [parameters] セクションで negs=1 とすることにより起動します。その際、file(20) で EGS5 用データライブラリ格納フォルダを指定する必要があります。ただし、現在のところ、EGS5 を使ってメモリ共有型並列計算を行うことはできませんのでご注意ください。また、EGS5 を使った場合、定義できる物質数が 100 に制限されます。（バージョン 2.80 よりこの制限はなくなりました。）詳しくは、4.2.20 節をご参照ください。 この開発は、KEK の平山英夫氏と波戸芳仁氏と共同で実施しました。
- 非共鳴領域の光核反応モデルを JAM に組み込むことにより、1TeV までの光核反応を再現可能としました。
- ミューオンの仮想光子放出断面積モデル（Minorikawa *et al.* Nuove Cimento 1981）と光核反応モデルを組み合わせることにより、1TeV までのミューオン核反応を再現可能としました。このモデルは、[parameters] セクションで imuint=1 とすることにより起動します。
- 核データライブラリに格納された荷電粒子放出断面積（(n,p), (n,α) など）を読み込むようにイベントジェネレータモード ver.2 を改良しました。この改良により、20MeV 以下の中性子核反応から放出される荷電粒子スペクトルが精度よく計算できるようになりました。このモードは、[parameters] セクションで e-mode=2 とすることにより起動します。
- 重イオン核反応モデル JQMD に相対論効果や初期状態安定アルゴリズムを組み込んだ JQMD-2.0 モデルを開発しました。JQMD-2.0 モデルは、[parameters] セクションで irqmd = 1 とすることにより起動します。この開発は、フランス CEA の Davide Mancusi 氏と共同で実施しました。
- [t-deposit] で output=deposit としてイベント毎の付与エネルギーを計算する際、実測を再現するように意図的にエネルギー分解能を持たせる機能を追加しました。

バージョン 2.67 では主に次の改良を行いました。

- 領域エラーチェック機能を追加しました。ジオメトリを 2 次元表示するタリーで自動的に実行され、2 重定義や未定義の領域がある場合にその領域に色を付けて出力します。詳しくは 10 節を参照してください。
- 20MeV 以下の中性子輸送にイベントジェネレータモードを適用した際、二次中性子が 2 個以上放出される反応で二次中性子のエネルギー・角度分布が核データからずれるのを阻止する拡張機能を導入しました。e-mode=2 で使用できます。この改良により 20MeV 以下の中性子に対するイベント解析の精度が向上しました。詳しくは 4.2.22 節をご参照ください。
- file(6)(D=phits.out) に出力される PHITS のサマリー情報を指定するパラメータ infout を用意しました。これにより必要とする情報のみを出力させることができるようになります。

[6] H. Hirayama *et al.*, SLAC-R-730 (2005) and KEK Report 2005-8 (2005).

- コンソール画面に現在計算中のバッチ番号を出力するようにしました。また、断面積ファイルが見つからないなど、PHITS の実行を停止してしまうような重要なエラーメッセージをコンソール画面にも出力するようにしました。
- `s-type=18,19` とすることで、円錐形状の線源領域を指定できるようになりました。
- `dumpall` オプションと `[t-cross]`, `[t-time]`, `[t-product]` における `dump` 機能が再開始計算を実行した場合でも利用できるようになりました。各タリーで `dump` 機能を使用した際に出力していた.cfg の内容を `file=`で指定するファイルに出力し、dump data 自体は "_dmp" が付いたファイルに書き出すことにしました。
- `param.inc` の中で指定している *PHITS* が使えるメモリの最大値 (`mdas`) を 120,000,000（=1GB 相当）に上げ、lattice 格子数の最大値 latmax を 25,000,000 に上げました。この変更により ICRP ファントムなど詳細な人体ボクセルファントムが再コンパイルなしで扱えるようになりました。

バージョン 2.66 では主に次の改良を行いました。

- DWBA（歪曲波ボルン近似）計算で求めた離散スペクトルを考慮する機能を追加しました。特定の標的原子核における陽子、重陽子入射反応を対象として、放出中性子や陽子のエネルギースペクトルに、DWBA によって計算した離散スペクトルを加えます。
- 光子入射によるパイオン生成反応として Δ 共鳴と N^* 共鳴過程を組み込むことにより、1GeV までの光核反応を計算できるようにしました。（バージョン 2.70 より 1TeV まで計算できるようになりました。）

- Gy 単位によるタリー出力オプションを `[t-heat]` タリーにも拡張しました。また、質量密度が 0 の場合に結果が `NaN` となってしまうバグを修正しました。
- [source] セクションにおいて、`e-type = 2,3,5,6,7,12,15,16` を設定して関数系でエネルギー分布を与える場合に、その分割数 `nm` が負の場合に発生するバグを修正しました。同様に、`a-type = 5,6,15,16` で角度分布を関数系で与える場合に、分割数 `nn` を負とした際のバグを修正しました。

バージョン 2.65 では、`[t-deposit]` タリーを改良し、Gy の単位で dose 結果を出力できるようにしました。また、[material] と [cell] セクションにおいて、密度を質量密度で定義していた場合に発生するバグを修正しました。原子数密度への変換を行う際に間違いがあり、中性子が過剰な元素を扱う場合にその影響が出ていました。最大で 0.6%程度の影響がありました。

バージョン 2.64 では、光核反応や EBITEM、その他の幾つかについてバグ修正を行いました。また、`[t-heat]` を使って以下の核種に対する中性子からの付与エネルギーを計算した場合に、結果が `NaN` となってしまうバグがありましたので、それらを修正しています。カーマ近似を使わない場合の結果に影響はありません。

```
As075 Ba130 Ba132 Ba134 Ba135 Ba136 Ba137 Ba140 Br079 Br081 Cd106 Cd108
Cd110 Cd111 Cd112 Cd113 Cd114 Cd116 Ce141 Ce142 Ce143 Ce144 Cf250 Fe059
Ga069 Ga071 Hf174 Hf176 Hf177 Hf178 Hf179 Hf180 Hf181 Hf182 I_127 I_129
I_130 I_131 I_135 In113 In115 Kr078 Kr080 Kr082 Kr083 Kr084 Kr085 La138
La139 La140 Mo092 Mo094 Mo095 Mo096 Mo097 Mo098 Mo099 Mo100 Nb094 Nb095
Ni059 Pr141 Pr143 Rb085 Rb086 Rb087 Rh103 Rh105 Ru096 Ru098 Ru099 Ru100
Ru101 Ru102 Ru103 Ru104 Ru105 Ru106 Sb121 Sb123 Sb124 Sb125 Sb126 Se074
Se076 Se077 Se078 Se079 Se080 Se082 Sr084 Sr086 Sr087 Sr088 Sr089 Sr090
Tc099 Te120 Te122 Te123 Te124 Te125 Te126 Te127m Te128 Te129m Te130 Te132
Xe124 Xe126 Xe128 Xe129 Xe130 Xe131 Xe132 Xe133 Xe134 Xe135 Y_089 Y_090
Y_091 Yb168 Yb170 Yb171 Yb172 Yb173 Yb174 Yb176 Zr093 Zr095
```

バージョン 2.60 では主に次の改良を行いました。

- 核反応後に生成する残留核の脱励起を ENSDF (Evaluated Nuclear Structure Data File) データベースに基づいて計算するモデル (EBITEM: ENSDF-Based Isomeric Transition and isomEr production Model) を導入しました。この改良により、不連続なピークを持つ即発 γ 線のエネルギースペクトルを精度よく再現できるようになりました。また、このモデルを用いることで準安定核（アイソマー）の生成率を計算できるようになりました（詳しくは 4.2.5 節を参照してください）。

- 光子入射による準重陽子崩壊過程を組み込むことにより、140MeV までの光核反応を計算できるようにしました。（バージョン 2.70 より 1TeV まで計算できるようになりました。）また、^{6}Li, ^{12}C, ^{14}N, ^{16}O に対して、巨大共鳴が起きたときの蒸発過程にアイソスピン依存性を考慮するようにしました。この改良により、これらの核種から α 粒子の放出が抑えられ、より多くの中性子や陽子が生成されるようになりました。

- 電磁混合場における放射線挙動（電子除く）を模擬できるようにしました。詳しくは 4.11 節を参照してください。

- `[source]` セクションにおいてエネルギー分布を定義する際、エネルギー微分値、すなわち単位が [/MeV] で与えられた線源スペクトルをそのまま利用できるようになりました。また、エネルギー分布としてポイントワイズ、すなわち各点各点で与えられる不連続なものも定義できるようになりました。詳細は 4.3.16 節をご覧ください。

- いくつかのアルゴリズムを最適化して計算時間を短縮しました。特にメッシュ数の多いタリー計算で計算時間が短くなりました。また、タリー及び $A_N G_E L$ におけるメモリ使用方法を改善しました。本改良は、原子力機構・システム計算科学センター・原子力コード高速化作業の一環として (株) 富士通システムズ・イーストの大日向大地さんに実施していただきました。

- 下記に示すバグ修正及び軽微な改良を行いました。
  - PHITS と INCL4.6 の参考文献を修正
  - `[t-dchain]` でタリー領域数の制限をなくし、Lattice に対応
  - JENDL-4.0 の更新に併せていくつかの核データを更新
  - 電子の最大エネルギー制限を 1GeV から 10GeV に変更
  - JAM、SMM でまれに発生するイベントによるバグを修正
  - δ 線生成に関するバグを修正
  - 電子の Lost Particle が起きたときのバグを修正
  - 異常終了したときの結果からも再開始計算を可能とした
  - 物質の密度を乗じる新たな multiplier function を追加
  - 7 桁のセル番号を使えるようにした
  - `istdev = 2` で `xyz-mesh` タリーを使用した場合に計算時間が膨大になることを回避
  - 陽子の断面積ライブラリ読み込み時のバグを修正

- また、スレッド並列化未対応のため、CG ジオメトリに対するサポートを終了しました。今後はマクロボディを活用して GG でジオメトリを作成してください。なお、CG ジオメトリで書いた PHITS のインプットファイルは、スレッド並列を使わない限り従来通り動作します。

バージョン 2.52 では主に次の改良を行いました。

- 電子、陽電子、および光子の輸送について、新規のアルゴリズムを導入しました。電子と陽電子の阻止能をそれらの計算打切エネルギーに応じて変化させ、高エネルギー電子の挙動が計算打切エネルギーに依存しないようにしました。加えて、光子や電子の輸送計算でもイベントジェネレータとなるように改良しました。
- また、DCHAIN-SP 用のインプットファイルを作成することができる `[t-dchain]` タリーを新しく実装しました。DCHAIN-SP は放射線による物質の放射化の時間変化を調べることができるコードで、このタリーを用いることで *PHITs* と DCHAIN-SP の接続が容易になります。詳細は 6.14 節をご覧ください。
- 新たなマクロボディーとして、楕円柱 REC (Right Ellliptical Cylinder)、カットされた円錐形 TRC (Truncated Right-angle Cone)、楕円体 ELL (Ellipsoid)、くさび形 WED (Wedge) を追加しました。

バージョン 2.50 では次の様々な機能を追加しました。

- タリー出力の統計誤差が正しく計算できるようになりました。また、統計が十分でなかった場合などに対応するため、古い計算結果から次の初期乱数とタリー出力を読み込んで再開始計算を行う機能を実装しました。詳細は 4.2.2 節をご覧ください。本改良は、原子力機構・システム計算科学センター・原子力コード高速化作業の一環として (株) 富士通システムズ・イーストの大日向大地さんに実施していただきました。
- メモリ共有型並列計算に対応させるため、ソースを大幅に書き換えました。ただし、メモリ共有型並列には、まだいくつかの制限がありますのでご注意下さい (11.2 節参照)。また、それに伴い、古いコンパイラー (f77, g77 など) ではコンパイルできなくなりました。詳しくは 2.4 節を参照してください。本研究の成果は次世代生命体統合シミュレーションソフトウェアの研究開発プロジェクト、理化学研究所戦略的研究展開事業、理化学研究所基礎科学特別研究員制度の支援によって得られたものです。また、開発において「京」コンピュータ試験利用および理研情報基盤センター RICC システムを利用させていただました。
- 巨大共鳴反応断面積を評価済み核データである光子入射反応データ (JENDL-PD/2004) を使用するように変更しました。ただし、光核反応の中で現在の *PHITs* が取り扱うことのできる反応は巨大共鳴反応だけです。入射光子エネルギーが巨大共鳴領域より大きい場合（約 20MeV 以上）、*PHITs* の計算は過小評価します。ご注意ください。
- 蒸発モデルの GEM を拡張し、統計マルチフラグメンテーションモデル (SMM) を加えました。これにより、元の核から大幅に (60-90 %程度) 質量が減るような核の生成を正確に評価できるようになりました。
- 核子、パイオン、軽イオン入射反応を精密に記述する INCL(Intra-Nuclear Cascade of Liège) 模型を核反応模型として組み込みました。本バージョン以降、これらの粒子が関与する反応では初期設定で INCL が選択されます。本模型の PHITS への組込は、(独) 日本原子力研究開発機構と CEA/Saclay の共同研究の一環として、Liège 大学の Joseph Cugnon 氏、CEA/Saclay の Davide Mancusi 氏、Alain Boudard 氏、Jean-Christophe David 氏、Sylvie Leray 氏らの協力のもと行いました。
- 最新の反応断面積模型である KUROTAMA 模型を組み込みました。天然に存在する安定核はもちろん不安定核種が関与する反応についても、非常に幅広いエネルギー領域に対して適用可能となっています。この組み込みは、(独) 理化学研究所の小濱洋央氏、(大) 高知大学の飯田圭氏、(学) 愛知淑徳学園の親松和浩氏らとの共同開発によるものです。
- 核子入射反応において軽イオン生成過程を取り入れた INC-ELF(Intra-Nuclear Cascade with Emission of Light Fragment) 模型を核反応模型として組み込みました。この組み込みは、(独) 日本原子力研究開発機構と (大) 九州大学の間の共同研究の一環として魚住研究グループに実施していただきました。

- ユーザーが任意の物理量を導出できるよう設定できるユーザー定義タリー `[t-userdefined]` を加えました。これにより、既存のタリーでは難しかったシミュレーション結果に対する詳細な分析が可能となります。ただし、ソースファイルのコンパイルが必要となります。詳しい使用方法は 6.16 節をご覧ください。
- ^{35}Cl など幾つかの核種に関して中性子の Kerma factor を更新しました。また、光子-原子および電子-原子データライブラリーをそれぞれ JENDL-4.0 とリバモア評価済電子データライブラリー (EEDL) を基にして新たに開発し、利用できるようにしました。

バージョン 2.30 では、材料損傷の指標である “原子あたりのはじき出し数 (Displacement Per Atom, DPA)” 導出の計算モデルにおいて、輸送荷電粒子のクーロン弾性散乱の寄与を含むように拡張しました。これにより従来よりも DPA の再現性が向上しました。また、[multiplier] セクションを追加し、任意のエネルギー依存の係数を [t-track] タリーの結果に掛けることが可能となりました。

バージョン 2.28 では、`dumpall` オプションと `[t-cross]`, `[t-time]`, `[t-product]` における dump 機能が MPI による並列計算でも利用できるようになりました。使用する並列 PE (Processor Element) 数 −1 個のファイルを作成し、PE 毎にファイルを変えて各結果を書き出します。読み込みも同様で、各 PE に対応したファイルの中身をそれぞれが読み込みます。

バージョン 2.26 では、荷電粒子が物質中を通過する際に発生する δ 線を 2 次粒子として実際に輸送させることができるようになりました。[delta ray] セクションを利用して領域毎にしきい値エネルギーを指定することにより、そのエネルギー以上の δ 線を発生させます。

1.2 開発者

PHITS は以下のメンバーにより開発を進めています。

(財) 高度情報科学技術研究機構 (RIST)
仁井田浩二

(国研) 日本原子力研究開発機構 (JAEA)
佐藤達彦、岩元洋介、橋本慎太郎、小川達彦、古田琢哉、安部晋一郎、甲斐健師、松田規宏、中島宏、深堀智生、奥村啓介、甲斐哲也

(共) 高エネルギー加速器研究機構 (KEK)
岩瀬広

(国) 東京工業大学 (TITech)
千葉敏

ウィーン工科大学, オーストリア
Lembit Sihver

また、これまでに以下の方々が *PHITS* の開発に寄与されました。

原子力機構
高田弘、明午伸一郎、勅使河原誠、前川藤夫、原田正英、池田裕二郎、坂本幸夫、野田秀作

東北大学工学部
中村尚司

Chalmers University, Sweden
Davide Mancusi

1.3 PHITS の参考文献

バージョンに関わらず、PHITS をご使用になられた場合は次の文献を引用してください。

- T. Sato, K. Niita, N. Matsuda, S. Hashimoto, Y. Iwamoto, S. Noda, T. Ogawa, H. Iwase, H. Nakashima, T. Fukahori, K. Okumura, T. Kai, S. Chiba, T. Furuta and L. Sihver, Particle and Heavy Ion Transport Code System PHITS, Version 2.52, J. Nucl. Sci. Technol. 50:9, 913-923 (2013).

この文献はオープンアクセスとなっており、次の URL からダウンロードできます。

`http://dx.doi.org/10.1080/00223131.2013.814553`

この他に、PHITS がもつ特徴についてまとめた文献には次のものがあります。

- H. Iwase, K. Niita, T.Nakamura, Development of general purpose particle and heavy ion transport Monte Carlo code. J Nucl Sci Technol. 39, 1142-1151 (2002).
- K. Niita, T. Sato, H. Iwase, H. Nose, H. Nakashima and L. Sihver, Particle and Heavy Ion Transport Code System; PHITS, Radiat. Meas. 41, 1080-1090 (2006).
- L. Sihver, D. Mancusi, T. Sato, K. Niita, H. Iwase, Y. Iwamoto, N. Matsuda, H. Nakashima, Y. Sakamoto, Recent developments and benchmarking of the PHITS code, Adv. Space Res. 40, 1320-1331 (2007).
- L. Sihver, T. Sato, K. Gustafsson, D. Mancusi, H. Iwase, K. Niita, H. Nakashima, Y. Sakamoto, Y. Iwamoto and N. Matsuda, An update about recent developments of the PHITS code, Adv. Space Res. 45, 892-899 (2010).
- K. Niita, N. Matsuda, Y. Iwamoto, H. Iwase, T. Sato, H. Nakashima, Y. Sakamoto and L. Sihver, PHITS: Particle and Heavy Ion Transport code System, Version 2.23, JAEA-Data/Code 2010-022 (2010).
- K. Niita, H. Iwase, T. Sato, Y. Iwamoto, N. Matsuda, Y. Sakamoto, H. Nakashima, D. Mancusi and L. Sihver, Recent developments of the PHITS code, Prog. Nucl. Sci. Technol. 1, 1-6 (2011).

2　インストール、コンパイル及び使用方法

PHITS は、Windows, Mac 及び Linux 上で動作する Fortran プログラムです。Windows 及び Mac に対しては、実行形式を含むインストーラを準備していますので、PHITS をコンパイルすることなく利用することができます。Linux に対しては実行形式を準備しておりませんので、makefile を用いて PHITS をコンパイルしてから利用する必要があります。また、Windows や Mac でも、必要に応じて PHITS を再コンパイルすることができます。

2.1　動作環境

PHITS は、Windows(XP 以降), Mac(OS X v10.6 以降), Linux, Unix など様々なコンピュータで動作しますが、快適に動作させるためには、メモリが 2GB 以上搭載されていることが望ましいです。また、PHITS をインストールするためには、約 4GB 以上のハードディスク空き容量が必要となります (推奨は 6GB 以上)。

PHITS を実行するために必要なソフトウェアは特にありません。ただし、PHITS の入力ファイルを作るためには、行番号を表示可能なテキストエディタがインストールされていることが望ましいです (エラーが生じたときに、原因となる入力ファイルの行番号が表示されるため)。また、画像出力ファイル (EPS 形式) を見るためには、Ghostscript 及び GSview をインストールする必要があります。Windows 用のフリーのテキストエディタは、

- TeraPad 1.08（日本語）(http://www5f.biglobe.ne.jp/ t-susumu/)
- Crimson Editor（英語）(http://www.crimsoneditor.com/)

などがあります。Ghostscript 及び GSview のインストールに関しては、下記のホームページをご参照ください。

- Ghostscript (http://www.ghostscript.com/)
- GSview (http://pages.cs.wisc.edu/ ghost/gsview/index.htm)

PHITS で使用するメモリ枠を拡張する場合 (2.9 参照) や、`usrsors.f` ファイルを使って線源を定義する場合 (4.3.15 参照) は、PHITS を再コンパイルする必要があります。事務局が奨励する PHITS 用コンパイラは Intel Fortran Compiler（11.1 以降）と gfortran（4.7 或いは 4.8）です。それ以外のコンパイラでは、コンパイル時や実行時にエラーが発生する可能性が高いです。

2.2　Windows でのインストール及び実行方法

(1) 古いバージョンの PHITS をインストールしている場合は、そのフォルダ名を変更 (同じフォルダにインストールしない)

(2) USB メモリもしくは DVD にある `setup-jpn.vbs` をダブルクリック

(3) インストールフォルダを指定 (`c:\` を奨励します)

(4) `\phits\lecture\basic\lec01\lec01.inp` を右クリックして「送る」→「PHITS」

(5) `xz_track_all.eps` が作成されたことを確認

メモリ共有型並列で実行する場合は `\phits\bin` フォルダにある `phits.bat` の 3 行目でコア数を指定してください。例えば 4 コアで計算する場合は、

```
set PHITS_PARALLEL=4
```

と設定してください。或いは、

```
set PHITS_PARALLEL=0
```

とした場合は、計算機が持つ全てのコアを使用します。ただし、バージョン 2.73 より、メモリ共有型並列計算は 64bit 版 Windows のみで動作するようにしていますのでご注意ください。

インストーラは、下記の内容を実施します。

(1) phits.zip を指定フォルダに解凍する

(2) PHITS 実行形式を含むフォルダ \phits\bin に PATH を通す。

(3) \phits\bin フォルダにある phits.bat、angel.bat と \phits\dchain-sp\bin フォルダにある dchain.bat のショートカットを sendto フォルダに作成する。

(4) \phits\data フォルダにある核データリストファイル xsdir.jnd の 1 行目を datapath=インストールフォルダ +\phits\XS に書き換える。

2.3 Mac でのインストール及び実行方法

インストール方法

USB メモリや DVD に入っている Mac フォルダの中（図 1 参照）の phits_installer をダブルクリックし、インストール先のフォルダを指定する（図 2 参照）と、インストール先に phits という名前のフォルダが作られます。このフォルダには PHITS 本体とソース、講習会の資料、例題などがすべて入っています。

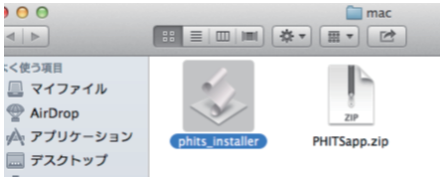


図 1: Mac フォルダの中

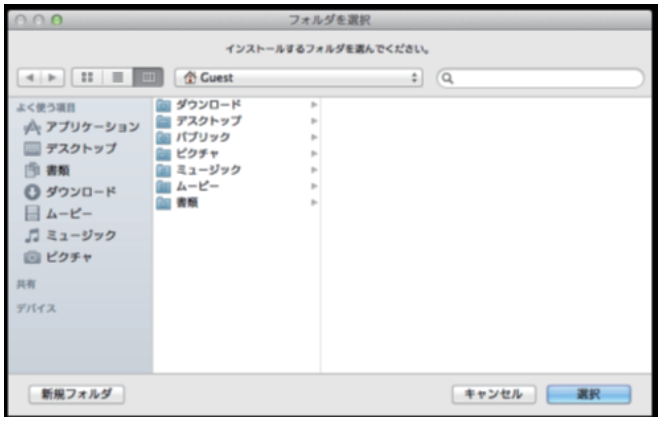


図 2: PHITS をインストールするフォルダを指定する画面

(注 1) インストール後に phits フォルダを別のフォルダへ移動させると PHITS は動作しなくなります。その場合はもう一度インストールし直してください。

(注 2) インストール先に phits という名前のフォルダが存在する場合は、古いフォルダは phits[今日の日付].[現在の時刻] に名称が変更されます。

(注 3) PHITS パッケージをコピーするフォルダやインストールフォルダの名称にスペースや漢字があるとエラーになる場合がありますのでご注意ください。

実行方法

PHITS を実行する場合は、ドラッグ&ドロップによる方法とターミナルを用いる方法があります。

ドラッグ&ドロップによる実行方法

ドラッグ&ドロップで実行する場合は、インプットファイルを Dock にある青い PHITS アイコンへドラッグ&ドロップします（図 3 参照）。計算結果はインプットファイルがあるフォルダに出力されます。

図 3: Dock にある PHITS アイコンへのドラッグ&ドロップ

メモリ共有型並列で実行する場合は、次の手順で Dock にある PHITS の設定ファイルを書き換えてください。

(1) DOC の PHITS アイコンを右クリックする。

(2) “オプション”→“Finder に表示” を選択。（/phits-office/phits/bin が表示される。）

(3) bin フォルダの PHITS アイコンを右クリックする。

(4) “パッケージの内容を表示” を選択。

(5) Contents/document.wflow をテキストエディタで開く。

(6) 63 行目付近のシングルに対応する実行ファイルの行を次のようにコメントアウトする。

```
# phitsexe="/Users/phits-office/phits/bin/phits282_mac.exe"
```

(7) その下の openMP に関する文章を次のように修正する。
（OMP_NUM_THREADS=の後の数字で使用するコア数を指定してください。）

```
# If you would like to use OpenMP version, please use the following commands
 phitsexe="/Users/phits-office/phits/bin/phits282_mac_openmp.exe"
# Please input your machine thread number when you use OpenMP version
 export OMP_NUM_THREADS=4
```

(8) Contents/document.wflow を保存する。

また、ANGEL 及び DCHAIN-SP を動かす際は、PHITS のタリーで出力されたそれぞれのインプットファイルを赤い ANGEL アイコンもしくは緑の DCHAIN アイコンにドラッグ&ドロップします。また、PHITS アイコンには、PHITS、ANGEL、DCHAIN-SP のインプットファイルを自動識別する機能がついておりますので、ANGEL や DCHAIN-SP のインプットファイルを PHITS アイコンにドラッグ&ドロップして ANGEL や DCHAIN-SP を起動することも可能です。

ターミナルからの実行方法

ターミナルからコマンド入力により PHITS を実行できます。「Finder」→「アプリケーション」→「ユーティリティ」→「ターミナル」を選択して、ターミナルを起動してください。

ターミナルを使って PHITS を初めて実行する場合は，PHITS の実行ファイルがあるフォルダに PATH を通す必要があります。PATH を通すためには，ターミナルで下記のコマンドを入力する必要があります。

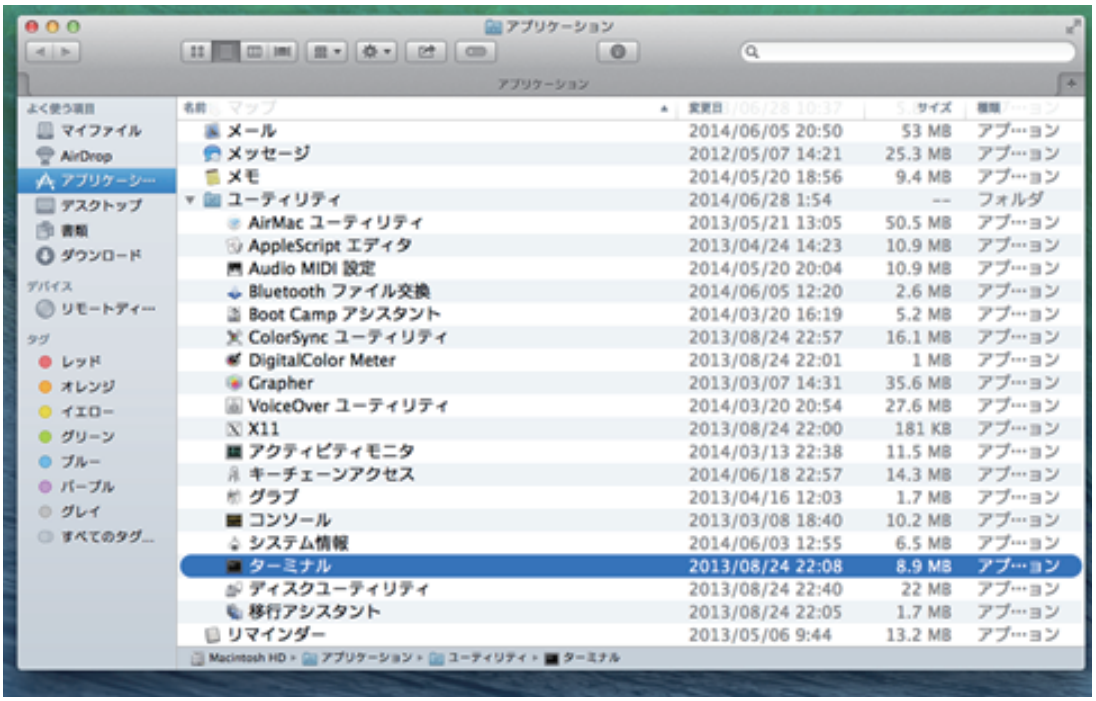


図 4: ターミナルを選択する画面

```
echo export PATH=/PATH-TO-PHITS/phits/bin:${PATH} >> .bash_profile
source .bash_profile
```

ここで「/PATH-TO-PHITS」は各自のインストール先のフォルダ名に変更してください（例 :/Users/iwamoto）。フォルダ名が分からない場合は，ターミナルを立ち上げた時点で pwd コマンドを実行して調べてください。なお，この PATH の設定は，初めて PHITS を実行する時のみ必要となり，それ以降は不要です。

PHITS の実行方法は,「cd」コマンドを使ってインプットファイルのあるフォルダに移動した後，ターミナルで下記のように入力します。

```
phitsXXX_mac.exe < your_input
```

ここで「XXX」は PHITS のバージョンを表します（例 : 282）。your_input は PHITS のインプットファイル名です（例 : lec01.inp）。ターミナルで「↑」キーを押せば過去のコマンド履歴が出ますので，同じインプットファイルを何度も実行する場合に便利です。これらの流れを図 5 に示します。

PHITS と同様，ANGEL や DCHAIN-SP もターミナルから実行できます。ANGEL の実行方法は，ターミナルで

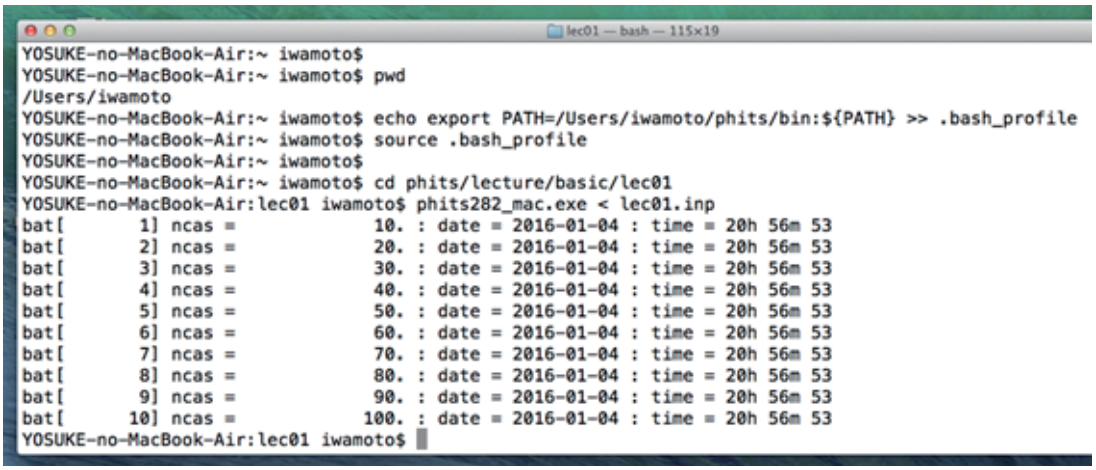
```
angel_mac.exe
```

と入力した後，次の行にインプットファイル名（PHITS のタリー出力，例えば track_xz.out）を入力します。

ターミナルから DHCAIN-SP を使う場合は，PHITS と同様に DCHAIN-SP の実行ファイルがあるフォルダに PATH を通す必要があります。PATH の通し方は，PHITS の場合と同様に

```
echo export PATH=/PATH-TO-PHITS/phits/dchain-sp/bin:${PATH} >> .bash_profile
source .bash_profile
```

となります。ここで「/PATH-TO-PHITS」は各自のインストール先のフォルダ名に変更してください（例 : /Users/iwamoto）。DCHAIN-SP の実行方法は，ターミナルで



```
lec01 — bash — 115×19
YOSUKE-no-MacBook-Air:~ iwamoto$
YOSUKE-no-MacBook-Air:~ iwamoto$ pwd
/Users/iwamoto
YOSUKE-no-MacBook-Air:~ iwamoto$ echo export PATH=/Users/iwamoto/phits/bin:${PATH} >> .bash_profile
YOSUKE-no-MacBook-Air:~ iwamoto$ source .bash_profile
YOSUKE-no-MacBook-Air:~ iwamoto$
YOSUKE-no-MacBook-Air:~ iwamoto$ cd phits/lecture/basic/lec01
YOSUKE-no-MacBook-Air:lec01 iwamoto$ phits282_mac.exe < lec01.inp
bat[       1] ncas =            10. : date = 2016-01-04 : time = 20h 56m 53
bat[       2] ncas =            20. : date = 2016-01-04 : time = 20h 56m 53
bat[       3] ncas =            30. : date = 2016-01-04 : time = 20h 56m 53
bat[       4] ncas =            40. : date = 2016-01-04 : time = 20h 56m 53
bat[       5] ncas =            50. : date = 2016-01-04 : time = 20h 56m 53
bat[       6] ncas =            60. : date = 2016-01-04 : time = 20h 56m 53
bat[       7] ncas =            70. : date = 2016-01-04 : time = 20h 56m 53
bat[       8] ncas =            80. : date = 2016-01-04 : time = 20h 56m 53
bat[       9] ncas =            90. : date = 2016-01-04 : time = 20h 56m 53
bat[      10] ncas =           100. : date = 2016-01-04 : time = 20h 56m 53
YOSUKE-no-MacBook-Air:lec01 iwamoto$
```

図 5: ターミナルを使った PHITS 実行の流れ（初期設定時）

```
dchain.sh your_input
```

と入力します。ここで your_input は，DCHAIN-SP の入力ファイル名（PHITS の [t-dchain] で指定したファイル名）です。

2.4 Makefile を利用したコンパイル

src フォルダ内にある makefile を使って、PHITS をコンパイルすることができます。そのためには、makefile を各自の環境に合わせて書き換える必要があります。具体的には、このファイルの最初の辺りにある ENVFLAGS に計算環境に応じた変数をセットしてください（例えば、Linux で Intel Fortran compiler を使用する場合は LinIfort）。また、メモリ分散型並列及びメモリ共有型並列を使用する場合は、USEMPI や USEOMP のある行の “#”を消して、これらを有効にしてください。作成される実行ファイル名は、phits_XXX.exe（XXX は ENVFLAGS にセットした変数）のようになります。例えば、ENVFLAGS=LinIfort の場合は、phits_LinIfort.exe が実行ファイル名です。更に、USEOMP を有効にした場合は phits_LinIfort_OMP.exe のようになります。makefile 中のコンパイラオプションなどは一例ですので、場合によってはそれらを変更することによりうまくコンパイルできる可能性があります。なお、付属の makefile が想定しているのは GNU make です。もし、make コマンドで動作しない場合は、gmake をお試しください。

Windows 用コンパイラとして gfortran をインストールする場合は、

- TDM-GCC (http://tdm-gcc.tdragon.net/download)

にアクセスし、このページからダウンロードできるインストーラーを使って行なってください[7]。なお、このパッケージをインストールした場合、make コマンドとして mingw32-make が利用できるようになります。

バージョン 2.50 以降、PHITS 内での大部分のメモリ使用方法を static から dynamic に変更したため、f77 や g77 ではコンパイルできないようになりました。PHITS 事務局で推奨しているコンパイラは Intel Fortran Compiler（11.1 以降）と gfortran（4.7 或いは 4.8）です。これら以外のコンパイラによる不具合に関しては十分なサポートができないことをご了承下さい。なお、makefile に書かれていないオプションでコンパイル&実行に成功した方は、PHITS 事務局までご連絡下さい。

[7] ただし、4.9 以降の gfortran のバージョンでは適切にコンパイルされない場合があります。その場合は 4.7 や 4.8 のバージョンをご使用ください。このバージョンのインストール方法については、\phits\document\Install_gfortran4-8-jp.pdf をご覧ください。

2.5 Microsoft Visual Studio と Intel Fortran を利用したコンパイル (Windows 用)

`\phits\bin` フォルダ内に、Microsoft Visual Studio と Intel Fortran を組み合わせた環境に対するソリューションファイル (`bin.sln`) とプロジェクトファイル (`phits-intel.proj`) が含まれています。以下の手順に従って、これらのファイルを使って PHITS をコンパイル、実行できます。

(1) `bin.sln` をダブルクリック (Visual Studio や Intel Fortran のバージョン新しい場合は自動更新する場合があります。また、Visual Studio 2005 以前や Intel Fortran 11.1 より前のバージョンではうまく開けない場合があります)

(2) リリースモードで `phits-intel.proj` をビルドする（1 度で成功しない場合は何度か試してみてください。）

(3) `bin` フォルダ内に PHITS のインプットファイルを作成

(4) リリースモードで実行

(5) 起動した実行画面で `file=PHITS` インプットファイル名と入力

(6) `xz_track_all.eps` が作成されたことを確認

メモリ共有型並列計算用の実行ファイルを作成する場合は、「ビルド」する前に、`phits-intel.vfproj` ファイルの `Source files` にある一覧表で `a-angel.f` を `a-angel-winopenmp.f` に変更し、`/Qopenmp` オプションを追加してください。（Microsoft Visual Studio ウインドーの上部メニューから「プロジェクト」→「`phits-intel` のプロパティ」を選択肢、「Fortran」→「コマンドライン」の「追加のオプション」に追加します。）

もし、作成した実行ファイルを「送る」機能で使用する場合は、`\phits\bin` フォルダにある `phits.bat` をテキストエディタで開き、中に書かれている環境変数の `PHITS_EXE` を、例えば次のように、書き換えてください。

```
set PHITS_EXE=C:\phits\bin\Release\phits-intel.exe
```

実行ファイル名の方を `phits.bat` に書かれている名称（例えば、phits282_win.exe）に変更しても動作しますが、PHITS の更新を行う際に必要となりますので、元のファイルを消さないようにご注意ください。

2.6 ANGEL のコンパイル

ANGEL は、簡単なインプットから、EPS (Enhanced PostScript) 形式のグラフを素早く描くために設計されたプログラムです。すなわち、*ANGEL* は、Angel 言語 (数値データファイルをグラフ化するために書き加える必要最低限の命令) から PostScript 言語 (Adobe 社のグラフィックコントロールプログラムの規格のプログラム言語) への翻訳機です。*ANGEL* は、PHITS のソースに含まれ、また PHITS の出力もアスキーファイルの他に `eps` ファイルの出力を得ることができますが、その後のグラフの整形などに *ANGEL* が必要になる場合があります。単体の *ANGEL* をコンパイルするためには、`src` フォルダにある `make.ang` を `makefile` に名称変更して `make` する必要があります。その他の *ANGEL* に関することは、*ANGEL* のマニュアルを参照してください。

2.7 実行シェル

PHITS を実行するための特別なシェルは必要ありません。PHITS のプログラムは、標準入力からインプットパラメータを読み込み、標準出力にサマリーとエラーメッセージを書き出します。その他の出力ファイルの指定は、インプットデータで行います。従って、最も簡単な実行コマンドは、

List 2.1 ● command line to execute PHITS

```
phits100 < input.dat > output.dat
```

となります。ここで、phits100 は PHITS の実行ファイル、input.dat は PHITS の書式で書かれたインプットファイル、output.dat は標準出力の内容が書き出されるアウトプットファイルです。

Windows 系の環境で実行する場合も同様です。ただし、Windows 系では標準入力の rewind が使えないため、パラメータ infl を用いて input.dat 以外の外部ファイルを利用する場合はエラーとなります。そこで infl を使用する場合は、input.dat の 1 行目を

List 2.2 ● the first line of the standard input

```
file = input.dat
```

としてください。このときプログラムは、input.dat というファイルを改めてオープンして、インプットデータを読み込みます。この方法は、Windows 系以外でも使えます。infl の使い方については、3.3 節をご覧ください。

また、メモリ分散型並列では標準入力からの読み込みをしない仕様になっています。実行ディレクトリの phits.in のファイルから入力ファイル名を読み込みます。この phits.in は、固定です。このファイルの 1 行目に

```
file = input.file
```

のように入力ファイル名を記述します。これは、メモリ分散型並列だけの制約です。

2.8 実行の途中中断

プログラムを実行すると、カレントディレクトリに batch.out [8] というファイルが作られます。そのファイルには、ひとつのバッチが終了する毎に、メモリ分散型並列の場合には、バッチ数 × (PE −1) 毎に、そのバッチの計算時間など、簡単な情報が出力されます。メモリ分散型並列の場合には、各 PE の状態が含まれます。異常終了が起こった PE はこれでチェックできます。このファイルの 1 行目は、

```
1 <--- 1:continue, 0:stop
```

となっています。この最初の “1” を “0” に書きかえると、次のバッチで計算が終了し、そこまでのサマリー、タリーの出力をします。

これに関係して、[parameters] セクションにおいて itall を設定できます。

```
itall = 1  # (D=0) 0:no tally at batch, 1:same, 2:different
```

のように指定します。itall = 1 とすることで、バッチ毎に（メモリ分散型並列の場合にはバッチ数 × (PE −1) 毎に）、タリーの途中結果をユーザーが指定したファイルに、上書きで出力します。

この二つの機能で途中経過を見ながら、計算を停止することが可能です。また、タリーの epsout のパラメータを使うと、途中結果をグラフでモニターすることができます。(5.7.16 を参照)

バージョン 2.86 以降、[parameters] セクションにおいて file(22) を指定することにより、batch.out のファイル名を変更できるようになりました。これにより、同一ディレクトリにおいて複数のインプットファイルに対して PHITS を実行することが可能となりました。ただし、[t-dchain] を含んだインプットファイルの場合、“n.flux” などのファイル名は同じものが使用されるのでご注意ください。

batch.out に出力される rijk は、各バッチ毎 (各 IP number 毎) に与えられた初期乱数の値です。特定のバッチで異常終了が起こった時など、任意のバッチの計算を再現したい場合にこの値を利用できます。

[8] Version 2.85 以前は batch.now というファイル名でした。

2.9 配列の大きさの変更

インクルードファイル param.inc の中に、ユーザーが場合により変更しなければならない配列の大きさが記述されています。特に重要なのが mdas で、これは幾何形状とタリー、核データ関係、また、バンクの配列に必要な配列の大きさを指定します。インプットエコーを見て現在の使用量をチェックしてみてください。バンクの配列の大きさは、パラメータセクションで指定します。mdas の余った配列は、バンクが足りなくなった場合に自動的に使われます。

以下に現在のデフォルト値が記述された param.inc を示します。

List 2.3 ● param.inc

```
 1:  ************************************************************************
 2:  *                                                                      *
 3:  *     'param.inc'                                                      *
 4:  *                                                                      *
 5:  ************************************************************************
 6:
 7:        parameter ( mdas   =120000000 )
 8:        parameter ( kvlmax =    3000 )
 9:        parameter ( kvmmax = 1000000 )
10:        parameter ( itlmax =      60 )
11:        parameter ( inevt  =      70 )
12:        parameter ( isrc   =      50 )
13:        parameter ( latmax = 25000000 )
14:
15:        common /mdasa/ das( mdas )
16:        common /mdasb/ mmmax
17:
18:  *----------------------------------------------------------------------*
19:  *                                                                      *
20:  *       mdas   : total memory * 8 = byte                               *
21:  *       mmmax  : maximum number of total array                         *
22:  *                                                                      *
23:  *       kvlmax : maximum number of regions, cell and material          *
24:  *       kvmmax : maximum number of id for regions, cel and material    *
25:  *                                                                      *
26:  *       itlmax : number of maximum tally entry                         *
27:  *       inevt  : number of collision type for summary                  *
28:  *       isrc   : number of multi-source                                *
29:  *       latmax : maximum number of lattice in a cell + 1               *
30:  *                                                                      *
31:  *----------------------------------------------------------------------*
```

3　入力ファイルの書式

新しい入力ファイルは、[******] で始まる次の様な各セクション単位でデータを読み込みます。セクションの順番は自由です。[******] の先頭の空白は 4 個まで無視しますが、それ以上はセクションの始まりとは認識しませんので注意して下さい。

3.1　セクションの種類

セクションには、次の様なものがあります。

表 1: セクションの種類 (1)

name	説明
[title]	計算のタイトルを定義します。
[parameters]	ヒストリー数など、計算の動作を決めるパラメータを定義します。
[source]	入力ソース粒子の情報を定義します。
[material]	体系を構成する物質を定義します。
[surface]	体系を構成する面を定義します。
[cell]	体系を構成する cell を定義します。
[transform]	体系を構成する面などの座標変換を定義します。
[temperature]	cell の温度を定義します。
[mat time change]	物質の時間変化を定義します。
[magnetic field]	体系を構成する領域の磁場を定義します。
[electro magnetic field]	体系を構成する領域の電磁混合場を定義します。
[delta ray]	δ 線を発生させます。
[super mirror]	低エネルギー中性子のスーパーミラーを定義します。
[elastic option]	低エネルギー中性子の弾性衝突のオプションを定義します。
[data max]	核種毎に核データライブラリを利用する上限エネルギーを定義します。
[frag data]	ユーザー指定断面積読み込み機能について定義します。
[importance]	領域の importance を定義します。
[weight window]	領域の weight window を定義します。
[forced collisions]	強制衝突の領域とファクターを定義します。
[brems bias]	bremsstrahlung の bias を定義します。
[photon weight]	photon 生成の weight を定義します。
[volume]	体系を構成する領域の体積を定義します。
[multiplier]	multiplier を定義します。
[mat name color]	物質表示の名前、色、大きさを指定します。
[reg name]	領域表示の名前、大きさを指定します。
[counter]	counter を定義します。
[timer]	粒子の固有時間をリセット、ストップを定義します。

表 2: セクションの種類 (2)

name	説明
[t-track]	粒子の飛跡長 (track length) やフルエンスを導出するタリー。
[t-cross]	粒子の面横断回数やフルエンスを導出するタリー。
[t-point]	ある点や線上のフルエンスを導出するタリー。
[t-heat]	物質におけるエネルギー付与を導出するタリー。
[t-deposit]	荷電粒子の物質におけるエネルギー付与を導出するタリー。
[t-deposit2]	2 つの領域でのエネルギー付与の相関を出力するタリー。
[t-yield]	残留核の生成量を導出するタリー。
[t-product]	線源や核反応による生成粒子を導出するタリー。
[t-dpa]	原子あたりのはじき出し数 (DPA) を導出するタリー。
[t-let]	LET の関数として飛跡長や線量を導出するタリー。
[t-sed]	微小領域におけるエネルギー付与分布を導出するタリー。
[t-time]	Energy cut off と escape 粒子の個数を導出するタリー。
[t-star]	Star density を導出するタリー。
[t-dchain]	DCHAIN-SP 用入力ファイルを作成するタリー。
[t-wwg]	[weight window] のパラメータを出力するタリー。
[t-userdefined]	ユーザー定義による任意の物理量を導出するタリー。
[t-gshow]	ジオメトリ（仮想空間）を 2 次元で表示するタリー。
[t-rshow]	ジオメトリを 2 次元で物理量による色分けをして表示するタリー。
[t-3dshow]	ジオメトリを 3 次元で表示するタリー。
[end]	入力ファイルの終了を示します。

これらのセクション単位でデータを読み込みます。従って、入力ファイルの先頭から最初のセクションまで、また [end] 以下の部分は読み飛ばします。

3.2 読み込みコントロール

(1) 大文字、小文字、空白
ファイル名を除いて、大文字小文字は区別しません。行頭、行末の空白は無視します。セクション名は間に空白を入れてもかまいません。ただし、[******] の先頭の空白は 4 個まで無視しますが、それ以上はセクションの始まりとは認識しませんので注意して下さい。

(2) タブ
タブは、8 個のブランクに置き換えます。

(3) 継続行
一行に書くことができる文字数は半角文字で 200 字までです。それ以降の文字は読み込まれません。一行に収まらないデータは、行末に “ ¥ ” をつけると次行を含めて一行と扱います。複数行が可能です。ただし、[cell]、[surface] の継続行は、先頭に 5 個以上の空白が必要です。

(4) 行連結
短いパラメータ文が続くとき、複数行を“ ; ”でつなぐことにより一行に表示することができます。例えば、

```
idbg = 0 ;  ibod = 1 ;  naz =  0
```

のようにです。但し、メッシュの記述のようにサブセクションの書式が決まっているときは、使えません。

(5) コメント文字
コメント文字として、#, %, !, $ の 4 文字が定義されています。これらの文字以降行末まで無視されます。また、左から 5 カラムまでに“c”が書かれている行はコメント行となり、読み飛ばされます。このため、[material] セクションで炭素の自然同位体を定義するときに、“C”、とするとコメント行として扱われます。この場合は、6000、で指定するようにしてください。
(注意: [cell], [surface] セクションの場合、セルの定義に“#”を用いますので、コメント文字として“$”以外は、このセクションでは利用できません。)

(6) 空白行
空白行は、読み飛ばします。コメント文字で始まる行も読み飛ばします。

(7) セクションの読み飛ばし
ひとつのセクションを読み飛ばしたいとき、[******] off のようにセクション名の後に“off”をつけると、次のセクションまでを読み飛ばします。

(8) セクション途中からの読み飛ばし
セクションの途中で、行頭に qp: を挿入すると、この行以降次のセクションまでを読み飛ばします。

(9) 全ての読み飛ばし
入力ファイルのどこでも、q: を行頭に入れると、それ以降全てを読み飛ばします。これは、[end] と同じです。

3.3 ファイルの挿入

入力ファイルのどの場所でも他のファイルを取り込めます。インクルードファイルの書式は、

infl: { *file.name* } [$n_1 - n_2$]

です。{ } 内がファイル名、[ ] 内が行数指定です。n_1 行から n_2 行をインクルードします。行数指定は省略できます。省略した場合は、そのファイル全てをインクルードします。また、

[n_1−]
[−n_2]

という表現も可能です。上が n_1 行から最後まで、下が先頭から n_2 行までをインクルードします。インクルードは、何重にもネストできます。インクルードファイルを読み終えると、一層上のファイルに戻ります。

WINDOWS 系の環境でコマンドラインインタプリタ (コマンドプロンプト) により *PHITS* を実行している場合に、`infl` を使用する際は注意が必要です。このときは、インプットファイルの 1 行目を

```
file = input.dat
```

としてください。ここで、`input.dat` はインプットファイルです。コマンドラインインタプリタを用いた実行方法については、2.7 節をご参照ください。

3.4 ユーザー定義定数

入力ファイルの数字の入力箇所で、ユーザーが定義した定数が利用できます。定数のセットの書式は、

```
set:  c1[ 52.3 ] c2[ 2 * pi ] c3[ c1 * 1.e-8 ]
```

この定数セットは、どの場所でも可能です。定数名は、`c1` から `c99` まで、何度でもセットし直せます。セットした以降、次にセットされるまで、その値を保持します。定義の中で他の定数を用いる場合 (上の 3 番目の例)、その時点での値が参照され、それ以降、定義の中で用いられている定数 (上の 3 番目の例では `c1`) をリセットしても定数の値は変化しません。`pi` は、デフォルトで定義されている定数です。

3.5 数式の利用

入力ファイルで数字の入力箇所で、数式の利用が可能です。数式の書式は、FORTRAN 形式です。利用できる内部関数は、

表 3: Intrinsic Function.

Intrinsic Function							
FLOAT	INT	ABS	EXP	LOG	LOG10	MAX	MIN
MOD	NINT	SIGN	SQRT	ACOS	ASIN	ATAN	ATAN2
COS	COSH	SIN	SINH	TAN	TANH		

例として、

```
param = c1 * 3.5 * sin( 55 * pi / 180 )
```

この例のように、`param =` の後に 1 個の数字が期待されるときは、数式の途中に空白が許されますが、幾つかの数字の並びが期待される場合は、空白が数字の区切りを示しますので空白は使えません。このときは、空白を省いて記述するか、`{ c1 * 2 / pi }` のように `{ }` で括ってください。

3.6 粒子の表式

`[source]` セクションや、タリーセクションで用いる 粒子の表式を下の表にまとめます。粒子を定義するときは、symbol か kf-code を用います。symbol のないものは、kf-code だけ利用可能です。

粒子タイプ 11 の other particles は、以下のように kf コードで指定します。また、これらの粒子の崩壊チャンネルは以下のようなものが含まれています。

JQMD コードの導入に伴い、原子核の入射、輸送が可能となりました。原子核種の指定の仕方は、208Pb, 56Fe の形式です。Pb, Fe と指定すれば、同位体全てを意味します。もちろん入射粒子の指定には使えません。粒子タイプ 20 番の nucleus は、全ての原子核から 15 番から 18 番、deuteron, triton, 3He, alpha を除いた全てを意味します。kf コードで原子核を指定するには、kf = Z * 1000000 + A とします。

以前は、光子の名前を “gamma”としてきましたが、“photon”と変更しましたので注意して下さい。

表 4: List of the transport particles.

ityp	symbol	kf-code	particle name
1	proton	2212	proton
2	neutron	2112	neutron
3	pion+	211	π^+
4	pion0	111	π^0
5	pion−	−211	π^-
6	muon+	−13	μ^+
7	muon−	13	μ^-
8	kaon+	321	K^+
9	kaon0	311	K^0
10	kaon−	−321	K^-
11	other	below	other particle
12	electron	11	e^-
13	positron	−11	e^+
14	photon	22	γ
15	deuteron	1000002	deuteron
16	triton	1000003	triton
17	3he	2000003	^{3}He
18	alpha	2000004	α
19	nucleus	Z*1000000+A	nucleus
20	all	−	all particles

ityp	symbol	kf-code	particle name
11	−	+ − 12	$\nu_e\ \overline{\nu_e}$
11	−	+ − 14	$\nu_\mu\ \overline{\nu_\mu}$
11	−	−2212	$\bar{p}$
11	−	−2112	$\bar{n}$
11	−	−311	$\overline{K^0}$
11	−	+ − 221	$\eta\ \bar{\eta}$
11	−	331	η'
11	−	+ − 3122	$\Lambda^0\ \overline{\Lambda^0}$
11	−	+ − 3222	$\Sigma^+\ \overline{\Sigma^+}$
11	−	+ − 3212	$\Sigma^0\ \overline{\Sigma^0}$
11	−	+ − 3112	$\Sigma^-\ \overline{\Sigma^-}$
11	−	+ − 3322	$\Xi^0\ \overline{\Xi^0}$
11	−	+ − 3312	$\Xi^-\ \overline{\Xi^-}$
11	−	+ − 3334	$\Omega^-\ \overline{\Omega^-}$

表 5: 崩壊チャンネルと寿命

								分岐率	寿命 (sec)
π^0	$\rightarrow$	γ	+	γ				100%	0
π^+	$\rightarrow$	μ^+	+	ν_μ				100%	2.6029e−8*
π^-	$\rightarrow$	μ^-	+	ν_μ				100%	2.6029e−8
μ^+	$\rightarrow$	e^+	+	$\overline{\nu_e}$	+	ν_μ		100%	2.19703e−6
μ^-	$\rightarrow$	e^-	+	$\overline{\nu_e}$	+	ν_μ		100%	2.19703e−6
K^0	$\rightarrow$	π^+	+	π^-				68.61%	8.922e−11
	$\rightarrow$	π^0	+	π^0				31.39%	
	$\rightarrow$	γ	+	γ				other	
K^+	$\rightarrow$	μ^+	+	ν_μ				63.51%	1.2371e−8
	$\rightarrow$	π^+	+	π^-				other	
K^-	$\rightarrow$	μ^-	+	ν_μ				63.51%	1.2371e−8
	$\rightarrow$	π^+	+	π^-				other	
η	$\rightarrow$	γ	+	γ				38.9%	0
	$\rightarrow$	π^0	+	π^0	+	π^0		31.9%	
	$\rightarrow$	π^+	+	π^-	+	π^0		23.7%	
	$\rightarrow$	π^+	+	π^-	+	γ		other	
η'	$\rightarrow$	π^+	+	π^-	+	η		44.1%	0
	$\rightarrow$	π^0	+	π^0	+	η		20.5%	
	$\rightarrow$	π^+	+	π^-	+	γ		30.1%	
	$\rightarrow$	γ	+	γ				other	
Λ^0	$\rightarrow$	p	+	π^-				64.1%	2.631e−10
	$\rightarrow$	n	+	π^0				other	
Σ^+	$\rightarrow$	p	+	π^0				51.57%	7.99e−11
	$\rightarrow$	n	+	π^+				other	
Σ^0	$\rightarrow$	Λ^0	+	γ				100%	0
Σ^-	$\rightarrow$	n	+	π^-				100%	1.479e−10
Ξ^0	$\rightarrow$	Λ^0	+	π^0				100%	2.90e−10
Ξ^-	$\rightarrow$	Λ^0	+	π^-				100%	1.639e−10
Ω^-	$\rightarrow$	Λ^0	+	K^-				67.8%	8.22e−11
	$\rightarrow$	Ξ^0	+	π^-				23.6%	
	$\rightarrow$	Ξ^-	+	π^0				other	

*2.6029×10^{-8}

4　セクション書式

4.1　[Title] セクション

このセクションでは、計算のタイトルを定義します。タイトルは、標準出力の先頭に印字されます。書式は次の様なものです。何行でもかまいません。ただし、このセクションでは空白行は無視されます。

```
[ T i t l e ]
This is a test calculation of PHITS.
Any number of title lines are allowed.
..........
```

4.2 [Parameters] セクション

このセクションでは、*PHITS* の動作を決定するパラメータを指定します。書式は次の様なものです。

```
[ P a r a m e t e r s ]
para1 = number | file.name
para2 = number | file.name
..........
```

パラメータの順番は自由、省略すれば、デフォルト値が読み込まれます。

以下に各々のパラメータの意味と値を説明します。(D=) は、デフォルト値です。

4.2.1 計算モード

表 6: パラメータ 1

パラメータ	値	説明
icntl	(D=0)	基本動作オプション
	= 0	normal *PHITS* 計算
	= 1	nuclear reaction 計算、 未完成
	= 2	CGVIEW インプットファイル書き出し
	= 3	input echo だけの書き出し
		メモリー空間、ライブラリー、ファイルのチェック
	= 4	MARS-PF インプットファイル書き出し
	= 5	no reaction, no ionization, 全てを void にする。
		geometry check, 体積、面積計算に使う
	= 6	ソースのチェック。[t-product] でソース粒子をタリーできます
	= 7	[t-gshow] タリーの実行
	= 8	xyz メッシュタリーの gshow オプションのあるものの幾何形状表示
	= 9	[t-rshow] タリーの実行
	= 10	reg メッシュタリーの rshow オプションのあるものの幾何形状表示
	= 11	[t-3dshow] 3 次元幾何形状表示タリーの実行
	= 12	dumpall ファイルから読み込んで再計算。file(15) でファイルを指定
	= 13	タリー結果の統合機能（Sumtally 機能）の利用

icntl=1 の nuclear reaction 計算は、開発が取り残されているため、現在、まだ工事中と考えてください。

icntl=12 を指定すると、dumpall=1 で計算した時にダンプしたデータを基に再計算します。輸送に関しては、最初に計算した全ての情報を再現します。再計算の時には、最初に計算したインプットファイルが必要になります。タリーを除いて同じものを使わないと矛盾が生じます。maxcas, maxbch は、ダンプファイルに指定されているものを使いますので変更できません。最初に計算したときと異なるタリーを入れて再計算したいときなど有効です。ただし、dumpall=1 で計算するときは、巨大なデータファイルが作成されることがありますから注意してください。この dumpall オプションは、GG のときのみ有効です。dumpall=1 でダンプする情報は、4.2.21 のところで解説します。

icntl=13 を指定することにより、Sumtally 機能を利用することができます。この機能の詳細は 5.8 をご覧ください。

4.2.2 ヒストリー数、バンク配列の大きさ

表 7: パラメータ 2

パラメータ	値	説明
irskip	(D=0)	乱数のコントロール
	irskip>0	irskip 回ヒストリーをスキップして計算を開始 (デバッグ用)
	irskip<0	irskip 回ヒストリーの乱数をスキップして計算を開始 (手動並列用)
rseed	(D=0.0)	初期乱数オプション
	rseed<0	計算開始時間情報より初期乱数を設定
	rseed=0	6647299061401. デフォルトの初期乱数
	rseed>0	rseed を初期乱数とする
maxcas	(D=10)	1 バッチのヒストリー数。上限は 2147483647。
maxbch	(D=10)	バッチ数。上限は 2147483647。
maxbnk	(D=10000)	バンク配列の大きさ
istdev	(D=0)	新規 / 再開始計算及び統計誤差導出方法のコントロール。
	=-2	再開始計算 (ただし、過去のタリー結果が全くない場合は、istdev=2 として新規計算を開始。
	=-1	再開始計算 (ただし、過去のタリー結果が全くない場合は、istdev=1 として新規計算を開始。
	=0	新規計算。統計誤差は、メモリ共有型並列の場合はバッチ分散を、それ以外はヒストリー分散を自動的に選択します。
	=1	新規計算。統計誤差は、各バッチ間の分散により導出。
	=2	新規計算。統計誤差は、各ヒストリー間の分散より導出。
ireschk	(D=0)	再開始計算時のタリーパラメータ一致性確認の有無。
	=0	読み込むタリー出力の設定が現在のタリー設定と一致しているか確認する。
	=1	上記確認を行わない。極めて長いタリー設定をした場合、タリー出力に全ての設定が書き込まれずに、新旧ファイル一致性の確認でエラーが出てしまう場合があります。その場合、ireschk=1 とし、一致性を確認しないモードで再開始計算を実行してください。

メモリ分散型並列計算では、バッチ単位で並列化しています。従って、入力のバッチ数 maxbch は、並列の実行 PE 数 (コントロールに 1PE 使うので、全 PE 数 −1) の整数倍になるように指定して下さい。もし整数倍でないときは、整数倍になるように、またトータルヒストリー数が入力データとほぼ同じになるように、プログラムが自動で変換します。変換した場合は、出力のインプットエコーの最後にコメントが出力されます。

バージョン 2.50 より、タリー出力の統計誤差が正しく計算できるようになりました。また、統計が十分でなかった場合などに対応するため、古い計算結果から次の初期乱数とタリー出力を読み込んで再開始計算を行う機能を実装しました。

統計誤差計算は、各バッチ間のタリー量の分散より標準偏差を導出するバッチ分散モードと、各ヒストリー間のタリー量の分散より標準偏差を導出するヒストリー分散モードがあります。どちらも、次式で標準

偏差 σ を計算し、その値とタリー量の平均値 $\bar{X}$ の比を相対誤差として出力します。

$$\sigma = \sqrt{\frac{\sum_{l=1}^{N}(x_i w_i/\bar{w})^2 - N\bar{X}^2}{N(N-1)}} \tag{1}$$

ここで N は標本数 (`istdev=1` のときは全バッチ数、`istdev=2` のときは全ヒストリー数) であり、x_i, w_i は各標本のタリー量及びソースウェイト、$\bar{w}$ はソースウェイトの平均値です。メモリ共有型並列計算の場合は、原理的に各ヒストリーの分散を計算することが不可能ですので、バッチ分散モードのみ選択可能です。バッチ分散の場合は、同じ全ヒストリー数でも、`maxcas` と `maxbch` の組み合わせによって標準偏差が変化しますのでご注意ください。基本的には、`maxbch` を大きくした方が計算の信頼性は高くなりますので、正確に統計誤差を計算したい場合は、`maxbch` を少なくとも 10 以上としてください。ただし、*PHITS* は、各バッチ終了時にいろいろな処理をしますので、`maxbch` を大きくしすぎると計算時間が長くなりますのでご注意ください。一方、ヒストリー分散モードの場合は、標準偏差は全ヒストリー数のみに依存し、`maxcas` と `maxbch` の組み合わせには依存しません。したがって、メモリ共有型並列を行わない場合は、ヒストリー分散モードを選択することをお薦めします。しかしながら、xyz メッシュタリーを細かく設定する等、メモリを多く使用するタリーを設定している場合には、ヒストリー分散モードでの計算時間が莫大となることがありますのでご注意ください。なお、各ヒストリーの分散を導出するタリー (`[t-deposit]` タリーの `output = deposit` や `[t-deposit2]` など) の場合は、統計誤差として標準誤差を出力しており、`istdev` の値に関係なく、寄与したヒストリー数の平方根から計算します。すなわち、寄与したヒストリー数が K の場合、その相対誤差は $1/\sqrt{K}$ となります。

相対誤差は、通常、タリー量の右側の `r.err` カラムに出力されます。ただし、`axis = xy, rz` など 2 次元プロットの場合は、各タリーで指定した file 名の拡張子の手前に `_err` と付けた別ファイルに出力します。例えば、`file=tally.out` と指定した場合、相対誤差ファイルは `tally_err.out` となります。誤差ファイルの書式は通常のタリー出力と同じですので、*ANGELA* で処理することにより誤差の 2 次元プロットを作成することができます。

PHITS で評価しているのは標準偏差や標準誤差なので、必ずしも真値がこの誤差の範囲内に収まるわけではないのでご注意ください。また、核反応モデルに起因する系統誤差は評価されていませんのでご注意ください。

`istdev<0` とした場合は、再開始計算となります。その場合、統計誤差計算モードは、過去のタリー結果を踏襲します。再開始計算は、各タリー別に行いますので、過去のインプットファイルに新たにタリーを加えた再開始計算も可能です。再開始計算の流れは、以下のようになります。

(1) 各タリーの `resfile` (デフォルトは `file` で指定した出力ファイル名と同じ) で指定したファイルの有無をチェックする。

(2) ファイルが存在しない場合は、そのタリーは新規タリーとして扱う。全てのファイルが存在しない場合は、`istdev = -2` の場合はヒストリー分散、`istdev= -1` の場合はバッチ分散モードとして新規計算を開始する。

(3) ファイルが存在する場合は、そのファイルから統計誤差計算モード (`istdev`)、これまでの全ウェイト数 (`resc2`) や全ヒストリー数 (`resc3`)、バッチ当たりのヒストリー数 (`maxcas`、バッチ分散モードのみ必要)、次の初期乱数シード (`rijklst`)、タリーパラメータ、結果及び相対誤差などを読み込む。

(4) 過去のタリーと入力ファイルで指定したタリーの一致性の確認する (一致しない場合はエラーメッセージを出力して実行停止。ただし、全てのパラメータをチェックするわけではないので注意)。

(5) 過去のタリーが複数ある場合、それらの間での統計誤差モードや `maxcas` (バッチ分散モードのみ) の一致性を確認する。一致した場合は、その値を再開始計算で利用する。一致しない場合は、エラーメッセージを出力して実行停止する。

(6) 初期乱数シードを一番目に読み込んだ過去タリーの `rijklst` に変更。(`rijklst` が過去タリー間で異なる場合は、警告を出力するが計算は可能)

(7) 再開始計算を実行し、その結果を過去のタリー結果に加えて `file` で指定したファイルに出力する。`resfile` を指定しない場合、出力ファイル (`file`) と過去タリーファイル (`resfile`) が同じであるため、過去の結果に新しい結果を上書きして出力する。

再開始計算を行う際の注意点としては、[1] 統計誤差計算モードの違うタリー同士で再開始計算をしないこと、[2] バッチ分散の場合は必ず `maxcas` を統一すること、[3] ジオメトリや入射粒子に関しては新旧ファイルの一致性を確認しないので、ユーザーの責任で必ず一致させること、などがあります。

4.2.3 計算打切エネルギー、切り替えエネルギー

表 8: パラメータ 3

パラメータ	値	説明
`emin(1)`	`(D=1.0)`	陽子の計算打切エネルギー (MeV)。
`emin(2)`	`(D=1.0)`	中性子の計算打切エネルギー (MeV)。
`emin(i)`	`(D=1.0)`	`(i = 3-10)` パイオン、ミューオン、ケイオンの計算打切エネルギー (MeV)。粒子番号 `i` は表 4 を参照。
`emin(11)`	`(D=2.0)`	その他の粒子の計算打切エネルギー (MeV)。
`emin(i)`	`(D=1.e+9)`	`(i = 12-14)` 電子、陽電子、光子の計算打切エネルギー (MeV)。粒子番号 `i` は表 4 を参照。
`emin(i)`	`(D=1.0)`	`(i = 15-19)` 重陽子、トリチウム、^{3}He、アルファ粒子、原子核の計算打切エネルギー (MeV/u)。これらの粒子のエネルギーは核子あたりのエネルギー。粒子番号 `i` は表 4 を参照。
`esmin`	`(D=0.001)`	荷電粒子の range 計算の最小エネルギー (MeV/u)
`esmax`	`(D=300000)`	荷電粒子の range 計算の最大エネルギー (MeV/u)
`cmin(i)`	`(D=emin(i))`	i-th 粒子の核反応計算打切エネルギー (MeV/u)
		これ以下のエネルギーの核反応は無視する。
	`i = 15-19`	これらの粒子のエネルギーは、核子当たり
`dmax(i)`	`(D=emin(i))`	i-th 粒子のライブラリー利用の上限エネルギー

これらのパラメータで各々のエネルギー領域に制限を与える際は、下限の場合はその値以上、上限の場合はその値未満となります。したがって例えば、`emin(1)` で陽子の計算打切エネルギーの上限値を与えた場合、丁度その値のエネルギー (デフォルトだと 1.0MeV) をもつ陽子は計算打切とはなりません。

$emin < energy < dmax$ の範囲がライブラリーによる計算になります。$emin \geq dmax$ とすれば、ライブラリーを用いた計算をしません。陽子、中性子、光子、電子のライブラリーの上限は、現在のところ、それぞれ 3GeV、3GeV、100GeV、10GeV です。

荷電粒子の range 計算において、$esmin < energy < esmax$ の範囲で range テーブルを作成して計算します。より小さい、もしくは、より大きなエネルギーを取り扱いたい場合に設定して下さい。通常はデフォルトのままで計算を行います。

荷電粒子の計算打ち切りエネルギー `emin` を `esmin` よりも小さくすることはできません。その場合、`emin` は自動的に `esmin` の値に修正されます。

表 9: パラメータ 4

パラメータ	値	説明
ejamnu	(D=20.)	核子の Bertini モデル (もしくは JQMD) から JAM への切り替えエネルギー (MeV/u)
ejampi	(D=20.)	パイオンの Bertini モデルから JAM への切り替えエネルギー (MeV)
eisobar	(D=0.0)	isobar=1 の時の isobar モデルの上限エネルギー (MeV)
eqmdnu	(D=20.)	核子の Bertini モデルから JQMD への切り替えエネルギー (MeV)
eqmdmin	(D=10.0)	JQMD 適用の下限エネルギー (MeV/u)
ejamqmd	(D=3500.0)	原子核反応の JQMD から JAMQMD への切り替えエネルギー (MeV/u)
inclg	(D=1)	核反応モデルとして INCL を使用する。
	=0	INCL を使用しない。
	=1	陽子、中性子、パイオン、軽イオン ($d, t, {}^3$He,α) が関与する核反応で INCL を使用する。
	=2	陽子、中性子、パイオンが関与する核反応で INCL を使用する。
einclmin	(D=1.0)	INCL が適用されるの下限のエネルギー (MeV/u)。
einclmax	(D=3000.0)	INCL が適用されるの上限のエネルギー (MeV/u)。
incelf	(D=0)	核子が関与する核反応モデルとして INC-ELF を使用する。
	=0	INC-ELF を使用しない。
	=1	INC-ELF を使用する。
eielfmin	(D=1.0)	INC-ELF が適用されるの下限のエネルギー (MeV)。
eielfmax	(D=3500.0)	INC-ELF が適用されるの上限のエネルギー (MeV)。
irqmd	(D=0)	核反応モデルとして JQMD か JQMD-2.0 を使用する。
	=0	JQMD を使用する。
	=1	JQMD-2.0 を使用する。
Kerma	(D=0)	Kerma 計算に関するオプション。
	=0	このオプションを使用しない。
	=1	中性子や陽子の核データを用いた計算において、陽子以外の荷電粒子の生成を無視します。

d, t, α 原子核を輸送している場合、*eqmdmin* 以下のエネルギーでは JQMD による原子核反応をさせません。低エネルギーでの JQMD の適用には限界がありますし、通常の物質中ですと低エネルギーでは飛程が短いので核反応を考慮しなくても影響は小さいです。

高エネルギーの原子核反応は、デフォルトで 3.5GeV/u で、JQMD から JAMQMD モデルへと切り替わります。この切り替えエネルギーを *ejamqmd* で変えられます。核子入射反応でも、*eqmdnu, ejamnu,* とこの *ejamqmd* の値を調整すれば、JAMQMD モデルで計算することも可能です。

INCL (Intra-Nuclear Cascade of Liège) は、核子、パイオン、軽イオン入射反応を記述する核反応模型です。

バージョン 2.50 から、明示的に模型の選択を行わない場合、これらの入射粒子の反応に対しては基本的にこの模型が使用されるようになりました。もし、本模型を用いて得られた結果を発表する場合は、必ず下記文献[9]を引用してください。

INC-ELF(Intra-Nuclear Cascade with Emission of Light Fragment) は軽イオン生成機構をもつ核反応模型です。核子 (陽子、中性子) が入射反応の時に使用できます。本模型を用いて得られた結果を発表する際は、必ず下記文献[10] を引用するようにしてください。

[9] A. Boudard, J. Cugnon, J.-C. David, S. Leray, and D. Mancusi, Phys. Rev C87, 014606 (2013).
[10] Y. Sawada, Y. Uozumi, S. Nogamine, T. Yamada, Y. Iwamoto, T. Sato, and K. Niita, Nucl. Instr. & Meth. B 291, 38-44 (2012).

JQMD, JQMD-2.0 は核反応を記述する模型で、特に重イオン入射反応を記述するのに優れています。バージョン 2.7 以降は、それまで使われていた JQMD に加えて、JQMD-2.0 も選択できるようになりました。JQMD-2.0 [11] は JQMD を改良し、反応の記述をより合理的にした模型で、周辺衝突反応の扱いが特に改善されています。ただし核反応の計算にかかる時間が 2 倍程度長くなります。

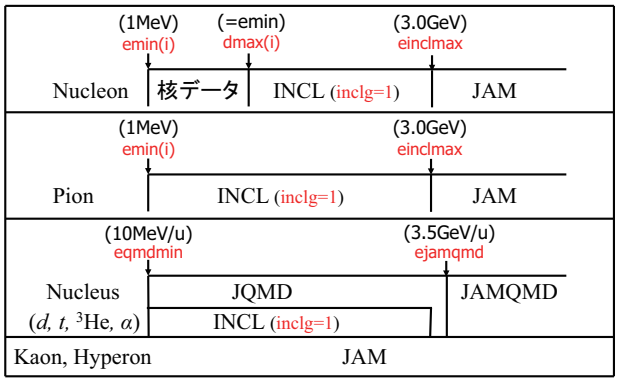


図 6: 核反応モデルの切り替えエネルギー

[11] T. Ogawa, T. Sato, S. Hashimoto, D. Satoh, S. Tsuda, and K. Niita, Phys. Rev C92, 024614 (2015).

4.2.4 時間カット、ウエイトカット、ウエイトウインドウ

表 10: パラメータ 5

パラメータ	値	説明
tmax(i)	(D=1.e+9) i = 1-20	i-th 粒子の計算打切時間 (nsec) 粒子番号は、表を参照
wc1(i) wc2(i) swtm(i)	(D=-0.5) (D=wc1/2) (D=1.0)	i-th 粒子の minimum weight i-th 粒子の cutoff weight i-th 粒子の minimum source weight
wupn	(D=5)	ウェイトウィンドウの上限値 = [Weight Window] セクションで指定した下限値 × wupn wupn ≥ 2
wsurvn	(0.6*wupn)	サバイバルウエイト値 1 < wsurvn < wupn
mxspln	(D=5)	スプリット数の最大値、サバイバルの最大倍数 mxspln > 1
mwhere	(D=0)	ウエイトウインドウのアクションの場所 -1: 核反応時、 0: 両方、 1: 境界横断時

時間のカットオフは粒子毎に tmax(i)= で指定します。単位は nsec です。カットオフ時間を上回った粒子は輸送を終了し、殺されます。高エネルギーの場合ほとんど意味がありませんが、低エネルギーの輸送で利用できます。

ウエイトカットオフは、インポータンス、強制衝突、implicit capture、ウエイトウインドウを利用し、粒子のウエイトが変化する場合に、指定のウエイトカットオフ値になった粒子にロシアンルーレットを施すものです。ウエイトウインドウを指定した粒子については作用しません。

粒子のウエイトが、$WC2$ とソースの発生場所のインポータンスと現在のインポータンスの比 R の積、$WC2 \times R$ より小さくなった時、現在の粒子ウエイト WGT の関数、$WGT/(WC1 \times R)$ の確率で存続させ、その時粒子のウエイトを $WGT = WC1 \times R$ とします。それ以外は、その粒子を殺します。$WC1, WC2$ の入力が負の数で与えられた場合は、$|WC1| \times SWTM$ また、$|WC2| \times SWTM$ が $WC1, WC2$ として設定されます。インポータンスが指定されていない粒子や、領域は、インポータンスは全て 1 にセットされます。

ウエイトウインドウのパラメータについては、通常はデフォルト値で省略できます。

4.2.5 計算モデルオプション (1)

表 11: パラメータ 6

パラメータ	値	説明
ielas	(D=2)	弾性散乱オプション
	= 0	弾性散乱を考慮しない
	= 1	中性子の弾性散乱を考慮する
	= 2	中性子と陽子の弾性散乱を考慮する
ielms	(D=100)	弾性散乱角度分布分点数
inmed	(D=1)	Bertini モデルの核子核子断面積オプション
	= 0	free (nmtclk25.dat)
	= 1	Cugnon old (nmtclk95.dat)
	= 2	Cugnon new (nmtclk30.dat)
nevap	(D=3)	蒸発モデルのオプション
	= 0	蒸発モデルを用いない
	= 1	DRES モデルを用いる
	= 2	SDM モデルを用いる
	= 3	GEM モデルを用いる
ismm	(D=0)	統計マルチフラグメンテーションモデル (SMM) のオプション。
	= 0	統計マルチフラグメンテーションモデルを用いない。
	= 1	統計マルチフラグメンテーションモデルを用いる。JQMD を使う場合、JQMD から GEM への切り替え時間がデフォルトの 100fm/c から 75fm/c に変わります。
igamma	(D=0)	残留核の γ 崩壊オプション
	= 0	γ 崩壊を考慮しない
	= 1	γ 崩壊を考慮する。
	= 2	γ 崩壊を EBITEM モデルを用いて考慮する。
	= 3	γ 崩壊とアイソマー生成を EBITEM モデルを用いて考慮する。バージョン 2.73 以前では、1-3 の場合に file(14) = trxcrd.dat を [parameters] セクションで設定する必要があります。
isobar	(D=0)	isobar モデルのオプション
	= 0	isobar モデルを用いない
	= 1	isobar モデルを用いる

inmed オプションは、inmed=1 の inmedium 断面積がデフォルトになっています。

統計マルチフラグメンテーションモデルは、鉛や水銀などの重い核、または 100MeV/u 前後での核反応で核種生成の再現性を向上させます。ただし核反応の計算にかかる時間が場合によっては数倍長くなります。本モデルに関する詳細は文献[12] [13] をご覧ください。

EBITEM モデルを用いてアイソマー生成を考慮すると、[t-yield] において axis=chart, dchain とした場合に、生成されたアイソマーの情報が出力されます。本モデルに関する詳細は文献[14] をご覧ください。

[12] T. Ogawa, T. Sato, S. Hashimoto, and K. Niita, Nuclear Instruments and Methods in Physics Research A 723 (2013) 36-46.
[13] J.P. Bondorf, A.S. Botvina, A.S. Iljinov, I.N. Mishustin, and K. Sneppen, Physics Reports 257 (1995) 133-221.
[14] T. Ogawa, S. Hashimoto, T. Sato, and K. Niita, Nuclear Instruments and Methods in Physics Research B 325 (2014) 35-42.

4.2.6 計算モデルオプション (2)

表 12: パラメータ 7

パラメータ	値	説明
ipreeq	(D=0) = 0 = 1	nevap=1 の時の preequ モデルのオプション preequ モデルを用いない preequ モデルを用いる
level	(D=3) = 1 = 2 = 3	nevap=1 の時の 状態密度のオプション 8/A 馬場のパラメータ Ignatyuk のパラメータ
ieleh	(D=0) = 0 = 1	電子、陽電子の輸送オプション dmax(12) 以上のエネルギーでは、減速、反応を行わない。 dmax(12) 以上で e=dmax(12), weight=e/dmax(12) とする。
ipnint	(D=0) = 0 = 1 = 2	光核反応のオプション。 (バージョン 2.30 以前、ipngdr というパラメータ名であったもの。) 光核反応を考慮しない。 光核反応を考慮する。ただし核共鳴蛍光散乱を除く。 核共鳴蛍光散乱を含む全ての光核反応を考慮する。 （偏光については欄外参照。）
pnimul	(D=1.0)	光核反応のサンプリング確率に乗じる倍率。
imucap	(D=1) = 0 = 1	負ミューオンが止まったときのミューオン捕獲反応に関するオプション。 ミューオニックアトム生成後の特性 X 線の放出と核吸収反応を考慮する。 考慮しない。 考慮する。
imuint	(D=1) = 0 = 1	仮想光子を介して高エネルギーミューオンが引き起こす核反応に関する オプション。 考慮しない。 考慮する。
imubrm	(D=1) = 0 = 1	ミューオンによる制動放射に関するオプション。 考慮しない。 考慮する。
imuppd	(D=1) = 0 = 1	ミューオンによる電子・陽電子対生成に関するオプション。 考慮しない。 考慮する。

ipnint=2 とした際に考慮される核共鳴蛍光散乱については、入射光子が偏極している場合、[source] セクションで sx,sy,sz を用いて偏光方向を指定できます。その偏極方向を考慮して、散乱光子の角度分布が決定されます。

4.2.7 計算モデルオプション (3)

表 13: パラメータ 8

パラメータ	値	説明
emumin	(D=200.0)	仮想光子を介して核反応を引き起こすミューオンの最低エネルギー (MeV)。imuint = 1 の場合のみ有効。
emumax	(D=1.0e+6)	仮想光子を介して核反応を引き起こすミューオンの最大エネルギー (MeV)。imuint = 1 の場合のみ有効。
npidk	(D=0) = 0 = 1	計算打切エネルギーに達した負電荷の崩壊粒子の取り扱い 吸収反応を強制的に考慮する 崩壊させる
imagnf	(D=0) = 0 = 1	磁場のオプション 磁場を考慮しない 磁場を考慮する
ielctf	(D=0) = 0 = 1	電磁混合場のオプション 電磁混合場を考慮しない 電磁混合場を考慮する
andit	(D=0) = 0 = 1 = 2	Bertini モデルの Δ 角分布オプション 50%等方、50%前方 全て等方分布 全て前方分布
icxsni	(D=0) = 0 = 1 = 2	核子・原子核間の反応断面積、弾性散乱、全断面積オプション Pearlstein-Niita の式 KUROTAMA 模型 Sato の式
icrhi	(D=1) = 0 = 1 = 2	原子核・原子核反応の全断面積オプション Shen の式 NASA の式 KUROTAMA 模型
icrdm	(D=0) = 0 = 1	重陽子の全反応断面積オプション icrhi と同じ MWO の式

emumin を変更した場合、ミューオン核反応断面積が全エネルギー領域で変化しますのでご注意ください。

崩壊チャンネルを持つ粒子が、計算打切エネルギーに達した時には、全て崩壊させます。その中で負電荷を持つ粒子は、npidk=0 の場合は、まず強制的に核反応を試み吸収させます。それでも吸収されなかった場合は、やはり崩壊させます。

KUROTAMA 模型は、幅広いエネルギー範囲において様々な原子核間同士の反応断面積を再現する最新の模型です。詳細はこの文献[15] をご覧ください。また、本模型を使用して得られた結果を使用して論文発表等を行う場合は、必ず該当文献をご引用ください。

MWO の式は、重陽子の全反応断面積を精度良く記述する模型です。詳細はこの文献[16] をご覧ください。また、本模型を使用して得られた結果を使用して論文発表等を行う場合は、必ず該当文献をご引用ください。

[15] K. Iida, A. Kohama, and K. Oyamatsu, J. Phys. Soc. Japan 76, 044201 (2007), and L. Sihver *et al.*, Nucl. Instr. & Meth. B 334, 34-39 (2014).

[16] K. Minomo, K. Washiyama, and K. Ogata, J. Nucl. Sci. Tech. DOI:10.1080/00223131.2016.1213672

4.2.8 計算モデルオプション (4)

表 14: パラメータ 9

パラメータ	値	説明
icxspi	(D=1)	パイオンの全反応断面積オプション
	= 0	幾何学的な断面積
	= 1	Hashimoto の式
iidfs	(D=0)	中性子入射核分裂反応のオプション
	= 1	文献値に基づき中性子多重度 ν とエネルギー分布を再現する
idwba	(D=0)	DWBA スペクトルのオプション
	= 0	DWBA 計算で求めた離散スペクトルを考慮しない
	= 1	DWBA 計算で求めた離散スペクトルを考慮する
gravx	(D=0)	重力場の方向ベクトルの x 成分
gravy	(D=0)	重力場の方向ベクトルの y 成分
gravz	(D=0)	重力場の方向ベクトルの z 成分
usrmgt	(D=1)	[magnetic field] で time を指定した時のユーザーサブルーティンオプション
	= 1	usrmgt1.f を使用、現在、Wobbler magnet がインストール
	= 2	usrmgt2.f を使用、現在、中性子用パルスマグネットがインストール
usrelst	(D=1)	[elastic option] でのユーザーサブルーティンオプション
	= 1	usrelst1.f を使用、現在、中性子用 Bragg 散乱用プログラム
	= 2	usrelst2.f を使用、現在、中性子用サンプルプログラム

バージョン 2.86 より、Hashimoto の式がパイオンの全反応断面積の模型としてデフォルトで使用されます。この式は、従来の幾何学的な断面積と比べて、入射エネルギー依存性を考慮しており、デルタ共鳴によるピーク構造を再現します。

`iidfs=1` を指定した場合、下に示す 18 核種の核分裂で放出される中性子の多重度 ν とエネルギー分布を文献[17] にあるデータに基づき再現します。

U-238, Pu-238, Pu-240, Pu-242, Cm-242, Cm-244, Cf-252, Th-232, U-232, U-233, U-234, U-235, U-236, Np-237, Pu-239, Pu-241, Am-241, Bk-249.

`idwba` は、特定の標的原子核における陽子、重陽子入射反応を対象として、放出中性子や陽子のエネルギースペクトルに、DWBA（歪曲波ボルン近似）によって計算した離散スペクトルを追加するオプションです。具体的には、以下の入射エネルギーと反応が対象となります。

30–400 MeV における ^{7}Li$(p,n)^{7}$Be 反応
10–50 MeV における ^{9}Be$(p,n)^{9}$B 反応
10–50 MeV における ^{6,7}Li$(d,n)^{7,8}$Be および ^{6,7}Li$(d,p)^{7,8}$Li 反応
5–25 MeV における ^{9}Be$(d,n)^{10}$B および ^{9}Be$(d,p)^{10}$Be 反応
10–50 MeV における ^{12,13}C$(d,n)^{13,14}$N および ^{12,13}C$(d,p)^{13,14}$C 反応

`gravx, gravy, gravz` は、重力場の方向ベクトル。重力を設定したときは、1eV 以下の中性子に作用します。`gravx=1, gravy=0, gravz=0` を指定したときは、x 負方向に重力が作用します。

[17] J. M. Verbeke, C. Hagmnn, and D. Wright, “Simulation of Neutron and Gamma Ray Emission from Fission and Photofission”, UCRL-AR-228518 (2014).

4.2.9 計算モデルオプション (5)

表 15: パラメータ 10

パラメータ	値	説明
ndedx	(D=3)	荷電粒子の dE/dx オプション
	= 0	重イオンは SPAR[18]、他は NMTC オリジナル
	= 1	重イオン、陽子は ATIMA[19]、他は NMTC オリジナル
	= 2	重イオン、陽子、パイオン、ミューオンは SPAR、他は NMTC オリジナル
	= 3	全ての荷電粒子に対して ATIMA を利用する
mdbatima	(D=500)	ATIMA 用データベースの最大値
dbcutoff	(D=0.0)	ATIMA 用データベース作成の下限エネルギー (MeV/u)
ih2o	(D=-1)	ATIMA の時の水 (H2O のみ) の Ionization Potential オプション
	< 0	デフォルト値、75 eV
	> 0	水の Ionization Potential (eV)
nspred	(D=0)	荷電粒子のクーロン散乱 (angle straggling) オプション
	= 0	クーロン散乱を考慮しない
	= 1	クーロン散乱を考慮する。NMTC オリジナル
	= 2	（推奨）クーロン散乱を考慮する。Moliere 理論に基づく Lynch の式[20]
	= 10	クーロン散乱を考慮する。ATIMA の計算式
ascat1	(D=13.6)	Lynch の式の S_2 パラメータ。nspred = 2 の場合のみ有効
ascat2	(D=0.088)	Lynch の式の ϵ パラメータ。nspred = 2 の場合のみ有効
nedisp	(D=0)	荷電粒子の減速過程でのエネルギー分散 (energy straggling) のオプション
	= 0	エネルギー分散を考慮しない
	= 1	（推奨）エネルギー分散を考慮する。Landau Vavilov[21]
	= 10	エネルギー分散を考慮する。ATIMA の時
e-mode	(D=0)	Event generator mode オプション。
	= 0	通常モード。
	= 1	Event generator mode Ver.1.
	= 2	Event generator mode Ver.2.
		バージョン 2.73 以前では、1,2 の場合に file(14) = trxcrd.dat を [parameters] セクションで設定する必要があります。

ndedx は、バージョン 2.00 以前は ndedx=0 を、それより後からバージョン 2.85 までは ndedx=2 をデフォルトとしておりましたので、注意してください。

バージョン 2.85 より、阻止能計算モデル ATIMA のアルゴリズムを改良し、高速化しました。従来のアルゴリズムでは、輸送計算の途中で新たな粒子が発生したり、別の物質に到達したりする度に、毎回 ATIMA による阻止能計算を実行し、粒子の飛程を求めていました。そこで、計算された粒子・物質の組みに対する情報をデータベースとして記憶しておき、次回の同じ組の呼び出しではデータベースを参照することで、計算負荷の高い ATIMA の呼び出し回数を減らすように変更しました。この改良により、精度の高い ATIMA を用いた PHITS シミュレーションが、SPAR を用いた場合とほぼ同程度の計算時間で実行できるようになりました。

e-mode オプションの詳細は、4.2.22 をご覧ください。

[18] T.M. Armstrong and K.C. Chandler, ORNL Report, ORNL-4869 (1973).
[19] ATIMA モデルについては、次のウェブサイトをご覧ください。https://web-docs.gsi.de/ weick/atima/
[20] $\sigma = S_2 \frac{\sqrt{X/X_0}}{p\beta}[1 + \epsilon \log_{10}(X/X_0)]$: G.R. Lynch and O.I. Dahl, Nucl. Instrum. Methods Phys. Res, B 58, 6-10 (1991) の式 (4).
[21] 本モデルについては、次のウェブサイトをご覧ください。http://www.dnp.fmph.uniba.sk/cernlib/asdoc/geantold/H2GEANTPHYS332.html

4.2.10 出力オプション (1)

表 16: パラメータ 11

パラメータ	値	説明
infout	(D=7)	file(6) に出力する情報を指定するオプション 次の I, II, III, IV, V, VI については表外の説明を参照
	= 0	I
	= 1	I, II
	= 2	I, III
	= 3	I, IV
	= 4	I, II, III
	= 5	I, II, IV
	= 6	I, III, IV
	= 7	I, II, III, IV, VI
	= 8	I, II, III, IV, V, VI
nrecover	(D=0)	Lost particle が発生して recover に成功した場合に、 warning message を出力する回数。

パラメーター infout は、*PHITS* のサマリーファイルに出力する情報を指定するオプションです。情報は次の 6 カテゴリに分類されており、infout によって出力させるカテゴリを指定できます。

I 基本情報
ロゴ（PHITS 著者の情報を除く）、計算プロセス (ncol) 情報、発生線源情報、幾何形状エラー情報、乱数情報、計算時間情報

II インプットエコー

III メモリー情報、バッチ情報

IV 粒子情報

V 詳細情報（分散低減情報、領域別散乱情報、物質別散乱情報）

VI PHITS 著者の情報

これらの情報は、下記の順番で書き出されます。

(1) ロゴ（カテゴリ I. ただし、著者情報はカテゴリ VI）
PHITS のバージョン、PHITS 著者、JOB タイトル、スタート時間

(2) インプットエコー（カテゴリ II）
ユーザーのインプットファイルのエコー

(3) メモリー情報（カテゴリ III）
どのパートにどのくらいのメモリーが使われているかの情報

(4) バッチ情報（カテゴリ III）
バッチ毎のヒストリー数、計算時間等の情報

(5) 計算プロセス (ncol) 情報（カテゴリ I）
領域横断、散乱、カットオフ等、輸送のプロセス別の回数

(6) 分散低減情報（カテゴリ V）
ウエイトを変化させる importance, weight window, forced collision 等における情報

(7) 領域別散乱情報（カテゴリ V）
領域ごとの散乱の情報

(8) 物質別散乱情報（カテゴリ V）
物質ごとの散乱の情報

(9) 粒子情報（カテゴリ IV）
輸送粒子、生成粒子、崩壊粒子、カットオフ粒子

(10) 発生線源情報（カテゴリ I）
発生粒子数とそのウエイト

(11) 幾何形状エラー情報（カテゴリ I）
ロスト粒子の数と情報

(12) 乱数情報（カテゴリ I）
初期乱数と次の計算の初期乱数

(13) 計算時間情報（カテゴリ I）
総計算時間と各プロセスのコール回数に関する情報
各項目の意味は以下の通りです。（ただし、`icput=0` の場合は total cpu time 以外の計算時間が出力されません。またその場合、transport と set data の項目自体出力されません。）

- total cpu time: 総計算時間
- transport: 総計算時間から以下の 5 項目に要した時間を除いた時間
- set data: 入力パラメータなどのデータ設定に要した時間
- analysis: 何らかのデータ処理に要した時間および実行回数
- nevap: 蒸発モデルによる蒸発過程の計算に要した時間および実行回数
- dexgam: 原子核の脱励起の計算に要した時間および実行回数
- nreac: 核反応モデルによる直接過程・前平衡過程計算およびライブラリーによる反応計算に要した時間および実行回数
- dklos: 粒子の崩壊の計算回数
- hydro: 標的が 1H, 入射粒子が原子核で無い場合の計算回数
- n-data: 中性子ライブラリーによる計算回数
- h-data: 陽子ライブラリーによる計算回数
- p-data: 光子ライブラリーによる計算回数
- e-data: 電子ライブラリーによる計算回数
- p-egs5: EGS5 を用いた光子の計算回数
- e-egs5: EGS5 を用いた電子の計算回数
- photonucl: 光核反応の計算回数

- muon: ミューオン核反応の計算回数
- elast: 弾性散乱の計算回数
- ncasc: 核反応モデルによる直接過程・前平衡過程の計算回数
- bertini: Bertini モデルの計算回数
- isobar: isobar モデルの計算回数
- JAM: JAM モデルの計算回数
- QMD: JQMD モデル、JQMD-2.0 の計算回数
- JAMQMD: JAMQMD モデルの計算回数
- INCL: INCL モデルの計算回数
- INCELF: INC-ELF モデルの計算回数

また、`infout=8` とした場合は核反応モデル毎のサマリーが出力され、各入射粒子の反応が何回モデル計算されたか (count) とその計算が成功した回数 (real) 及びその成功率 (%) が書きだされます。

4.2.11 出力オプション (2)

表 17: パラメータ 12

パラメータ	値	説明
incut	(D=0)	計算打切エネルギー以下の中性子情報の出力オプション
	= 0	出力なし
	= 1	ファイル名 file(12) の ncut ファイルに出力
	= 2	file(12) に時間の情報も出力
igcut	(D=0)	計算打切エネルギー以下の γ 線、電子情報の出力オプション
	= 0	出力なし
	= 1	γ 線情報をファイル名 file(13) の gcut ファイルに出力
	= 2	file(13) に γ 線の時間の情報も出力
	= 3	file(13) に γ 線、電子、陽電子の情報を出力
ipcut	(D=0)	計算打切エネルギー以下の陽子情報の出力オプション
	= 0	出力なし
	= 1	ファイル名 file(10) の pcut ファイルに出力
	= 2	file(10) に時間の情報も出力
inpara	(D=0)	メモリ分散型並列計算の時、ncut ファイル名のオプション
	= 0	/wk/uname/ファイル名 file(12)
	= 1	/wk/uname/ファイル名 file(12) の後に PE 番号を付加
	= 3	ファイル名 file(12) の後に PE 番号を付加
igpara	(D=0)	メモリ分散型並列計算の時、gcut ファイル名のオプション
	= 0	/wk/uname/ファイル名 file(13)
	= 1	/wk/uname/ファイル名 file(13) の後に PE 番号を付加
	= 3	ファイル名 file(13) の後に PE 番号を付加
ippara	(D=0)	メモリ分散型並列計算の時、pcut ファイル名のオプション
	= 0	/wk/uname/ファイル名 file(10) そのまま
	= 1	/wk/uname/ファイル名 file(10) の後に PE 番号を付加
	= 3	ファイル名 file(10) の後に PE 番号を付加
	/wk/uname/	/wk/ はデフォルトのディレクトリ名 uname は環境変数 LOGNAME から読み込まれるユーザー名

incut, igcut のデフォルトをゼロに変えましたので注意してください。

メモリ分散型並列計算時は PE (Processor Element) 毎に別のファイルを用意して書き出します。inpara, igpara, ippara=0, 1 の場合は、各 PE がもっているハードディスク中の/wk/uname/のディレクトリにファイルを作成します。inpara, igpara, ippara =1, 3 の場合は、ファイル名の最後に各 PE 番号を付けて総 PE 数 −1 個のファイルを作成します。各 PE は対応するファイルにのみその PE で得られたデータを書き出します。

4.2.12 出力オプション (3)

表 18: パラメータ 13

パラメータ	値	説明
itall	(D=0)	タリーの出力をバッチ毎にするオプション
	= 0	出力なし
	= 1	出力あり
itstep	(D=0)	magnetic field などの粒子の運動量が変化する場でのタリーを取るオプション
	= 0	境界、核反応までタリーを取らない
	= 1	ステップごとにタリーを取る
imout	(D=0)	[material] セクションでの物質表示オプション
	= 0	mat[12], 208Pb.33c タイプ
	= 1	mat[12], Pb-208.33c タイプ
	= 2	m12, 82208.33c MCNP 入力タイプ
jmout	(D=0)	[material] セクションでの物質密度表示オプション
	= 0	入力のまま
	= 1	粒子密度に変換する
	= 2	質量比 (%) に変換する
kmout	(D=0)	核データ物質情報表示オプション
	= 0	表示無し
	= 1	input echo に表示する
		Version 2.86 より、核種毎の dmax も出力されるようになりました。
matadd	(D=1)	[cell] セクションにおいて同一の物質番号を用いて密度が異なる領域を定義した場合のオプション。
	= 0	同じ物質番号で扱う。
	= 1	新しい物質番号を付与する。
natural	(D=1)	[material] セクションにおいて質量数を指定せずに書いた元素に対し、天然同位体比による展開を行うかどうかのオプション。
	= 0	展開しない。
	= 1	展開する。ただし、file(6) には展開せずに書き出す。
	= 2	展開する。ただし、file(6) に展開した結果を書き出す。

磁場など粒子の運動量を変化させる場のある時、タリーは通常境界面を横断するときと、核反応を起こすタイミングでしか呼ばれないので、磁場中の軌跡を track タリーで表示すると境界間は直線でしか表現できませんでした。そこで、バーチャルな境界面などを設定して表現していたのですが、この itstep のオプションを用いることにより、運動の各ステップごとにタリーを呼び、曲線の軌跡を追うことができるようになりました。磁場のある場合のステップの最大値は、deltm で指定します。

[cell] セクションにおいて同一の物質番号で密度の違う領域を複数定義した場合、デフォルト設定 (matadd=1) では、最初のもの以外の密度の物質に対して新しい物質番号を付与します。matadd=0 とすることで同じ物質番号を指定できるようになりますが、[Mat Name Color] セクションで指定する物質番号には影響がないのでご注意ください。すなわち、このセクションでは自動的に付与された新しい番号を指定する必要があります。この番号は file(6) (D=phits.out) の最初に Warning として出力されますので、その値をご参照ください。

4.2.13 出力オプション (4)

表 19: パラメータ 14

パラメータ	値	説明
iggcm	(D=0) = 0 = 1	GG の warning 等のメッセージをエコーする エコーしない インプットエコー部分に書き出す
ivout	(D=0) = 0	input echo での volume 表示のオプション [volume] セクションに書き出す
ipout	(D=1) (D=0 for GG) = 0	input echo での importance 表示のオプション [importance] セクションに書き出す 但し、全粒子に同じ値が定義されているときに限る
icput	(D=0) = 0 = 1	CPU 時間のカウントオプション カウント無し カウント有り
ipara	(D=0) = 0 = 1	パラメータ表示オプション 入力のみの表示 全てを表示
nwsors	(D=0)	nwsors 個のソース情報を file(6) に書き出す
dumpall	(D=0) = 0 = 1 = -1	全ての情報のダンプオプション 出力なし ファイル名 file(15) の dumpall ファイルにバイナリ形式でデータを出力 ファイル名 file(15) の dumpall ファイルにアスキー形式でデータを出力
idpara	(D=3) = 0 = 1 = 3	メモリ分散型並列計算の時、dumpall ファイル名のオプション /wk/uname/ファイル名 file(15) /wk/uname/ファイル名 file(15) の後に PE 番号を付加 ファイル名 file(15) の後に PE 番号を付加
	/wk/uname/	/wk/ はデフォルトのディレクトリ名 uname は環境変数 LOGNAME から読み込まれるユーザー名

計算時間短縮のため、cpu 時間のカウントをデフォルトで無しにしました。各過程の計算時間を知りたい時は、icput=1 として下さい。

input echo の入力パラメータの表示は、デフォルトで入力のあったものだけですが、ipara=1 とすれば全てのパラメータのデフォルト値がエコーされます。

dumpall=1, -1 を指定すると、輸送に関して全ての情報をファイルに書き出します。このファイルを基に、icntl=12 を指定して再計算ができます。最初に計算したときと異なるタリーを入れて再計算したいときなど有効です。ただし、dumpall の機能を利用するときは、巨大なデータファイルが作成されることがありますから注意してください。この dumpall オプションは、GG のときのみ有効です。実際にダンプする情報は、4.2.21 のところで解説します。

メモリ分散型並列計算時は PE (Processer Element) 毎に別のファイルを用意して書き出し・読み込みを行います。idpara=0, 1 の場合は、各 PE がもっているハードディスク中の/wk/uname/のディレクトリにファイルを作成します。idpara=1, 3 の場合は、ファイル名の最後に各 PE 番号を付けて総 PE 数 −1 個のファイルを作成します。各 PE は対応するファイルにのみデータを書き出し、そのファイルからのみデータの読み込みを行います。

4.2.14 出力オプション (5)

表 20: パラメータ 15

パラメータ	値	説明
ivoxel	(D=0)	ボクセルデータのバイナリ形式による出力・読み込み。
	=0	出力なし。
	=1	ファイル名 file(18) からバイナリ形式のボクセルデータを読み込む。
	=2	ファイル名 file(18) にバイナリ形式のボクセルデータを出力する。
itetvol	(D=0)	連続四面体形状利用時 (LAT=3) における体積の自動計算。
	=0	自動で計算しない。
	=1	自動で計算しタリーなどで体積の値として使用する。
itetra	(D=0)	連続四面体データのバイナリ形式による出力・読み込み。
	=0	出力なし。
	=1	ファイル Tetra.bin（固定）からバイナリ形式の連続四面体データを読み込む。
	=2	ファイル Tetra.bin（固定）にバイナリ形式の連続四面体データを出力する。
ntetsurf	(D=100)	連続四面体計算のメッシュ分割の際に、一つの領域に許容する連続四面体の外部面の数。
ntetelem	(D=200)	連続四面体計算のメッシュ分割の際に、一つの領域に許容する四面体の数。

ivoxel=2 を指定すると、ボクセルデータをバイナリ形式でファイル名 file(18) に出力して計算を終了します (バージョン 2.30 以前では、終了せずに引き続き計算を行います)。その後、このデータファイルがある状態で ivoxel=1 として再度 *PHITS* を実行させると、通常一時的に行うバイナリデータの作成を省略して、以後の計算を行います。*PHITS* では、読み取ったインプットデータを一時的にバイナリ形式で書き出し、それを読み込んでから粒子輸送計算を行います。したがって、読み込むデータ量が大きくなると、それに比例して書き出しの際の計算時間がかかることになります。ivoxel を利用すると、バイナリデータの書き出しを省略するため、何度も *PHITS* を実行する場合に計算時間が短縮されます。

itetvol=1 を指定することで、連続四面体形状 (LAT=3) を使用している場合に、連続四面体形状内に現れる Universe の体積を自動で計算して使用します。ただし、単純に四面体の体積を足し上げるため、Universe と Fill の入れ子構造で、連続四面体形状の一部を切り出して使用している場合などは、正しい体積の値とならないので、注意が必要です。itetvol=1 が指定されている場合は上記の計算値が Volume セクション等での指定よりも優先されます。

itetra=2 を指定すると、連続四面体形状を読み込んで輸送計算に必要な情報データを生成した後に、その情報をバイナリ形式のデータファイル Tetra.bin （ファイル名固定）に出力して計算を終了します。その後、このデータファイルがある状態で itetra=1 として再度 PHITS を実行させると、データ処理を省略して必要情報をファイルから読み込みます。四面体数の多い連続四面体データを使用する際には、データ処理に計算時間がかかるため、同じ連続四面体形状を使用して何度も PHITS を実行する場合には、計算時間の短縮が図れます。

連続四面体形状を使用する計算では、計算スピードを向上させるため、形状を収めた直方体を自動で細分化し、個々の領域に入る四面体の数を減らす二種類のアルゴリズムが組み込まれています。一つは、連続四面体の外側から連続四面体内へと粒子が入射される場合にどの四面体の面を通過するかの判定に使用するアルゴリズムで、領域を細分化して一つの領域に含まれる外部面の数を ntetsurf 以下に限ることで、輸送計算を高速化します。もう一つは、連続四面体内に線源が存在する場合に、放射線が発生する位置がどの四面体に属するかの判断に使用するアルゴリズムで、領域を細分化して、一つの領域に含まれる四面体の数を ntetelem 以下に限ることで、輸送計算を高速化します。

4.2.15 幾何形状のエラー関係

表 21: パラメータ 17

パラメータ	値	説明
nlost	(D=10)	lost particle の許容値 (1PE あたり)
igerr	(D=1)	region error 時の recovery 回数
igchk	(D=0)	=0: 領域横断後の幾何形状チェックを行いません。 =1: 領域横断後の flight mesh を deltb に設定し、 幾何形状のチェックを行います。
deltb	(D=1.e-5)	領域横断後の flight mesh (cm) igchk=1 で、この距離で幾何形状のチェックを行います。 また、importance, forced collision で追加される粒子の 境界面からの距離。また、source で粒子が境界上にある ときの移動距離としても使います。
deltm	(D=20.12345)	最大飛行メッシュ(cm)。 idelt に連動。
deltc	(D=2.012345)	エネルギー分散を考慮した場合 (nedisp=1,10) の荷電粒子の最大飛行 メッシュ(cm)。 idelt に連動。
delt0	(D=0.1)	クーロン多重散乱を考慮した場合 (nspred=1,2,10) や磁場を考慮した場 合 (imagnf=1) の最小飛行メッシュ(cm)。ただし、厚さが 1cm 以下 の薄膜等による散乱計算を行う場合は、散乱を正確に記述するた めにその厚さの 1/10 に設定してください。
deltg	(D=1.0)	磁場における最大飛行メッシュ(cm)。
deltt	(D=1.0)	時間に依存した磁場における最大飛行時間 (msec)。
idelt	(D=1)	最大飛行メッシュに関するオプション。 =0: 最大飛行メッシュとして deltm, deltc を使用。 =1: deltm, deltc を領域の密度で割った値を最大飛行メッシュとする。

PHITS ver.2.80 以降、idelt=1 となっており、deltm, deltc を領域の密度で割った値が最大飛行メッシュとして使われます。これにより、数百 m 以上の空気中の粒子輸送計算を行う場合に、計算時間が短縮されます。

4.2.16 入出力ファイル名

表 22: パラメータ 18

パラメータ	値	説明
file(2)	(D=cgview.in)	icntl=2 の時、EGS4 粒子飛跡表示プログラム CGVIEW 用出力ファイル名
file(3)	(D=cgview.set)	icntl=2 の時、CGVIEW 用セットアップ情報ファイル名 この部分が CGVIEW のセットアップになる
file(4)	(D=marspf.in)	icntl=4 の時、MARS-PF 用出力ファイル名
file(6)	(D=phits.out)	サマリーの出力ファイル名。指定しない時は、標準出力
file(7)	(D=c:/phits/data/xsdir.jnd)	断面積ディレクトリファイル名
file(11)	(D=nuclcal.out)	核反応計算の出力ファイル名
file(12)	(D=fort.12)	計算打切エネルギー以下の中性子情報の出力ファイル名
file(13)	(D=fort.13)	計算打切エネルギー以下の γ 線情報の出力ファイル名
file(10)	(D=fort.10)	計算打切エネルギー以下の陽子情報の出力ファイル名
file(15)	(D=dumpall.dat)	dumpall=1 を指定した時、情報の出力ファイル名
file(18)	(D=voxel.bin)	ivoxel=1, 2 を指定した際のバイナリデータ用ファイル名
file(20)	(D=c:/phits/XS/egs/)	EGS5 用断面積データを格納したフォルダ名。 negs=1 のときに必要。
file(21)	(D=c:/phits/dchain-sp/data/)	DCHAIN-SP 用データを格納したフォルダ名
file(22)	(D=batch.out)	現在のバッチ情報を出力するファイル名
file(23)	(D=pegs5)	PEGS5 用出力ファイル名
file(24)	(D=c:/phits/data)	DECDC データ (RIsource.dat) を格納したフォルダ。
dircha	(D=0)	ディレクトリ文字
	=0	“/ ”
	=1	“backslash”

file(7) は、他と同様にフルパス名で書いてください。

dircha は、Windows での運用を想定したパラメータです。UNIX 系では、ディレクトリ文字が ’/’ ですが、Windows 系では ’\’ なので、その違いを吸収します。ただし、WindowsXP では、’/’ のままでも動いています。

バージョン 2.74 より、igamma≥1 のときでも、γ 崩壊情報データファイル trxcrd.dat を file(14) に指定する必要がなくなりました。

4.2.17 その他

表 23: パラメータ 19

パラメータ	値	説明
inucr	(D=1) = 1 = 2 = 3 = 4 = 5 = 6	核反応計算のオプション 二重微分断面積計算 全、弾性、弾性外断面積出力 シミュレーションによる弾性外断面積計算 弾性散乱の角分布計算 pp, np, π-p, 断面積出力 シミュレーションによる pp, np, π-p, 断面積計算
idam(i) rdam(i)	integer real*8 i = 1 - 100	ユーザー定義の整数変数 ユーザー定義の実数変数 これらは、各プログラムの中で common /userp/ idam(100), rdam(100) を引用すれば用いることができる

icntl=1 の nuclear reaction 計算は、開発が取り残されているため、現在、まだ工事中と考えてください。これらの inucr の機能が現在も利用可能か定かではありません。

4.2.18 低エネルギー中性子の物理パラメータ

以下のパラメータは、20MeV 以下の中性子の計算に用います。

表 24: パラメータ 20

パラメータ	値	説明
emcnf	(D=0.0)	中性子捕獲のしきい値 (MeV) これ以上で implicit capture、以下で analog capture
iunr	(D=0)	現在、0 で固定
dnb	(D=-1) =-1 = 0 > 0	核分裂からの delayed neutron 数 natural sampling no delayed neutron neutron number
nonu	(D=1) =0	通常の fission として取り扱う fission を起こさない。単なる中性子捕獲反応として扱う。 光子が生成される。

nonu=0 とすることで、計算体系内に設定された核分裂性核種に、輸送中の中性子が核分裂反応を誘起した際、2 次中性子による中性子増倍効果を発現させず、即ち、中性子を発生させずに、単に中性子捕獲反応として輸送計算を継続することができます。用途としては、臨界解析コード等により別途評価した同体系における中性子実効増倍率を加味した中性子線源等を用いて、実質的な中性子透過問題の評価が行えます。これは、MCNP コードにおける NONU カード相当に相当します。

4.2.19 光子・電子輸送オリジナルモデルの物理パラメータ

以下のパラメータは、`negs=0` のときのみ有効となります。

表 25: パラメータ 21

パラメータ	値	説明
`emcpf`	`(D=100)`	photon の詳細モデルの上限 (MeV)
`ides`	`(D=0)`	photon の electron 生成のオプション
	`= 0`	electron を生成、もしくは、brems.photon の生成
	`= 1`	electron を生成しない
`nocoh`	`(D=0)`	photon の coherent scattering のオプション
	`= 0`	coherent scattering を起こす
	`= 1`	coherent scattering を起こさない
`iphot`	`(D=0)`	electron の photon 生成のオプション
	`= 0`	photon を生成
	`= 1`	photon を生成しない
`ibad`	`(D=0)`	brems. の角分布のオプション
	`= 0`	full brems. tabular angular distribution
	`= 1`	simple brems. angular distribution approximation
`istrg`	`(D=0)`	straggling のオプション
	`= 0`	sampled straggling for electron energy loss
	`= 1`	expected-value straggling for electron energy loss
`bnum`	`(D=1)`	brems. photon のオプション
	`= 0`	brems. photon を生成しない
	`> 0`	analog brems. photon 数
`xnum`	`(D=1)`	x-ray photon のオプション
	`= 0`	x-ray photon を生成しない
	`> 0`	analog x-ray photon 数
`enum`	`(D=1)`	secondary electron のオプション
	`= 0`	secondary electron を生成しない
	`> 0`	analog secondary electron 数
`numb`	`(D=0)`	brems. process のオプション
	`= 0`	nominal brems. production
	`> 0`	produce brems. on each substep
`rnok`	`(D=1)`	knock-on electron のオプション
	`= 0`	knock-on electron を生成しない
	`> 0`	knock-on electron 数

4.2.20 EGS5 用パラメータ

表 26: パラメータ 22

パラメータ	値	説明
negs	(D=0) = 0 = 1	光子・電子・陽電子の輸送に関するオプション オリジナルモデルを使う（EGS5 を使用しない）。この場合、表 25 の パラメータが有効になります。 EGS5 を使う。この場合、file(20) の指定が必要となります。
ipegs	(D=0) = -1 = 0 = 1 = 2	PEGS5（EGS5 用断面積生成プログラム）コントロールオプション。 pegs5.inp をユーザー自身で変更したいときに使用します。なお，ipegs が 1 以上の場合，iegsout は自動的に 1 となり PEGS5 関連ファイルを残す ようにします。（ただし，iegsout = 2 としている場合は変更しません。） また，negs = 0 のときは無効となります。 PEGS5.inp の作成○，PEGS5 の実行○，PHITS の実行× PEGS5.inp の作成○，PEGS5 の実行○，PHITS の実行○ PEGS5.inp の作成×，PEGS5 の実行○，PHITS の実行○ PEGS5.inp の作成×，PEGS5 の実行×，PHITS の実行○
imsegs	(D=1) = 0 = 1	多重散乱（Multiple Scattering）の扱いに関するオプション。 EGS5 オリジナル手法。ソース発生直後のみ starting scattering strength を 使って多重散乱を詳細に模擬し，その後は，maximum scattering strength を 使用する。 PHITS-EGS5 独自の手法。物質が変わるたびに starting scattering strength を 使用して多重散乱を詳細に模擬する。これにより，多少計算時間が長くなる が，chard パラメータを変更することなく薄膜による散乱をある程度正確に 模擬できるようになる。
iegsout	(D=0) = 0 = 1 = 2	EGS5 用出力ファイルオプション EGS5 用の出力ファイルを残さない。ただし、計算を途中で打ち切った 場合は、ファイルが残る場合があります。 再読込するファイル（pegs5.inp, pegs5.dat, pegs5.msfit）のみ残す 全て残す。各ファイルに出力される情報の詳細は、EGS5 のマニュアル[22]を ご参照ください。

negs=1 の場合、電子・陽電子・光子の輸送エネルギー下限・上限に対するデフォルト値が emin(12,13)=0.1, emin(14)=0.001, dmax(12-14)=1000.0 となります。ただし、これらの値を [parameters] セクションで直接指定した場合は、指定した値が優先されます。また、多くの EGS5 用パラメータのデフォルト値がオリジナル EGS5 の値と異なっていることにご注意ください。

バージョン 2.76 より EGS5 を使用している場合でも光核反応を考慮することができるようになりました。ipnint=1 とした場合は、以前のバージョンと結果が変わりますのでご注意ください。

[22] H. Hirayama *et al.*, SLAC-R-730 (2005) and KEK Report 2005-8 (2005).

表 27: パラメータ 23

パラメータ	値	説明
iegsrand	(D=-1)	EGS5 で使う乱数初期値。0 以上を指定した場合、EGS5 用の乱数を使うため、MPI 並列計算や再開始計算ができなくなることにご注意ください。
	< 0	EGS5 用の乱数は使わない（PHITS で使う乱数をそのまま利用する）
	= 0	EGS5 のデフォルト乱数初期値（314159265）を使う
	> 0	指定した値を EGS5 乱数初期値として使う
iedgfl	(D=1)	K 及び L 殻特性 X 線に関するオプション
	= 0	K 及び L 殻特性 X 線を扱わない
	= 1	K 及び L 殻特性 X 線を扱う
iauger	(D=1)	K 及び L 殻オージェ電子に関するオプション
	= 0	K 及び L 殻オージェ電子を扱わない
	= 1	K 及び L 殻オージェ電子を扱う
iraylr	(D=1)	レイリー散乱（Rayleigh scattering）に関するオプション
	= 0	レイリー散乱を扱わない
	= 1	レイリー散乱を扱う
lpolar	(D=0)	光子散乱での直線偏光（linearly polarized photon scattering）に関するオプション（未対応）
	= 0	直線偏光を扱わない
	= 1	直線偏光を扱う
iunrst	(D=0)	PEGS5 による阻止能出力オプション。ただし，計算結果が変わってしまうので，0 以外の値を指定する場合は ipegs=-1 として PEGS5 のみ実行すること。
	= 1	unrestricted collision only
	= 2	unrestricted collision and radiative
	= 3	unrestricted collision, restricted radiative
	= 4	restricted collision, unrestricted radiative
	= 5	unrestricted radiative only
	= 6	restricted radiative only
	= 7	restricted collision only
chard	(D=0.1)	EGS5 のステップサイズをコントロールするパラメータ。imsegs=0 として薄膜のある体系を計算する場合，その薄膜の厚さ程度に設定する必要があります。オリジナルの EGS5 では物質毎に設定できますが，PHITS では全ての物質に対して同じ値を設定します。物質毎に調整したい場合は，ipegs パラメータを使って手動で pegs5.inp を変更してください。 また，負値を入力した場合は，その絶対値を各物質の密度 (g/cm^3) で割った値を EGS5 で使用する chard パラメータに設定します。密度が薄く巨大な体系（km オーダー）では，負値を設定することにより，計算時間を大幅に短縮できます。
epstfl	(D=0)	EPSTAR（ICRU37 を作成する際に使用したデータベース）に格納された衝突断面積に関する密度補正係数を利用するオプション。
	= 0	EPSTAR を使わない
	= 1	EPSTAR を使用する。この場合，PHITS 入力ファイルと同じフォルダに epstar.dat と名付けたファイルを作成し，[material] セクションで指定した物質の順番に EPSTAR のデータ（XS\density_corrections に格納）をコピー＆ペーストしておく必要があります。

表 28: パラメータ 24

パラメータ	値	説明
incohr	(D=1)	コンプトン散乱角の非干渉散乱関数（incoherent scattering function、S/Z rejection）に関するオプション
	= 0	非干渉散乱関数を使わない（自由電子との散乱）
	= 1	非干渉散乱関数を使う（束縛電子との散乱）（ibound が自動的に 1 に固定されます）
iprofr	(D=1)	コンプトン散乱におけるドップラー効果に関するオプション
	= 0	ドップラー効果を考慮しない
	= 1	ドップラー効果を考慮する（incohr と ibound が自動的に 1 に固定されます）
impacr	(D=1)	EII（Electron impact ionization）に関するオプション
	= 0	EII を考慮しない
	= 1	EII を考慮する
ieispl	(D=0)	EII により発生する光子の扱いに関するオプション
	= 0	分割しない（no splitting）
	= 1	分割する（splitting）
neispl	(D=0)	ieispl=1 のときに分割する光子の数
ibrdst	(D=1)	制動放射の発生角度に関するオプション
	= 0	デフォルト値
	= 1	サンプリング
iprdst	(D=1)	電子対生成の発生角度に関するオプション
	= 0	デフォルト値
	= 1	サンプリング
iphter	(D=1)	光電子の角度分布に関するオプション
	= 0	デフォルト値（入射光子の方向）
	= 1	サンプリング
ibound	(D=1)	コンプトン断面積に関するオプション
	= 0	自由コンプトン断面積 (free Compton cross section)
	= 1	束縛全コンプトン断面積 (bound total Compton cross section)
iaprim	(D=1)	制動放射断面積に関するオプション
	= 0	Motz *et al.* の経験式を使う
	= 1	ICRU-37 で与えられた放射阻止能（radiative stopping power) に規格化する
	= 2	規格化しない

4.2.21 Dumpall オプション

`dumpall=1, -1` を指定して計算すると、再計算に必要な輸送の情報を全てファイルに書き出します。このファイルを用いて `icntl=12` を指定すると、同じ `dumpall` の値で計算した時にダンプしたデータを基に再計算します。輸送に関しては、最初に計算した全ての情報を再現します。再計算の時には、最初に計算したインプットファイルが必要になります。タリーを除いて同じものを使わないと矛盾が生じます。maxcas, maxbch は、ダンプファイルに指定されているものを使いますので変更できません。最初に計算したときと異なるタリーを入れて再計算したいときなど有効です。ただし、`dumpall` の機能を利用するときは、巨大なデータファイルが作成されることがありますから注意してください。この dumpall オプションは、GG のときのみ有効です。

バイナリ形式で書き出したデータは、マシン間での互換性がありませんので注意してください。以下に書き出されるデータの種類と順序を説明します。

(1) NCOL

NCOL は、プログラムの中の変数で、粒子輸送のタイミングの認識変数です。

NCOL

1	:	start of calculation
2	:	end of calculation
3	:	end of a batch
4	:	source
5	:	detection of geometry error
6	:	recovery of geometry error
7	:	termination by geometry error
8	:	termination by weight cut-off
9	:	termination by time cut-off
10	:	geometry boundary crossing
11	:	termination by energy cut-off
12	:	termination by escape or leakage
13	:	(n,x) reaction
14	:	(n,n’x) reaction
15	:	sequential transport only for tally

NCOL=1, 2, 3 の時は、書き出しは NCOL だけです。従って、以下の書き出しは、NCOL≥4 の時です。

(2) NOCAS, NOBCH, RCASC, RSOUIN

これら 4 つの変数は、NCOL=4 の時にだけ書き出されます。それぞれの意味は、

NOCAS	:	current event number in this batch
NOBCH	:	current batch number
RCASC	:	real number of NOCAS+maxcas*(NOBCH-1)
RSOUIN	:	sum of the weight of source particle

(3) NO, MAT, ITYP, KTYP, JTYP, MTYP, RTYP, OLDWT

これらの変数の意味は、

NO	:	cascade id in this event
MAT	:	material id
ITYP	:	particle type
KTYP	:	particle kf-code
JTYP	:	charge number of the particle
MTYP	:	baryon number of the particle
RTYP	:	rest mass of the particle (MeV)
OLDWT	:	wight of the particle at (x,y,z)

(a) QS
この変数は、ITYP=12, 13 の時、すなわち電子、陽電子の時のみ出力されます。QS は、電子の dE/dx です。

QS : dE/dx of electron at (x,y,z)

(4) IBLZ1, IBLZ2, ILEV1, ILEV2
これらの変数の意味は、

IBLZ1	:	cell id at (x,y,z)
IBLZ2	:	cell id after crossing
ILEV1	:	level structure id of the cell at (x,y,z)
ILEV2	:	level structure id of the cell after crossing

(a) ILAT1
階層構造をもつ CELL の変数です。書き出しは、ILEV1>0 の時に

```
write(io) ( ( ILAT1(i,j), i=1,5 ), j=1,ILEV1 )
```

で行われます

(b) ILAT2
階層構造をもつ CELL の変数です。書き出しは、ILEV2>0 の時に

```
write(io) ( ( ILAT2(i,j), i=1,5 ), j=1,ILEV2 )
```

で行われます

(5) COSTH, UANG(1), UANG(2), UANG(3), NSURF
各変数の意味は以下の通りです。これらはバージョン 2.30 から出力されるようになりました。

COSTH	:	cosine of an angle of incidence in a surface crossing
UANG(1,2,3)	:	x,y,z component of a normal vector of its surface, respectively
NSURF	:	internal number of the surface Note that this is different from the surface number defined in the [surface] section

(6) NAME, NCNT(1), NCNT(2), NCNT(3)
これらの変数の意味は、

NAME : collision number of the particle
NCNT(1,2,3) : values of counter 1, 2, and 3

(7) WT, U, V, W
これらの変数の意味は、

WT : wight of the particle at (xc,yc,zc)
U, V, W : unit vector of momentum of the particle

(8) E, T, X, Y, Z
これらの変数の意味は、

E : energy of the particle at (x,y,z) (MeV)
T : time of the particle at (x,y,z) (nsec)
X, Y, Z : position coordinate of the preceding event point (cm)

(9) EC, TC, XC, YC, ZC
これらの変数の意味は、

EC : energy of the particle at (xc,yc,zc) (MeV)
TC : time of the particle at (xc,yc,zc) (nsec)
XC, YC, ZC : position coordinate of the particle (cm)

(10) SPX, SPY, SPZ
これらの変数の意味は、

SPX, SPY, SPZ : unit vector of spin direction of the particle

(11) NZST
この変数は粒子がもつ電荷 (charge state) を表します。

(12) NCLSTS
この変数は、Collision が起こった時に、すなわち、NCOL=13, 14 の時にのみ書き出されます。意味は、衝突もしくは崩壊による生成粒子の個数です。原子核も含みます。NCLSTS>0 の時に、次の変数が書き出されます。

(a) MATHZ, MATHN, JCOLL, KCOLL
これらの変数の意味は、

MATHZ : Z number of the mother nucleus
MATHN : N number of the mother nucleus
JCOLL : reaction type id1
KCOLL : reaction type id2

JCOLL, KCOLL はそれぞれ次のような意味を持ちます。

JCOLL

0	:	nothing happen
1	:	Hydrogen collisions
2	:	Particle Decays
3	:	Elastic collisions
4	:	High Energy Nuclear collisions
5	:	Heavy Ion reactions
6	:	Neutron reactions by data
7	:	Photon reactions by data
8	:	Electron reactions by data
9	:	Proton reactions by data
10	:	Neutron event mode
11	:	delta ray production
13	:	Photon reactions by EGS5
14	:	Electron reactions by EGS5

KCOLL

0	:	normal
1	:	high energy fission
2	:	high energy absorption
3	:	low energy n elastic
4	:	low energy n non-elastic
5	:	low energy n fission
6	:	low energy n absorption

(b) ICLUSTS, JCLUSTS, QCLUSTS, JCOUNT

これらの変数は、各生成粒子の情報を記述するもので、NCLSTS だけ繰り返し、また、それぞれの変数も以下のように配列を持ちます。書き出しは以下のように行われます。

```
do i = 1, NCLSTS
   write(io) ICLUSTS(i)
   write(io) ( JCLUSTS(j,i), j=0,7)
   write(io) ( QCLUSTS(j,i), j=0,12)
   write(io) ( JCOUNT(j,i),  j=1,3)
end do
```

これらの変数の意味は、

ICLUSTS		kind of particle
0	:	nucleus
1	:	proton
2	:	neutron
3	:	pion
4	:	photon
5	:	kaon
6	:	muon
7	:	others

JCLUSTS(i)

i = 0	:	angular momentum
= 1	:	proton number
= 2	:	neutron number
= 3	:	ityp
= 4	:	status of the particle 0: real, <0 : dead
= 5	:	charge number
= 6	:	baryon number
= 7	:	kf code

QCLUSTS(i)

i = 0	:	impact parameter
= 1	:	px (GeV/c)
= 2	:	py (GeV/c)
= 3	:	pz (GeV/c)
= 4	:	$e_{tot} = \sqrt{p^2 + m^2}$ (GeV)
= 5	:	rest mass (GeV)
= 6	:	excitation energy (MeV)
= 7	:	kinetic energy (MeV)
= 8	:	weight
= 9	:	time (nsec)
= 10	:	x coordinate (cm)
= 11	:	y coordinate (cm)
= 12	:	z coordinate (cm)

4.2.22 Event Generator Mode

通常、核データを使う計算では、イベント毎のエネルギーと運動量は保存しません。モンテカルロのシミュレーションを多数回繰り返して、平均を取ると、エネルギーと運動量が保存することが保証されます。これは、核データを用いたモンテカルロシミュレーションが、一体の Boltzmann 方程式を数値的に解いていることに対応しています。従って、核データを用いた計算では、一体の観測量しか計算できません。例えば、ある領域の熱量、フラックスなどです。これらは平均値です。また、モンテカルロ計算で用いる核データも一体 (inclusive) のデータしか含まれていません。平均値の周りの分散は、二体以上の分布関数に関係しますから、一体の Boltzmann 方程式では導くことはできません。例えば、*PHITs* の [t-heat] にある、deposit エネルギー分布は、平均値の周りの分散ですから、二体以上の分布関数によって決まる量です。ですから、核データを用いた計算では求めることができません。反応前と反応後で入射粒子と原子核が変化しないのであれば、つまり、弾性散乱だけが起こると仮定すると、一体の情報から全てのエネルギーと運動量が決まりますから、計算できるわけです。これは極めて例外的な状況で、反応後の生成粒子が変化したり、複数になったりすると破綻します。

高エネルギーの核反応に対しては、通常、核データが整備されていないので、カスケードモデル、JAM、QMD、等の核反応モデルを利用します。これらのモデルは、ひとつの核反応毎に、エネルギー、運動量の保存したモンテカルロ手法で核反応を計算します。このような核反応モデルを用いた輸送計算は、全体として、ひとつのイベント毎にエネルギー、運動量の保存したシミュレーションになっていますから、実際の現象を模擬するという意味で、Event Generator と呼ばれています。輸送コードとして、核データを使ったシミュレーションと似ていますが、概念的には全く異なるものです。このシミュレーションでは、ひとつのイベント毎の観測量は物理的に意味を持ちません。ヒストリーで平均した量だけが物理的に意味を持ちます。平均値周りのイベントによる分散も計算できますが、これは、モンテカルロ手法の統計分散で、物理的なものではありません。物理的な分散を記述する二体以上の相関の情報が核データにも基礎となる方程式にも含まれていないからです。一方、Event Generator では、観測量の分散は物理的な分散に対応します。その精度の問題はありますが、全ての相関を記述している枠組みになっています。高エネルギーのシミュレーションは、大抵この Event Generator になっていて、一体の観測量のほかに、コインシデンス実験の解析や、測定器のレスポンス関数のシミュレーションに応用することができます。

最近、低エネルギーの中性子の関係した分野でも、従来の一体の観測量、熱量やフラックスのほかに、deposit エネルギー分布、2 粒子相関などの高次の相関が含まれる物理量が求められてくるようになりました。例えば、半導体素子の放射線によるソフトエラーの問題、細胞の放射線による影響の問題などは、微視的な領域での deposit エネルギーの分散が問題になります。これらの問題に従来の核データを用いたこのシミュレーションでは対応することができません。そこで、*PHITs* では、核データを使う低エネルギー領域でも Event Generator としてシミュレートできるモデルを作りました。これが、`e-mode = 1,2` で実行される EventGenerator Mode [23] [24] [25] です。

このモデルの詳細は、他の論文を見ていただくこととして、ここでは、まず、モデルの概略を説明します。中性子入射反応で核データが記述するものは、全断面積、Capture、Elastic、(n, n')、(n, Nn') 等のチャンネル毎の断面積、それと放出粒子の inclusive な微分断面積です。これらの情報からは、残留核の運動量、2 粒子の相関などは一意に決められません。情報が足りないからです。そこで、*PHITs* では、これら核データの情報を用いるとともに、特殊な統計崩壊モデルを導入しました。まず、核データのチャンネル毎の断面積を用います。次に、チャンネル毎に次のようなモデルを設定します。まず、Capture の場合、入射中性子が Capture された場合の融合原子核の運動量、励起エネルギーは運動学的に一意に決定されますから、この原子核に、中性子の崩壊確率をゼロにした統計崩壊を行います。この場合、光子の放出、荷電粒子の放出だけが含まれる統計崩壊を行うことになります。`e-mode = 1` では放出する粒子は区別されませんが、`e-mode=2` にすると反応チャンネルを反映して放出する粒子を決定します。たとえば、(n, γ) 反応なら必ず光子のみを放

[23] Y. Iwamoto *et al.*, International Conference on Nuclear Data for Science and Technology 2007, DOI: 10.1051/ndata:07417.
[24] K. Niita *et al.*, International Conference on Nuclear Data for Science and Technology 2007, DOI: 10.1051/ndata:07398.
[25] Y. Iwamoto *et al.*, Prog. Nucl. Sci. Technol. 2, 931-935 (2011).

出、(n,α) 反応なら α 粒子一個と光子のみを放出する、ということになります。次に、Elastic では、核データに従って、放出中性子の運動量を決定します。この運動量から運動学的に残留核の運動量が一意に決定されます。(n,n') では、核データに従って、まず、放出中性子の運動量を決定します。この運動量から運動学的に残留核の運動量と励起エネルギーが一意に決定されますから、この原子核に、中性子の崩壊確率をゼロにした統計崩壊を行います。Capture の時と同じように、光子の放出、荷電粒子の放出だけが含まれる統計崩壊を行うことになります。Elastic では `e-mode=1` も 2 も結果は同じになります。最後に、(n,Nn') 反応ですが、`e-mode=1` の場合はまず、核データを用いて一個目の放出中性子の運動量を決定します。この運動量から運動学的に残留核の運動量と励起エネルギーが一意に決定されますから、この原子核に、通常の統計崩壊を行います。即ち、中性子崩壊も含んだ、光子、荷電粒子の統計崩壊を行うことになります。この場合だけ、終状態の中性子の数が核データで指定された N になる保証はありませんが、そのずれは小さいことを確認しています。`e-mode=2` では、核データを元にした統計サンプリングを行うことで、N 個の中性子全ての運動量を同時に決定します。この運動量から運動学的に残留核の運動量と励起エネルギーが一意に決定され、続いて光子放出のみを考慮する統計崩壊を行います。このとき終状態の中性子の数は必ず N 個となり、二個目以降の中性子の運動量も核データから決定されるため、`e-mode=1` より精度が上がります。このようなモデルを用いることにより、核データを用いた低エネルギーの中性子の反応も、イベント毎にエネルギー、運動量の保存した Event Generator になり、核データだけでは記述できなかった、残留核の運動量、2 粒子相関など全ての情報がイベント毎に記述できるようになります。

次に、`e-mode` を使う場合の注意点を示します。まず、`dmax(2)` を設定し、核データを用意し、中性子の核データによる計算を可能にします。粒子放出の精度をあげるために、原子核の基底状態の近くの励起状態の情報を必要としますので、`igamma=1-3` が必要になります。次に、統計崩壊モデルは、GEM 即ち `nevap=3` が必要です。更に、熱中性子の温度依存の散乱の記述はできません。常に T=0 の取り扱いになります。これらは、`e-mode=1,2` を指定すると自動的に設定されます（特に指定しなければ `igamma` は 2 になります）。もし過去の計算と比較するなどの理由で `e-mode=1, igamma=1` を使いたい場合は、`igamma=1` としても計算はできます。

次に、`e-mode` を使うと可能になる新しい観測量を示します。`[t-yield], [t-product]` では、通常、dmax(2) 以下のエネルギー、即ち、中性子の核データを使う領域の生成核種、生成粒子はタリーされませんが、これが可能になります。`[t-heat]` では、`dmax(2)` 以下のエネルギー、即ち、中性子の核データを使う領域の発熱は、核データの Kerma Factor を用いますが、`e-mode` では、中性子からの直接の発熱の寄与はありません。中性子反応から生成された荷電粒子、残留核の Ionization Loss から計算されます。光子は、`electron=0` を指定すると光子の Kerma Factor を用いますが、電子の輸送も含み `electron=1` を指定すれば、電子の Ionization Loss から熱を計算します。`[t-dpa]` では、`dmax(2)` 以下のエネルギー、即ち、中性子の核データを使う領域でも、ライブラリーを用いず DPA を計算することが可能になります。

4.3 [Source] セクション

このセクションでは、入射粒子の情報を定義します。ソースタイプは、`s-type = 番号` で指定します。

表 29: ソースタイプ

ソースタイプ	説明
`s-type = 1` `s-type = 4`	円柱 (円面、ペンシルビーム) エネルギー分布を有する円柱
`s-type = 2` `s-type = 5`	角柱 (長方形) エネルギー分布を有する角柱
`s-type = 3` `s-type = 6`	ガウス分布 (x, y, z 独立) エネルギー分布を有するガウス分布 (x, y, z 独立)
`s-type = 7` `s-type = 8`	一般パラボラ分布 (x, y, z 独立) エネルギー分布を有する一般パラボラ分布 (x, y, z 独立)
`s-type = 9` `s-type = 10`	球及び球殻分布 エネルギー分布を有する球及び球殻分布
`s-type = 11`	ビーム方向に垂直な位相空間で一様分布
`s-type = 12`	decay-turtle の出力を読み込む
`s-type = 13` `s-type = 14`	ガウス分布 (xy 平面) エネルギー分布を有するガウス分布 (xy 平面)
`s-type = 15` `s-type = 16`	一般パラボラ分布 (xy 平面) エネルギー分布を有する一般パラボラ分布 (xy 平面)
`s-type = 17`	dump ファイルの読み込み
`s-type = 18` `s-type = 19`	円錐形状 エネルギー分布を有する円錐形状
`s-type = 20` `s-type = 21`	三角柱形状 エネルギー分布を有する三角柱形状
`s-type = 100`	ユーザー定義のソースプログラム usrsors.f にプログラムを書きこみコンパイルする。

4.3.1 <Source> : マルチソース

s-type で指定するソースを複数定義することができます。各ソースは、<source> = 相対比、の指定で始まるサブセクションで定義します。相対比は、マルチソースのソース粒子の相対比を指定します。また、totfact = で全体の規格化定数を定義します。ただし、<source>の上の行で reg を指定すると、<source>の行が継続行として扱われてエラーになるため、各<source>に属するパラメータとして reg を最後に書かないように注意してください。

表 30: マルチソース

パラメータ	説明
<source> =	マルチソースの開始、数字はこのソースの相対比
totfact =	(D=1) マルチソース全体の規格化定数 正の数で与えた場合、相対比に従ってソース粒子を生成。 負の数で与えた場合、同数のソース粒子を生成し、 相対比に従って weight を変化させる。
iscorr =	(D=0) マルチソース相関オプション。 0: 通常のマルチソース。 1: 複数のマルチソースを 1 つのイベントとしてまとめて発生させる。核反応により生成する複数の 2 次粒子を 1 つのイベントとして計算したい場合などに用いる。ただし、iscorr=2,3 の場合と違い、複数のソースが発生する場所は同一ではなく、それぞれが指定したソースの発生領域から確率的に選ばれる。 2: 1 と同じく複数のマルチソースを 1 つのイベントとするが、2 つ目以降の線源を 1 つ目の線源と同じ場所から発生させるオプション。任意の場所で発生する核反応を再現するときなどに使う。 3: 1 と同じく複数のマルチソースを 1 つのイベントとするが、2 つ目以降の線源を 1 つ目の線源と同じ場所から逆方向に発生させるオプション。任意の場所で 2 つに崩壊する粒子を再現するときなどに使う。 iscorr=1,2,3 の場合、<source>は整数の絶対値で定義し、その合計値を totfact で指定する必要があります。例えば、$(X, 1p2n)$ 反応を 1 イベントとして模擬する際は、陽子線源に対して<source>=1、中性子線源に対して<source>=2 とし、totfact=3 とします。

4.3.2 共通パラメータ

各ソースタイプに共通なパラメータを以下に示します。パラメータの順序は自由です。(D=***) のあるものは、省略可能です。原子核や軽複合粒子のエネルギーは、核子当たりで指定します。

表 31: 共通ソースパラメータ 1

パラメータ	説明
`proj =`	入射粒子 : 粒子の指定方法は表を参照

`sx =`
`sy =`
`sz =`
	(D=0) スピンの方向ベクトルの x 成分
	(D=0) スピンの方向ベクトルの y 成分
	(D=0) スピンの方向ベクトルの z 成分

`reg =`
 	
`ntmax =`
	(D=all) `[cell]` セクションで定義したセル番号を指定して、ソースを発生させる領域を限定します。すなわち、ソースタイプ毎に決まる領域とここで指定する領域が重なった部分からソースを発生させます。
	書式は `reg = { 1 - 5 } 10 34` です。また `reg = ( 6 < 10[1 0 0] < u=3 )` などの lattice, universe 構造も指定できます。詳細は、タリーの領域指定箇所を参照して下さい。
	(D=1000) 領域限定の際の最大再試行回数

`trcl =`	(D=なし) 座標変換番号もしくは座標変換定義
`wgt =`	(D=1.0) ソース粒子のウエイト。
`factor =`	(D=1.0) ソース粒子の規格化定数。全てのタリー結果を `factor` 倍します。ただし、マルチソースの場合は使えませんので、規格化は `totfact` を用いて行って下さい。また、マルチソースでない場合、`totfact * factor` が実質的な規格化定数になります。
`izst =`	(D=`proj` で指定した粒子の電荷) 入射粒子の電荷 (charge state)。 `[magnetic field]` や `[electro magnetic field]` セクションで定義できる磁場や電場中の運動を記述する際にのみ使用される。ここで指定した値は輸送の途中で変化しない。ただし、核反応で生成された原子核の電荷は Z に戻る。

入射粒子の方向は `dir, phi, dom` の 3 つパラメータを用いて指定します。図 7 にこれらと入射粒子の方向との関係を示します。入射粒子の方向を太い実線で表しており、極角と方位角をそれぞれ θ, ϕ とした場合に、$\cos\theta$ =`dir`, ϕ =`phi` の関係があります。ただし、`dir` は $\cos\theta$ の値を、`phi` は degree 単位の値を与えます。`phi` は省略可能で、その場合はランダムとなります。`dom` は入射粒子を発生させる際、一定の広がりをもたせる場合に利用します。`dir, phi` で決定した方向を中心に、平面角で ψ =`dom`、すなわち立体角として $2\pi(1-\cos\psi)$ の範囲で入射粒子の方向をランダムに決定します。`dom` の値も degree 単位です。

入射粒子の方向を等方分布させたい場合は `dir=all` と設定します。他に、`dir=data` とし、その下に `a-type` サブセクションを用意することにより、任意の関数やデータの角分布をもつ線源をシミュレートできます。

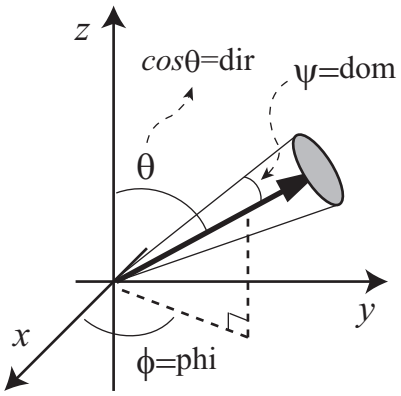


図 7: 入射粒子の方向とパラメータ `dir, phi, dom` の関係

`dir` は `s-type=9,10` では定義が違うので注意してください。また、`s-type=11,12` では `dir=` ±1 しか指定できません。

trcl の指定により、座標変換が使えます。wgt は、ソース粒子の初期ウエイトです。また、規格化定数 factor は、タリーなどの出力に乗じる定数です。dump ファイル読み込みの際に、オリジナルのソース当たりの出力にする場合などに便利です。wgt と factor は、逆数の関係になります。この規格化定数 factor は、マルチソースの場合は、各ソースの規格化です。全体のソースの規格化は、totfact を用います。

スピンは、指定しないか全てゼロの場合は、未定義のまま磁場に入ります。この場合、磁場に入ったときの磁場の方向と偏極率に従い、初期スピンが定義されます。ここで、スピンが指定された場合は、磁場の定義の偏極率に関係なく、ここで定義されたスピンの方向で磁場に入ります。

表 32: 共通ソースパラメータ 2

パラメータ	説明
ispfs = 0, 1, 2	(D=0) 自発核分裂からの中性子線源 ispfs = 1,2 を指定し、proj として、U-238, Pu-238, Pu-240, Pu-242, Cm-242, Cm-244, Cf-252, Th-232, U-232, U-233, U-234, U-235, U-236, Np-237, Pu-239, Pu-241, Am-241, Bk-249 の放射性各種を指定した時、これらの核種が自発核分裂を起こした場合に放出される中性子を線源とすることができる。 ispfs=1 の場合、タリーの計算結果は自発核分裂の個数当たりで規格化されるのに対し、ispfs=2 の場合は、自発核分裂で発生した中性子 1 個当たりで規格化される。この線源を使用した場合、[source] セクションで指定した e0 や e-type で指定されるエネルギー分布、また、dir や a-type で指定される角度分布は無視される。

ispfs を用いた中性子線源については、文献[26] にあるデータを参照しました。また、本文献にあるデータを用いる際、これらのデータ及び記述する FORTRAN プログラムの開発については、株式会社ナイスの Liem Peng Hong 氏のご協力を得ました。ここに氏への謝意を表します。

次のように、proj に Pu-240 を指定し、ispfs=1 とすると、Pu-240 の自発核分裂による放出される中性子が線源粒子となります。この場合、e0 の値も定義されていますが、無視されます。

```
[ S o u r c e ]
    s-type =   1
      proj =   240Pu
        e0 =   1.0
        z0 =   0
        z1 =   0
        r0 =   0.0
       dir =   1.0000
     ispfs = 1
```

[26] J. M. Verbeke, C. Hagmnn, and D. Wright, “Simulation of Neutron and Gamma Ray Emission from Fission and Photofission”, UCRL-AR-228518 (2014).

4.3.3 円柱分布ソース

円柱分布ソースタイプに必要なパラメータを以下に示します。パラメータの順序は自由です。(D=***) のあるものは、省略可能です。

表 33: 円柱分布ソースパラメータ

s-type = 1, 4	円柱のとき
x0 =	(D=0.0) 円柱の中心の x 座標 (cm)
y0 =	(D=0.0) 円柱の中心の y 座標 (cm)
z0 =	円柱の z 軸方向の下限 (cm)
z1 =	円柱の z 軸方向の上限 (円面の場合は下限と同じ) (cm)
r0 =	円柱の半径 (r0=0.0 のときペンシルビーム) (cm)
r1 =	(D=0.0) 円柱の内半径、中空部分を指定 (cm)
dir =	入射粒子の z 軸方向からの方向余弦
	all を指定した時は、等方分布
	data を指定した時は、a-type サブセクションが必要
phi =	(D=省略、ランダム) 入射粒子の方位角 (degree)。
dom =	(D=0.0) 入射粒子方向の立体角範囲 (degree)。
	= -1 ; $\cos^2$ bias 分布
e0 =	入射粒子のエネルギー (s-type=1) (MeV/u)

4.3.4 角柱分布ソース

角柱分布ソースタイプに必要なパラメータを以下に示します。パラメータの順序は自由です。(D=***) のあるものは、省略可能です。

表 34: 角柱分布ソースパラメータ

s-type = 2, 5	角柱のとき
x0 =	角柱の下限 x 座標 (cm)
x1 =	角柱の上限 x 座標 (cm)
y0 =	角柱の下限 y 座標 (cm)
y1 =	角柱の上限 y 座標 (cm)
z0 =	角柱の下限 z 座標 (cm)
z1 =	角柱の上限 z 座標 (長方形の場合は下限と同じ) (cm)
dir =	入射粒子の z 軸方向からの方向余弦
	all を指定した時は、等方分布
	data を指定した時は、a-type サブセクションが必要
phi =	(D=省略、ランダム) 入射粒子の方位角 (degree)。
dom =	(D=0.0) 入射粒子方向の立体角範囲 (degree)。
	= -1 ; $\cos^2$ bias 分布
e0 =	入射粒子のエネルギー (s-type=2) (MeV/u)

4.3.5 ガウス分布ソース (x, y, z 独立)

このガウス分布は、x, y, z 軸で独立なガウス分布です。ガウス分布 (x, y, z 独立) ソースタイプに必要なパラメーターを以下に示します。パラメーターの順序は自由です。(D=***) のあるものは、省略可能です。

表 35: ガウス分布 (x, y, z 独立) ソースパラメータ

`s-type = 3, 6`	ガウス分布 (x, y, z 独立) のとき
`x0 =`	(D=0.0) ガウス分布中心の x 座標 (cm)
`x1 =`	ガウス分布 x 方向半値全幅 (cm)
`y0 =`	(D=0.0) ガウス分布中心の y 座標 (cm)
`y1 =`	ガウス分布 y 方向半値全幅 (cm)
`z0 =`	(D=0.0) ガウス分布中心の z 座標 (cm)
`z1 =`	ガウス分布 z 方向半値全幅 (cm)
`dir =`	入射粒子の z 軸方向からの方向余弦。 all を指定した時は、等方分布。 data を指定した時は、a-type サブセクションが必要
`phi =`	(D=省略、ランダム) 入射粒子の方位角 (degree)
`dom =`	(D=0.0) 入射粒子方向の立体角範囲 (degree) $= -1$; $\cos^2$ bias 分布
`e0 =`	入射粒子のエネルギー (s-type=3) (MeV/u)

4.3.6 一般パラボラ分布ソース (x, y, z 独立)

このパラボラ分布は、x, y, z 軸で独立なパラボラ分布です。一般パラボラ分布 (x, y, z 独立) ソースタイプに必要なパラメータを以下に示します。パラメータの順序は自由です。(D=***) のあるものは、省略可能です。

表 36: 一般パラボラ分布 (x, y, z 独立) ソースパラメータ

`s-type = 7, 8`	一般パラボラ分布 (x, y, z 独立) のとき
`x0 =`	(D=0.0) X パラボラ分布中心の x 座標 (cm)
`x1 =`	X パラボラ分布の幅 (cm)
`y0 =`	(D=0.0) Y パラボラ分布中心の y 座標 (cm)
`y1 =`	Y パラボラ分布の幅 (cm)
`z0 =`	(D=0.0) Z パラボラ分布中心の z 座標 (cm)
`z1 =`	Z パラボラ分布の幅 (cm)
`rn =`	(D=2) 一般パラボラの次数
`dir =`	入射粒子の z 軸方向からの方向余弦。 all を指定した時は、等方分布。 data を指定した時は、a-type サブセクションが必要
`phi =`	(D=省略、ランダム) 入射粒子の方位角 (degree)
`dom =`	(D=0.0) 入射粒子方向の立体角範囲 (degree) $= -1$; $\cos^2$ bias 分布
`e0 =`	入射粒子のエネルギー (s-type=7) (MeV/u)

4.3.7 ガウス分布ソース (*xy* 平面)

このガウス分布は、*xy* 平面でのガウス分布です。ガウス分布 (*xy* 平面) ソースタイプに必要なパラメータを以下に示します。パラメータの順序は自由です。(D=***) のあるものは、省略可能です。

表 37: ガウス分布 (*xy* 平面) ソースパラメータ

`s-type = 13, 14`	ガウス分布 (*xy* 平面) のとき
`x0 =`	(D=0.0) ガウス分布中心の *x* 座標 (cm)
`y0 =`	(D=0.0) ガウス分布中心の *y* 座標 (cm)
`r1 =`	ガウス分布の半値全幅 (cm)
`z0 =`	*z* 軸方向の下限 (cm)
`z1 =`	(D=z0) *z* 軸方向の上限 (cm)
`dir =`	入射粒子の *z* 軸方向からの方向余弦。 all を指定した時は、等方分布。
	data を指定した時は、a-type サブセクションが必要
`phi =`	(D=省略、ランダム) 入射粒子の方位角 (degree)
`dom =`	(D=0.0) 入射粒子方向の立体角範囲 (degree)
	$= -1$; $\cos^2$ bias 分布
`e0 =`	入射粒子のエネルギー (s-type=13) (MeV/u)

4.3.8 一般パラボラ分布ソース (*xy* 平面)

このパラボラ分布は、*xy* 平面でのパラボラ分布です。一般パラボラ分布 (*xy* 平面) ソースタイプに必要なパラメータを以下に示します。パラメータの順序は自由です。(D=***) のあるものは、省略可能です。

表 38: 一般パラボラ分布 (*xy* 平面) ソースパラメータ

`s-type = 15, 16`	一般パラボラ分布 (*xy* 平面) のとき
`x0 =`	(D=0.0) パラボラ分布中心の *x* 座標 (cm)
`y0 =`	(D=0.0) パラボラ分布中心の *y* 座標 (cm)
`r1 =`	パラボラ分布の幅 (cm)
`z0 =`	*z* 軸方向の下限 (cm)
`z1 =`	(D=z0) *z* 軸方向の上限 (cm)
`rn =`	(D=2) 一般パラボラの次数
`dir =`	入射粒子の *z* 軸方向からの方向余弦。 all を指定した時は、等方分布。
	data を指定した時は、a-type サブセクションが必要
`phi =`	(D=省略、ランダム) 入射粒子の方位角 (degree)
`dom =`	(D=0.0) 入射粒子方向の立体角範囲 (degree)
	$= -1$; $\cos^2$ bias 分布
`e0 =`	入射粒子のエネルギー (s-type=15) (MeV/u)

4.3.9 球及び球殻分布ソース

球及び球殻分布ソースタイプに必要なパラメータを以下に示します。パラメータの順序は自由です。(D=***) のあるものは、省略可能です。

表 39: 球及び球殻分布ソースパラメータ

`s-type = 9, 10`	球及び球殻分布のとき
`x0 =`	(D=0.0) 球及び球殻分布中心の x 座標 (cm)
`y0 =`	(D=0.0) 球及び球殻分布中心の y 座標 (cm)
`z0 =`	(D=0.0) 球及び球殻分布中心の z 座標 (cm)
`r1 =`	球及び球殻分布の内半径、ゼロの場合は球 (cm)
`r2 =`	球及び球殻分布の外半径 (cm)
`dir =`	入射粒子の方向 `dir = 1.0` : 球の中心からの外向き法線方向 `dir = -1.0` : 球の中心からの法線の内向き `dir = all` : all を指定した時は、等方分布 `dir = -all` : 球の中心からの法線の内向き cos 分布 $\cos^2$ bias, 体積、面積計算に用いる data を指定した時は、a-type サブセクションが必要 `dir = iso` : 中心から `r1` の距離にある半径 `r2` の円面から垂直に球側に向けて一様分布。`r1=r2` の場合、`dir = -all` とほぼ同じ結果を与えるが、ウェイトを使っていないので、等方線源を使ってエネルギー付与の分散を計算する ([`t-deposit`] で output = deposit) 場合に用いる。
`e0 =`	入射粒子のエネルギー (s-type=9) (MeV/u)

s-type=9 のソースを flux を用いた体積、面積計算に用いるには、`dir = -all, r1 = r2` とします。その際の球の内側におけるフラックスは $1/\pi/(\mathtt{r1})^2$ となります。また、dir = iso とした場合も同様となります。

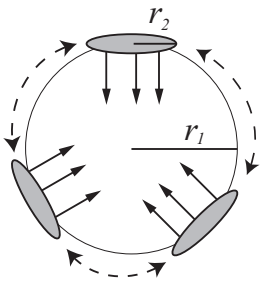


図 8: `dir=iso` の場合のイメージ図

4.3.10 s-type = 11 ソース

ビーム方向に垂直な位相空間で一様なソース分布を与えます。s-type = 11 の分布ソースタイプに必要なパラメータを以下に示します。パラメータの順序は自由です。(D=***) のあるものは、省略可能です。

表 40: s-type = 11 のソースパラメータ

s-type = 11	ビーム方向に垂直な位相空間で一様分布のとき
x0 =	(D=0.0) ビーム中心の x 座標 (cm)
x1 =	x 方向の最大半径/最大角度の比 (cm/mrad)
y0 =	(D=0.0) ビーム中心の y 座標 (cm)
y1 =	y 方向の最大半径/最大角度の比 (cm/mrad)
z0 =	z 軸方向の下限 (cm)
z1 =	z 軸方向の上限 (cm)
rx =	x 方向における位相空間の楕円の傾き (rad)
ry =	y 方向における位相空間の楕円の傾き (rad)
wem =	エミッタンス (π cm $\times$ mrad)
dir =	入射粒子の軸方向の方向余弦、(1 or −1 only)
e0 =	入射粒子のエネルギー (MeV/u)

4.3.11 s-type = 12 ソース

decay-turtle の出力をソースとして取り込みます。s-type = 12 の分布ソースタイプに必要なパラメータを以下に示します。パラメータの順序は自由です。(D=***) のあるものは、省略可能です。

入力ファイルのソースの数が足りなくなった場合は、初めに戻って同じデータを繰り返し使います。

表 41: s-type = 12 のソースパラメータ

s-type = 12	decay-turtle の出力を読み込むとき
x0 =	(D=0.0) ビームの x 座標オフセット (cm)
y0 =	(D=0.0) ビームの y 座標オフセット (cm)
z0 =	(D=0.0) ビームの z 座標オフセット (cm)
dir =	入射粒子の軸方向の方向余弦、(1 or −1 only)
file =	decay-turtle 出力ファイル名 (パスも含む)

decay-turtle の出力のデータ形式は、倍精度、アスキー形式で、各レコードは、

```
xp, xq, yp, yq, e0, wt0, pz0
```

で、それぞれは次の様な変数です。

表 42: decay-turtle データ

変数	説明
xp, yp	ビーム粒子の入射位置 (cm)
xq, yq	ビーム粒子の進行方向に垂直な面となす角度 (mrad)
e0	ビーム粒子の運動量 (GeV/c)
wt0	ビーム粒子のウエイト
pz0	ビーム粒子の偏極度 (未使用)

4.3.12 円錐形状分布ソース

円錐形状分布ソースタイプに必要なパラメータを以下に示します。パラメータの順序は自由です。(D=***)のあるものは、省略可能です。

表 43: 円錐形状分布ソースパラメータ

s-type = 18, 19	円錐形状
x0 =	(D=0.0) 円錐の頂点の x 座標 (cm)
y0 =	(D=0.0) 円錐の頂点の y 座標 (cm)
z0 =	(D=0.0) 円錐の頂点の z 座標 (cm)
x1 =	円錐頂点から底面中心に向けたベクトルの x 成分 (cm)
y1 =	円錐頂点から底面中心に向けたベクトルの y 成分 (cm)
z1 =	円錐頂点から底面中心に向けたベクトルの z 成分 (cm)
r0 =	(D=0.0) 頂点から円錐上面までの軸上における距離 (cm)
r1 =	頂点から円錐下面までの軸上における距離 (cm)
r2 =	円錐の母線と軸のなす角度 (degree)
dir =	入射粒子の z 軸方向からの方向余弦。 all を指定した時は、等方分布。 data を指定した時は、a-type サブセクションが必要
phi =	(D=省略、ランダム) 入射粒子の方位角 (degree)
dom =	(D=0.0) 入射粒子方向の立体角範囲 (degree) $= -1$; $\cos^2$ bias 分布
e0 =	入射粒子のエネルギー (s-type=18) (MeV/u)

4.3.13 三角柱形状分布ソース

`s-type = 20, 21` とすることで、三角柱形状に線源を発生させることができます。パラメータ `exa` を指定することで、上底面もしくは下底面から、指数関数的 $\exp(-ax)$ にソース強度が減衰する分布、もしくは一様な分布を定義することができます。前者は、`exa` に正の値を与えることで、係数の値 a を指定できます。

三角柱形状分布のソースタイプに必要なパラメータを以下に示します。パラメータの順序は自由です。(D=***) のあるものは、省略可能です。

表 44: 三角柱状分布ソースパラメータ

`s-type = 20, 21`	三角柱形状
`x0 =`	(D=0.0) 三角柱の頂点の x 座標 (cm)
`y0 =`	(D=0.0) 三角柱の頂点の y 座標 (cm)
`z0 =`	(D=0.0) 三角柱の頂点の z 座標 (cm)
`x1 =`	三角柱頂点から第 1 頂点へのベクトルの x 成分 (cm)
`y1 =`	三角柱頂点から第 1 頂点へのベクトルの y 成分 (cm)
`z1 =`	三角柱頂点から第 1 頂点へのベクトルの z 成分 (cm)
`x2 =`	三角柱頂点から第 2 頂点へのベクトルの x 成分 (cm)
`y2 =`	三角柱頂点から第 2 頂点へのベクトルの y 成分 (cm)
`z2 =`	三角柱頂点から第 2 頂点へのベクトルの z 成分 (cm)
`x3 =`	三角柱頂点から底面頂点へのベクトルの x 成分 (cm)
`y3 =`	三角柱頂点から底面頂点へのベクトルの y 成分 (cm)
`z3 =`	三角柱頂点から底面頂点へのベクトルの z 成分 (cm)
`exa =`	指数関数減衰の係数。exa>0; $\exp(-ax)$, exa=0; 一様分布
`dir =`	入射粒子の z 軸方向からの方向余弦。all を指定した時は等方分布。 data を指定した時は、a-type サブセクションが必要
`phi =`	(D=省略、ランダム) 入射粒子の方位角 (degree)
`dom =`	(D=0.0) 入射粒子方向の立体角範囲 (degree) $= -1$; $\cos^2$ bias 分布
`e0 =`	入射粒子のエネルギー (s-type=20) (MeV/u)

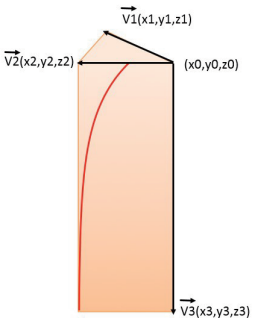


図 9: 三角柱分布の座標の指定

4.3.14 dump データソース

s-type=17 とすることで dump ファイルに落とされた粒子情報を線源として利用できます。パラメータ `idmpmode` を切り替えることにより、粒子情報の取り扱い方法として 2 種類を選択することができます。`idmpmode=0` の場合は、dump ファイルに記録された粒子を一つ一つの独立したイベントによる粒子であるとして統計処理を行います。対して `idmpmode=1` の場合は、dump ファイルに記録されたヒストリーとバッチの情報を利用することで、dump ファイルを作成した際のイベント情報を基に粒子間の相関を考慮した統計処理を行います。dump ファイルを用いたつなぎ計算を行う場合は、前段階のイベント情報を考慮して統計誤差を評価できるように `idmpmode=1` としてください。[27]

このソースタイプに関連するパラメータは以下の通りです。パラメータの順序は自由です。(D=***) のあるものは、省略可能です。

表 45: dump データのソースパラメータ (1)

`s-type = 17`	dump ファイルを読み込むとき
`file =`	dump ファイル名 (パスも含む)
`dump =` (次行)	dump データの個数、負で与えた時はアスキーファイル dump データの情報
(以下省略可能)	以下のものが指定されたとき、dump データがある場合も dump データより優先します。
`x0 =`	下限 x 座標 (cm)
`x1 =`	上限 x 座標 (cm)
`y0 =`	下限 y 座標 (cm)
`y1 =`	上限 y 座標 (cm)
`z0 =`	下限 z 座標 (cm)
`z1 =`	上限 z 座標 (cm)
`sx =`	スピンの方向ベクトルの x 成分
`sy =`	スピンの方向ベクトルの y 成分
`sz =`	スピンの方向ベクトルの z 成分
`dir =`	入射粒子の z 軸方向からの方向余弦 all を指定した時は、等方分布 data を指定した時は、a-type サブセクションが必要
`phi =`	(D=省略、ランダム) 入射粒子の方位角 (degree)
`dom =`	(D=0.0) 入射粒子方向の立体角範囲 (degree) $= -1$; $\cos^2$ bias 分布
`e0 =`	入射粒子のエネルギー (MeV/u)
`e-type =`	入射粒子のエネルギー分布指定
`wgt =`	(D=1.0) ソース粒子のウエイト

[27] `idmpmode=1` および `dmpmulti` による使い回し計算は、日本原子力学会「2015 年春の年会」での波戸氏 (KEK) らの発表 “モンテカルロつなぎ計算における不確かさ評価”（演題番号 N50）を参考に改良を行ったものです。

表 46: dump データのソースパラメータ (2)

(以下省略可能)	以下のものも指定可能です
factor =	(D=1.0) ソース粒子のウエイトの規格化定数
t-type =	(D=0) 時間分布
reg =	(D=all) 領域を限定する
ntmax =	(D=1000) 領域限定の際の最大再試行回数
trcl =	(D=なし) 座標変換番号もしくは座標変換定義
idmpmode =	dump ソースモードの選択 (D=0) dump データが nocas と nobach の情報を含まない場合 (D=1) dump データが nocas と nobach の情報を含む場合 =0: dump ファイルに記録された粒子情報を 1 つ 1 つの独立したヒストリーとして統計処理を実行する =1: dump ファイルを作成した際のヒストリーの情報（粒子間の相関）を考慮し統計処理を実行する
dmpmulti =	dump ファイルの使い回しの回数 (D=0.0) idmpmode=0 の場合 (D=1.0) idmpmode=1 の場合 =0.0: (maxcas*maxbch) で指定した全ヒストリー数の計算が終了するまで使いまわす。ただし、idmpmode=1 の場合は指定できない >0.0: 指定した回数だけ粒子情報を使いまわす。小数点以下の数字はデータを確率的に採用（ロシアンルーレット）することで考慮する

idmpmode=1 を使用するためには、使用する dump データにヒストリー番号 (nocas) とバッチ番号 (nobch) が含まれている必要があります。PHITS のタリー出力により dump ファイルを生成すると、ファイル名に “_dmp ” が付いた dump ファイルと共に “_dmp ” が付かない通常のタリーファイルが生成されますが（5.7.20 節参照）、idmpmode=1 を使用するためには、この通常タリーファイルも必要とされますので、使用する dump ファイルと同じフォルダに配置してください。idmpmode=1 では、この通常タリーファイルから前段階の maxcas,maxbch の情報を読み込みます。これによりインプットファイルに記述した maxcas,maxbch の値は無視されます。また、idmpmode=1 ではソースの規格化定数 totfact の値が無視されます。ソースの規格化定数 totfact を乗じた計算を行いたい場合は、dump ファイルを作成する前段階の計算において乗じるようにしてください。dump ファイル内の粒子のウェイト値に反映されます。idmpmode=1 はマルチソースと同時に使うことはできません。

dmpmulti は dump ファイルの使い回しの回数を制御します。dmpmulti=2.0 の場合 dump ファイルを 2 回使い回しますが、具体的には、1 個の粒子情報に対してウエイト値を 0.5 倍した 2 個の粒子を線源として発生させた計算を実行します。小数点以下の数字は、使い回す回数をもう 1 つ増やすかどうかに関連し、各々の粒子情報に対して乱数により決定します。例えば、dmpmulti=2.3 の場合、使い回す回数が 70%の確率で 2 回、30%の確率で 3 回となります。また、dmpmulti=0.0 を指定した場合は、maxcas,maxbch で指定される全イベントが終了するまで、dump ファイルを使い回します。ただし、これは idmpmode=0 でのみ有効となります。

dump ファイルを線源として使う計算では、基本的に再開始計算 (istdev<0) はできません。dmpmulti=0.0 の時のみ可能ですが、この場合は idmpmode=0 となるため、dump データの粒子情報を 1 つ 1 つの独立したヒストリーとして扱うことになり、データが十分にないと統計的に偏った結果を与えることになります。よって、統計量を増やした結果を得るためには、基本的には dmpmulti の値を大きく取って再度計算してく

ださい。ただしこの場合も、dump ファイルにあるデータ量が十分でないと、`dmpmulti` を増やしても、求める精度まで統計誤差が小さくならない可能性があるのでご注意ください。データ量が少ない場合は、dump ファイルにデータを蓄える計算の段階からやり直す必要があります。

dump データの情報は、まず `dump =`でデータの個数を指定します。正で与えた時はバイナリーファイルの読み込み、負で与えた時はアスキーファイルの読み込みをします。次の行に 1 つのレコードのデータの並びを指定します。その時の番号と物理量の関係は、以下の通りです。

表 47: dump データの種類と番号 (1)

物理量	kf	x	y	z	u	v	w	e	wt	time	c1	c2	c3	sx	sy	sz
番号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

表 48: dump データの種類と番号 (2)

物理量	name	nocas	nobch	no
番号	17	18	19	20

ここで、kf は粒子を識別する kf-code（表 4 を参照）、x, y, z 単位を cm とする空間座標、u, v, w は運動量の単位ベクトル、e は MeV を単位とするエネルギー（原子核の場合は MeV/u）、wt は粒子ウエイト、time は時間 (単位は nsec)、c1, c2, c3 はカウンターの値、sx, sy, sz はスピンの方向ベクトルです。name は粒子の衝突回数、nocas はあるバッチの中のヒストリー番号、nobch はバッチ番号、no はそのヒストリー中でのカスケード ID です。これらはバイナリーのときは real*8 のデータで、アスキーのときは *n*(1p1d24.15) のデータフォーマットで格納されています。

例えば、9 つのデータが次の順番で並んでいる時、

```
kf  e  wt  x  y  z  u  v  w
```

このデータを読み込むには、

```
dump = 9
  1  8  9  2  3  4  5  6  7
```

と指定します。

4.3.15 ユーザー定義ソース

usrsors.f にプログラムを書き込むことにより、ユーザー定義のソースを利用することができます。以下に示すソースのパラメータをソースセクションに指定した場合は、ユーザーソースで定義された値よりも優先します。

表 49: ユーザー定義ソースで指定できるパラメータ

`s-type = 100`	ユーザー定義ソースのとき
(以下省略可能)	以下のものが指定されたとき、ユーザー定義データがある場合もユーザー定義データより優先します
`x0 =`	下限 x 座標 (cm)
`x1 =`	上限 x 座標 (cm)
`y0 =`	下限 y 座標 (cm)
`y1 =`	上限 y 座標 (cm)
`z0 =`	下限 z 座標 (cm)
`z1 =`	上限 z 座標 (cm)
`sx =`	スピンの方向ベクトルの x 成分
`sy =`	スピンの方向ベクトルの y 成分
`sz =`	スピンの方向ベクトルの z 成分
`dir =`	入射粒子の z 軸方向からの方向余弦 all を指定した時は、等方分布 data を指定した時は、a-type サブセクションが必要
`phi =`	(D=省略、ランダム) 入射粒子の方位角 (degree)
`dom =`	(D=0.0) 入射粒子方向の立体角範囲 (degree) $= -1$; $\cos^2$ bias 分布
`e0 =`	入射粒子のエネルギー (MeV/u)
`e-type =`	入射粒子のエネルギー分布指定
`wgt =`	(D=1.0) ソース粒子のウエイト
`factor =`	(D=1.0) ソース粒子のウエイトの規格化定数
`t-type =`	(D=0) 時間分布
`reg =`	(D=all) 領域を限定する
`ntmax =`	(D=1000) 領域限定の際の最大再試行回数
`trcl =`	(D=なし) 座標変換番号もしくは座標変換定義

デフォルトで usrsors.f に入っているプログラムは、以下のようなものです。最初のコメント部に、ソースに必要な変数と、粒子指定に必要な kf code のリストがあります。次に、プログラム内で使用できる一様乱数と、ガウス乱数の function の説明があります。プログラムの最初の部分は、初期化の際のファイルのオープンの例があります。最後の部分に必要な変数の値の一つの例が書かれています。この例題を参考にユーザー定義ソースを書いてください。ソースセクションで指定した変数は、優先しますから、その場合は、ここで定義する必要はありません。

List 4.1 ● usrsors.f

```
 1:  ************************************************************************
 2:        subroutine usrsors(x,y,z,u,v,w,e,wt,time,name,kf,nc1,nc2,nc3,
 3:     &                   sx,sy,sz)
 4:  *        sample subroutine for user defined source.                    *
 5:  *        variables :                                                   *
 6:  *           x, y, z : position of the source.                          *
 7:  *           u, v, w : unit vector of the particle direction.           *
 8:  *           e       : kinetic energy of particle (MeV).                *
 9:  *           wt      : weight of particle.                              *
10:  *           time    : initial time of particle. (ns)                   *
11:  *           name    : usually = 1, for Coulmb spread.                  *
12:  *           kf      : kf code of the particle.                         *
13:  *           nc1     : initial value of counter 1                       *
14:  *           nc2     : initial value of counter 2                       *
15:  *           nc3     : initial value of counter 3                       *
16:  *           sx,sy,sz : spin components                                 *
17:  *-----------------------------------------------------------------------*
18:  *        kf code table                                                 *
19:  *           kf-code: ityp :  description                               *
20:  *             2212 :   1  :  proton                                    *
21:  *             2112 :   2  :  neutron                                   *
22:  *              211 :   3  :  pion (+)                                  *
23:  *              111 :   4  :  pion (0)                                  *
24:  *             -211 :   5  :  pion (-)                                  *
25:  *              -13 :   6  :  muon (+)                                  *
26:  *               13 :   7  :  muon (-)                                  *
27:  *              321 :   8  :  kaon (+)                                  *
28:  *              311 :   9  :  kaon (0)                                  *
29:  *             -321 :  10  :  kaon (-)                                  *
30:  *           kf-code of the other transport particles                   *
31:  *               12 :         nu_e                                      *
32:  *               14 :         nu_mu                                     *
33:  *              221 :         eta                                       *
34:  *              331 :         eta'                                      *
35:  *             -311 :         k0bar                                     *
36:  *            -2112 :         nbar                                      *
37:  *            -2212 :         pbar                                      *
38:  *             3122 :         Lanbda0                                   *
39:  *             3222 :         Sigma+                                    *
40:  *             3212 :         Sigma0                                    *
41:  *             3112 :         Sigma-                                    *
42:  *             3322 :         Xi0                                       *
43:  *             3312 :         Xi-                                       *
44:  *             3334 :         Omega-                                    *
45:  *-----------------------------------------------------------------------*
46:  *        available function for random number                          *
47:  *           unirn(dummy) : uniform random number from 0 to 1           *
48:  *           gaurn(dummy) : gaussian random number                      *
49:  *                          for exp( - x**2 / 2 / sig**2 ) : sig = 1.0  *
50:  ************************************************************************
51:        implicit real*8 (a-h,o-z)
```

```
*-----------------------------------------------------------------------
      parameter ( pi  = 3.141592653589793d0 )
      data ifirst / 0 /
      save ifirst
      character filenm*50
*-----------------------------------------------------------------------
*        example of initialization
*-----------------------------------------------------------------------
         if( ifirst .eq. 0 ) then
c              filenm = 'input.dat'
c              inquire( file = filenm, exist = exex )
c              if( exex .eqv. .false. ) then
c                 write(*,*) 'file does not exist =>  ', filenm
c                 call parastop( 887 )
c                 end if
c              open(71, file = file(i), status = 'old' )

c              close(71)
               ifirst = 1
         end if
*-----------------------------------------------------------------------
*        example for 3 GeV proton with z-direction
*-----------------------------------------------------------------------
            x = 0.0
            y = 0.0
            z = 0.0
            u = 0.0
            v = 0.0
            w = 1.0
            e = 3000.0
           wt = 1.0
         time = 0.0
         name = 1
           kf = 2212
          nc1 = 0
          nc2 = 0
          nc3 = 0
           sx = 0.d0
           sy = 0.d0
           sz = 0.d0
*-----------------------------------------------------------------------
      return
      end
```

4.3.16 エネルギー分布の定義

ソースがエネルギー分布をもつ s-type = 4, 5, 6, 8, 10, 14, 16、もしくは s-type = 17, 100 の場合は、e-type サブセクションを用いてソース粒子のエネルギー分布を指定する必要があります。原子核入射のときは、核子当たりのエネルギーです。

e-type = 1,4,11,14 の場合は、各ビンのエネルギー幅で積分した値を与えることで、エネルギー分布を表現します。一方、e-type = 21,24,31,34 の場合は、エネルギー微分値、すなわち単位が [particles/MeV] で与えられた線源スペクトルを定義してください。e-type = 2,3,5,6,7,12,14,15,16 の場合は、エネルギーの微分スペクトルを様々な関数を用いて与えることができます。また、e-type = 8,9,18,19 の場合は、エネルギー分布がポイントワイズ、すなわち各点各点で与えられる不連続なものとなります。e-type = 28,29 の場合は、RI 線源発生機能が利用できます。

(D=***) のあるものは、省略可能です。

表 50: ソースエネルギー分布パラメータ (1)

パラメータ	説明
e-type = 1, (11)	エネルギー分点 e(i) と各ビンにおけるソース粒子の生成確率の積分値 w(i) を与えることにより、任意のエネルギー分布を指定する。統計的に w(i) に比例するように各ビンに生成する粒子の数を調整し、エネルギー分布を表現する。11 の時は、エネルギーを波長 (Å) で与える。
ne =	エネルギー群数。ne が正の場合は、各ビン内の微分フラックスを [1/MeV] の単位で表示させた際に一定となるように粒子を生成し、負の場合は、[1/Lethargy] の単位で一定となるように粒子を生成する。データは自由フォーマットで次の様に与える。 (e(i),w(i),i=1,|ne|), e(|ne|+1) 各ビンに生成される粒子数の積分値は w(i) に比例する。
e-type = 4, (14)	e-type = 1, (11) と同じエネルギー分布を発生させる。ただし、e-type = 1, (11) が生成数を調整してエネルギー分布を表現するのに対し、e-type = 4, (14) では、全てのエネルギービンに同数の粒子を生成させ、粒子のウエイトの積分値を w(i) に比例して変化させることによりエネルギー分布を表現する。 また、p-type = 1 とし生成個数比 p(i) を与えることにより、各ビンの生成個数を変化させ、特定のエネルギーをもつソースの統計量を変化させることができる。14 の時は、エネルギーを波長 (Å) で与える。
ne =	エネルギー群数。ne が正の場合は、各ビン内の微分フラックスを [1/MeV] の単位で表示させた際に一定となるように粒子を生成し、負の場合は、[1/Lethargy] の単位で一定となるように粒子を生成する。データは自由フォーマットで次の様に与える。 (e(i),w(i),i=1,|ne|), e(|ne|+1) デフォルト（p-type = 0）では各ビンに等しい個数が生成される。 p-type = 1 で p(i) を設定した場合は、それらの値に各ビンの生成粒子数の積分値は比例する。
p-type = 0, 1	(D=0) 生成個数のオプション for 0, 全てのビンで p(i)=1、以下のデータは無し for 1, 各ビンの生成個数比 p(i) を次の行からデータで与える (p(i),i=1,ne)

中性子光学のために、エネルギー分布を波長 (Å) で指定できるオプションを設けました。e-type=1, e-type=2 の代わりに e-type=11, e-type=12 を指定すると、エネルギーの入力を全て波長に置き換えて読み込みます。その他の場合は、入力で変換式を使って MeV に直してください。例えば、e0 = 8.180425e-8/13**2 は、13Å の中性子のエネルギーを与えます。

`e-type = 1` の場合の入力フォーマットは以下のようになります。

```
e-type = 1
    ne = n
       e(1)   w(1)
       e(2)   w(2)
       e(3)   w(3)
       ....   ...
       e(n-1) w(n-1)
       e(n)   w(n)
       e(n+1)
```

この場合、エネルギー分布は、

```
e(1)-e(2)   w(1)
e(2)-e(3)   w(2)
e(3)-e(4)   w(3)
.......     ...
e(n-1)-e(n) w(n-1)
e(n)-e(n+1) w(n)
```

のように与えられます。よって例えば、0-2MeV 間を 0.2、2-4MeV 間を 0.6、4-6MeV 間を 0.2 としたい場合、入力のフォーマットは

```
e-type = 1
    ne = 3
       0   0.2
       2   0.6
       4   0.2
       6
```

となります。

表 51: ソースエネルギー分布パラメータ (2)

パラメータ	説明
e-type = 21, (31)	エネルギー分点 e(i) と各ビンにおけるソース粒子の生成確率の微分量 [dφ/dE](i) を与えることにより、任意のエネルギー分布を指定する。 統計的に [dφ/dE](i)*{e(i+1)-e(i)} に応じて各ビンに生成する粒子を調整し、エネルギー分布を表現する。 31 の時は、エネルギーを波長 (Å) で与える。
ne =	エネルギー群数。ne が正の場合は、各ビン内の微分フラックスを [1/MeV] の単位で表示させた際に一定となるように粒子を生成し、負の場合は、[1/Lethargy] の単位で一定となるように粒子を生成する。 データは自由フォーマットで次の様に与える。 (e(i),[dφ/dE](i),i=1,|ne|), e(|ne|+1) 各ビンに生成される粒子数の積分値は dφ/dE(i)*{e(i+1)-e(i)} に比例する。
e-type = 24, (34)	e-type = 21, (31) と同じエネルギー分布を発生させる。ただし、e-type = 21, (31) が生成数を調整してエネルギー分布を表現するのに対し、e-type = 24, (34) では、全てのエネルギービンに同数の粒子を生成させ、粒子のウエイトの積分値を w(i)*{e(i+1)-e(i)} に比例して変化させることによりエネルギー分布を表現する。 また、p-type = 1 とし生成個数比 p(i) を与えることにより、各ビンの生成個数を変化させ、特定のエネルギーをもつソースの統計量を変化させることができる。 34 の時は、エネルギーを波長 (Å) で与える。
ne =	エネルギー群数。ne が正の場合は、各ビン内の微分フラックスを [1/MeV] の単位で表示させた際に一定となるように粒子を生成し、負の場合は、[1/Lethargy] の単位で一定となるように粒子を生成する。 データは自由フォーマットで次の様に与える。 (e(i),w(i),i=1,|ne|), e(|ne|+1) デフォルト（p-type = 0）では各ビンに等しい個数が生成される。 p-type = 1 で p(i) を設定した場合は、それらの値に各ビンの生成粒子数の積分値は比例する。
p-type = 0, 1	(D=0) 生成個数のオプション for 0, 全てのビンで p(i)=1、以下のデータは無し for 1, 各ビンの生成個数比 p(i) を次の行からデータで与える (p(i),i=1,ne)

表 52: ソースエネルギー分布パラメータ (3)

パラメータ	説明
e-type = 8, (18)	エネルギー分点 e(i) と各点のソース粒子の生成確率 w(i) を与えることにより、任意のエネルギー分布を指定する。統計的に w(i) に比例するように各点に粒子を生成させ、不連続なエネルギー分布を表現する。 18 の時は、エネルギーを波長 (Å) で与える。
ne =	エネルギー点数。データは自由フォーマットで次の様に与える。 (e(i),w(i),i=1,ne) 各点に生成される粒子数は w(i) に比例。
e-type = 9, (19)	e-type = 8, (18) と同じエネルギー分布を発生させる。ただし、e-type = 8, (18) が生成数を調整してエネルギー分布を表現するのに対し、e-type = 9, (19) では、全てのエネルギー点に同数の粒子を生成させ、粒子のウエイトを w(i) に比例して変化させることにより不連続なエネルギー分布を表現する。 また、p-type = 1 とし生成個数比 p(i) を与えることにより、各点の生成個数を変化させ、特定のエネルギーをもつソースの統計量を変化させることができる。 19 の時は、エネルギーを波長 (Å) で与える。
ne =	エネルギー点数。データは自由フォーマットで次の様に与える。 (e(i),w(i),i=1,ne) デフォルト (p-type=0) で各点に等しい個数が生成される。 各点に生成される粒子数は p(i) に比例。
p-type = 0, 1	(D=0) 生成個数のオプション for 0, 全ての点で p(i)=1、以下のデータは無し for 1, 各点の生成個数比 p(i) を次の行からデータで与える (p(i),i=1,ne)

表 53: ソースエネルギー分布パラメータ (4)

パラメータ	説明
e-type = 2, (12)	微分線源スペクトル (dφ/dE) をガウス分布で与える。 12 の時は、エネルギーを波長 (Å) で与える
eg0 = eg1 = eg2 = eg3 =	ガウス分布の中心値 (MeV) ガウス分布の半値全幅 (MeV) ガウス分布のカットオフ最小値 (MeV) ガウス分布のカットオフ最大値 (MeV)
e-type = 3	微分線源スペクトル (dφ/dE) を Maxwellian 分布 $f(x) = x^{1.5}\exp(-x/T)$ で与える。
nm =	(D=−200) エネルギー群数。nm を正の数で与えた時は線形で等分点を決定し、負の数で与えた時は対数で等分点を決定。
et0 = et1 = et2 =	温度パラメーター T (MeV) Maxwellian 分布のカットオフ最小値 (MeV) Maxwellian 分布のカットオフ最大値 (MeV)
e-type = 7	e-type = 3 と同じエネルギー分布を発生させる。ただし、e-type = 3 が生成数を調整してエネルギー分布を表現するのに対し、e-type = 7 では、全てのエネルギービンに同数の粒子を生成させ、粒子のウエイトの積分値を Maxwellian 分布 $f(x) = x^{1.5}\exp(-x/T)$ に応じて変化させることによりエネルギー分布を表現する。 また、p-type = 1 とし生成個数比 p(i) を与えることにより、各ビンの生成個数を変化させ、特定のエネルギーをもつソースの統計量を変化させることができる。
nm =	(D=−200) エネルギー群数。nm を正の数で与えた時は線形で等分点を決定し、負の数で与えた時は対数で等分点を決定。 デフォルト（p-type = 0）では各ビンに等しい個数が生成される。 p-type = 1 で p(i) を設定した場合は、それらの値に各ビンの生成粒子数の積分値は比例する。
et0 = et1 = et2 =	温度パラメーター T (MeV) Maxwellian 分布のカットオフ最小値 (MeV) Maxwellian 分布のカットオフ最大値 (MeV)
p-type = 0, 1	(D=0) 生成個数のオプション for 0, 全てのビンで p(i)=1、以下のデータは無し for 1, 各ビンの生成個数比 p(i) を次の行からデータで与える、(p(i),i=1,nm)

表 54: ソースエネルギー分布パラメータ (5)

パラメータ	説明
e-type = 5, (15)	微分線源スペクトル (dφ/dE) を任意の関数 f(x) で与える。 15 の時は、エネルギーを波長 (Å) で与える。
f(x) =	FORTRAN 形式で書いた関数。x はエネルギー (MeV/u) を表す。 内部変数や定数が使えます。例）f(x) = exp(-c1*x**2)
nm =	(D=−200) エネルギー群数。nm を正の数で与えた時は線形で 等分点を決定し、負の数で与えた時は対数で等分点を決定。
eg1 =	エネルギー分布のカットオフ最小値 (MeV)
eg2 =	エネルギー分布のカットオフ最大値 (MeV)
e-type = 6, (16)	e-type = 5, (15) と同じエネルギー分布を発生させる。 ただし、e-type = 5, (15) が生成数を調整してエネルギー 分布を表現するのに対し、e-type = 6, (16) では、全ての エネルギービンに同数の粒子を生成させ、粒子のウエイトの 積分値が任意の関数 $f(x)$ に応じて変化させることによりエネ ルギー分布を表現する。 また、p-type = 1 とし生成個数比 p(i) を与えることにより、 各ビンの生成個数を変化させ、特定のエネルギーをもつソース の統計量を変化させることができる。 16 の時は、エネルギーを波長 (Å) で与える。
f(x) =	FORTRAN 形式で書いた関数。x はエネルギー (MeV/u) を表す。 内部変数や定数が使えます。例）f(x) = exp(-c1*x**2)
nm =	(D=−200) エネルギー群数。nm を正の数で与えた時は線形で 等分点を決定し、負の数で与えた時は対数で等分点を決定。 デフォルト（p-type = 0）では各ビンに等しい個数が生成 される。p-type = 1 で p(i) を設定した場合は、それらの 値に各ビンの生成粒子数の積分値は比例する。
eg1 =	エネルギー分布のカットオフ最小値 (MeV)
eg2 =	エネルギー分布のカットオフ最大値 (MeV)
p-type = 0, 1	(D=0) 生成個数のオプション for 0, 全てのビンで p(i)=1、以下のデータは無し for 1, 各ビンの生成個数比 p(i) を次の行からデータで与える、 (p(i),i=1,nm)

表 55: ソースエネルギー分布パラメータ (6)

パラメータ	説明
e-type= 28,29	放射性核種とその放射能（単位 Bq) を与えることにより、その放射性核種の崩壊に伴って放出されるガンマ線を線源とした計算が実行できます。放出されるガンマ線のスペクトル計算には、放射性核種崩壊データベース DECDC[28]が使用されます。なお、このデータベースは ICRP107 と同等です。 この機能を使う場合は、崩壊データファイル RIsource.dat が置かれたフォルダを [parameters] セクションにおいて file(24) (D=c:\phits\data\) により指定する必要があります。 e-type= 28 の場合、崩壊データベースにより計算されたガンマ線強度にしたがって、各エネルギーの粒子を生成します。 e-type= 29 の場合、崩壊データベースにより計算された全てのエネルギー点に同数の粒子を生成させ、そのガンマ線強度にしたがって、各エネルギーのウエイトを変化させます。
ni=	放射性核種の数。放射性核種とその放射能は次の様なフォーマットで指定する。 (RI(i),A(i),i=1,ni) ただし、核種 RI(i) は 137Cs あるいは Cs-137 の書式で指定する。 放射能 A(i) は Bq（ベクレル）単位で指定する。
dtime=	(D=−10.0) 時間経過の指定オプション dtime>0（単位は sec）: 指定した時間が経過した時点の指定核種及びその娘核種の放射能による崩壊ガンマ線を設定します。各 RI の放射能は、dtime 経過後の値に減衰します。例えば、半減期 1 分の RI を 100Bq で指定し、dtime=60.0 と設定した場合、その RI の放射能は 50Bq と見なされます。 dtime=0: 指定した放射能（時間経過なし）による崩壊ガンマ線を設定します。 dtime<0: 指定した放射性核種の半減期 ×|dtime| が経過した時点の指定核種及びその娘核種の放射能による崩壊ガンマ線を設定します。ただし、dtime>0 の場合と異なり、放射能の絶対値は減衰しません。例えば、半減期 1 分の RI を 100Bq で指定し、dtime=−1.0 と設定した場合、その RI の放射能は 100Bq のまま娘核種の崩壊を考慮可能です。放射平衡まで達する時間が分からない場合は、dtime=−10.0 のように長めの値を設定しておくと便利です。dtime<0 とする場合、親核と娘核（例えば Cs-137 と Ba-137m）を同時に指定することはできません。

マルチソースでこの機能を使用する際は、<source>=1.0 とし totfact を以下のように指定してください。

- マルチソースの全てで e-type = 28,29 を使用する場合は、<source>の個数を入力する。
- マルチソースで e-type = 28,29 以外を含む場合は、各<source>が絶対強度となるように設定し、すべての<source>の和を入力する。

[28] A. Endo, Y. Yamaguchi and K.F. Eckerman, Nuclear decay data for dosimetry calculation - Revised data of ICRP Publication 38, JAERI 1347 (2005).

表 56: ソースエネルギー分布パラメータ (7)

パラメータ	説明
	（e-type= 28,29 の場合の続き）
actlow=	(D=1.0e-10) 放射能の下限の指定オプション (Bq) 指定した時間経過後の放射能がこの値より小さい場合、その放射性核種の崩壊ガンマ線は線源に設定されません。
norm=	(D=0) 線源強度の規格化オプション 0: 1 秒あたり (/sec) の量を出力するように規格化します。その際、線源強度は、Bq 単位で指定した放射能強度（dtime>0 の場合はその崩壊も考慮した後）とします。 1: 崩壊ガンマ線を 1 つ放出するあたりの量を出力するように規格化します。（通常の PHITS の規格化方法に準拠）

e-type = 28,29 の場合の入力例は以下のようになります。

```
[ S o u r c e ]
 totfact = 2.0
 <source> = 1.0
   s-type =   4
     proj =  photon
      dir =   all
       r0 =   0.
       z0 =   0.
       z1 =   0.
   e-type =   28
       ni =   1
            Cs-137  100.
    dtime =  -10.0
   actlow =   1.0
 <source> = 1.0
   s-type =   4
     proj =  photon
      dir =   all
       r0 =   0.
       z0 =   0.
       z1 =   0.
   e-type =   28
       ni =   1
            Cs-134  100.
    dtime =  -10.0
   actlow =   1.0
```

ここでは、Cs-137 と Cs-134 が共に 100Bq ずつ存在し、それらが放射平衡になった状況を線源として指定しています。また、actlow を 1(Bq) することで、この数値以下の放射能の影響を無視しています。

4.3.17 角分布の定義

dir = data を指定した場合は、次のような a-type =で始まる角分布サブセクションが必要です。(D=***) のあるものは、省略可能です。

表 57: 角分布パラメータ (1)

パラメータ	説明
a-type = 1, (11)	角度分点 a(i) と各ビンにおけるソース粒子の生成確率 w(i) を与えることにより、任意の角度分布を指定する。 分点は、1 の時は cos で、11 の時は degree で与える。 各ビンに生成される積分粒子数は、w(i) に比例する。
na =	角度群数。データは自由フォーマットで次の様に与える。 (a(i),w(i),i=1,|na|), a(|na|+1)
a-type = 4, (14)	a-type = 1, (11) と同じ角度分布を発生させる。ただし、 a-type = 1, (11) が生成数を調整して角度分布を表現するのに 対し、a-type = 4, (14) では、全ての角度ビンに同数の粒子を 生成させ、粒子のウエイトの積分値が w(i) となるように 角度分布を表現する。 分点は、4 の時は cos で、14 の時は degree で与える。 各ビンに生成される積分粒子数は、q(i) に比例する。
na =	角度群数。データは自由フォーマットで次の様に与える。 (a(i),w(i),i=1,|na|), a(|na|+1) デフォルト（q-type = 0）では各ビンに等しい個数が生成される。 q-type = 1 で q(i) を設定した場合は、それらの値に各ビンの生成粒子数の積分値は比例する。
q-type = 0, 1	(D=0) 生成個数のオプション for 0, 全てのビンで q(i)=1、以下のデータは無し for 1, 各ビンの生成個数比 q(i) を次の行からデータで与える、(q(i),i=1,na)

表 58: 角度分布パラメータ (2)

パラメータ	説明
a-type = 5, (15) g(x) =	角度分布 (dφ/dΩ) を任意の関数 g(x) で与える。 分点は、5 の時は cos で与え、15 の時は degree で与える。 Fortran 形式で書いた関数。x は角度を表す。 内部変数や定数が使えます。例）g(x) = exp(-c1*x**2)
nn =	角度群数。
ag1 = ag2 =	角度分布のカットオフ最小値 角度分布のカットオフ最大値
a-type = 6, (16) g(x) =	a-type = 5, (15) と同じ角度分布を発生させる。 ただし、a-type = 5, (15) が生成数を調整して角度分布を表現するのに対し、a-type = 6, (16) では、全ての角度ビンに同数の粒子を生成させ、粒子のウエイトを任意の関数 $g(x)$ に比例して変化させることにより角度分布を表現する。 分点は、6 の時は cos で与え、16 の時は degree で与える。 Fortran 形式で書いた関数。x は角度を表す。 内部変数や定数が使えます。例）g(x) = exp(-c1*x**2)
nn =	角度群数。 デフォルト（q-type = 0）では各ビンに等しい個数が生成される。q-type = 1 で q(i) を設定した場合は、それらの値に各ビンの生成粒子数の積分値は比例する。
ag1 = ag2 =	角度分布のカットオフ最小値 角度分布のカットオフ最大値
q-type = 0, 1	(D=0) 生成個数のオプション for 0, 全てのビンで a(i)=1、以下のデータは無し for 1, 各ビンの生成個数比 a(i) を次の行からデータで与える、 (q(i),i=1,nm)

4.3.18 時間分布の定義

`t-type =`で始まる時間分布サブセクションを指定することにより、線源に時間分布をもたせることができます。`t-type =1,2` はそれぞれ矩形分布と Gauss 分布、`t-type =3,4` は時間分布を分点と強度分布で与えます。`t-type =5,6` は時間に関する強度分布を任意の関数で与えます。また、`t-type =100` では、エネルギーの関数として時間分布の型を指定できます。(D=***) のあるものは、省略可能です。

表 59: 時間分布パラメータ (1)

パラメータ	説明
`t-type = 0, 1, 2`	(D=0) 時間分布
	0: 時間分布なし。t=0.0
	1: 矩形分布
	2: Gauss 分布
`t0 =`	(D=0.0) 第 1 の時間分布の中心時刻 (ns)
`tw =`	時間分布の幅 (ns)。矩形分布の場合は全幅。Gauss 分布の場合は半値全幅。
`tn =`	時間分布の数
`td =`	時間分布の間隔 (ns)
`tc =`	(D=10×tw) Gauss 分布の場合の cut off 時間 (ns)

表 60: 時間分布パラメータ (2)

パラメータ	説明
t-type = 3	時間分点 t(i) と各ビンにおけるソース粒子の生成確率の積分値 w(i) を与えることにより、任意の時間分を指定する。統計的に w(i) に比例するように各ビンに生成する粒子の数を調整し、時間分布を表現する。
ntt =	時間群数。データは自由フォーマットで次のように与える。 (t(i),w(i),i=1,ntt), t(ntt+1) 各ビンに生成される粒子数の積分値は w(i) に比例する。
t-type = 4	t-type=3 と同じ時間分布を発生させる。ただし、t-type=3 が生成数を調整して時間分布を表現するのに対し、t-type=4 では、全ての時間ビンに同数の粒子を生成させ、粒子のウエイトの積分値を w(i) に比例して変化させることにより時間分布を表現する。また、o-type=1 とし生成個数比 o(i) を与えることにより、各ビンの生成個数を変化させ、特定の時刻をもつソースの統計量を変化させることが出来る。
ntt =	時間群数。データは自由フォーマットで次のように与える。 (t(i),w(i),i=1,ntt), t(ntt+1) デフォルト (o-type=0) では各ビンに等しい個数が生成される。 o-type=1 で o(i) を設定した場合は、それらの値に各ビンの生成粒子数の積分値は比例する。
o-type = 0, 1	(D=0) 生成個数のオプション for 0, 全てのビンで o(i)=1、以下のデータは無し for 1, 各ビンの生成個数比 o(i) を次の行からデータで与える (o(i),i=1,ntt)

表 61: 時間分布パラメータ (3)

パラメータ	説明
`t-type = 5`	微分時間スペクトル ($d\varphi/dt$) を任意の関数 $h(t)$ で与える。
$h(x)$	Fortran 形式で書いた関数。x は時刻 (ns) を表す。内部変数や定数が使える。
ll	時刻群数
tg1	時間分布のカットオフ最小値 (ns)
tg2	時間分布のカットオフ最大値 (ns)
`t-type = 6`	`t-type=5` と同じ時間分布を発生させる。ただし、`t-type=5` が生成数を調整して時間分布を表現するのに対し、`t-type=6` では、全ての時間ビンに同数の粒子を生成させ、粒子のウエイトの積分値を `w(i)` に比例して変化させることにより時間分布を表現する。 また、`o-type=1` とし生成個数比 `o(i)` を与えることにより、各ビンの生成個数を変化させ、特定の時刻をもつソースの統計量を変化させることが出来る。
$h(x)$	Fortran 形式で書いた関数。x は時刻 (ns) を表す。内部変数や定数が使える。
ll	時刻群数 デフォルト (`o-type=0`) では各ビンに等しい個数が生成される。`o-type=1` で `o(i)` を設定した場合は、それらの値に各ビンの生成粒子数の積分値は比例する。
tg1	時間分布のカットオフ最小値 (ns)
tg2	時間分布のカットオフ最大値 (ns)
`o-type = 0, 1`	(D=0) 生成個数のオプション for 0, 全てのビンで `o(i)=1`、以下のデータは無し for 1, 各ビンの生成個数比 `o(i)` を次の行からデータで与える `(o(i),i=1,ntt)`

`t-type =100` を指定し、PHIITS のソースファイルを書き換えることで、任意の時間分布を与えることができます。この機能を使用する場合は、ソースファイルのひとつ sors.f ファイルにある subroutine tdis01 を書き換えてください。

表 62: 時間分布パラメータ (4)

パラメータ	説明
`t-type = 100`	エネルギーの関数として時間分布を与える。 詳細は、ソースプログラムに書き込んで実現する。
tg1	時間分布のカットオフ最小値 (ns)
tg2	時間分布のカットオフ最大値 (ns)

4.3.19 マルチソースの例題

マルチソースの例題を紹介する。この例題には、関数を用いたエネルギー分布や、角度分布の例も含まれている。まず、ソースセクションのリストを示す。

List 4.2 ● マルチソース例題

```
[ S o u r c e ]
 totfact = 3
<source> = 9.72
  s-type = 4
    proj = proton
      z0 = 2
      z1 = 29
      r0 = 5
      r1 = 4
     dir = 0.0
  e-type = 6
     eg1 = 1.e-6
     eg2 = 1.e-3
     nm  = -200
    set: c10[1.e-4]
    f(x) = x**(1.5)*exp(-x/c10)
<source> = 1
  s-type = 4
    proj = photon
      z0 = 1
      z1 = 2
      r0 = 5
     dir = -1
  e-type = 5
     eg1 = 1.e-3
     eg2 = 5.e-1
     nm  = 200
     set: c10[1.e-1]
     set: c20[1.e-1/2.35482]
    f(x) = exp(-(x-c10)**2/2/c20**2)
<source> = 1
  s-type = 4
    proj = neutron
      z0 = 29
      z1 = 30
      r0 = 5
  e-type = 6
     eg1 = 1.e-2
     eg2 = 1.e+3
     nm  = -200
     set: c10[92.469]
     set: c20[5.644e+10]
    f(x) = c10/c20*exp(-sqrt(x*(x+1876))/c10)*(x+938)/sqrt(x*(x+1876))
     dir = data
  a-type = 5
     ag1 = 0
     ag2 = 1
     nn  = 200
    g(x) = exp(-(x-1)**2/0.3**2)
```

この例題には、3 つの<source>で始まるソースが含まれる。最初のソースは、円柱で z が 2cm から 29cm で、半径が 5cm、ただし、r1=4 が定義してあるので、半径 4cm の内側は含まれていない。中空の円柱ソースである。次のソースは、やはり円柱で半径 5cm、z が 1cm から 2cm までの厚さが 1cm の薄い円柱である。最後のソースは、前のと同じ薄い円柱で、z 座標が 29cm から 30cm までである。各ソースの<source>に定義される、各ソースの相対比である。ここでは、各ソースの体積比になっている。従って、このマルチソースでは、定義されるソース領域に均一に粒子が生成される。この座標分布を [t-product] で output=source, icntl=6 で計算したものを以下に示す。このソースは、円柱の表面、厚さ 1cm の領域を定義している。

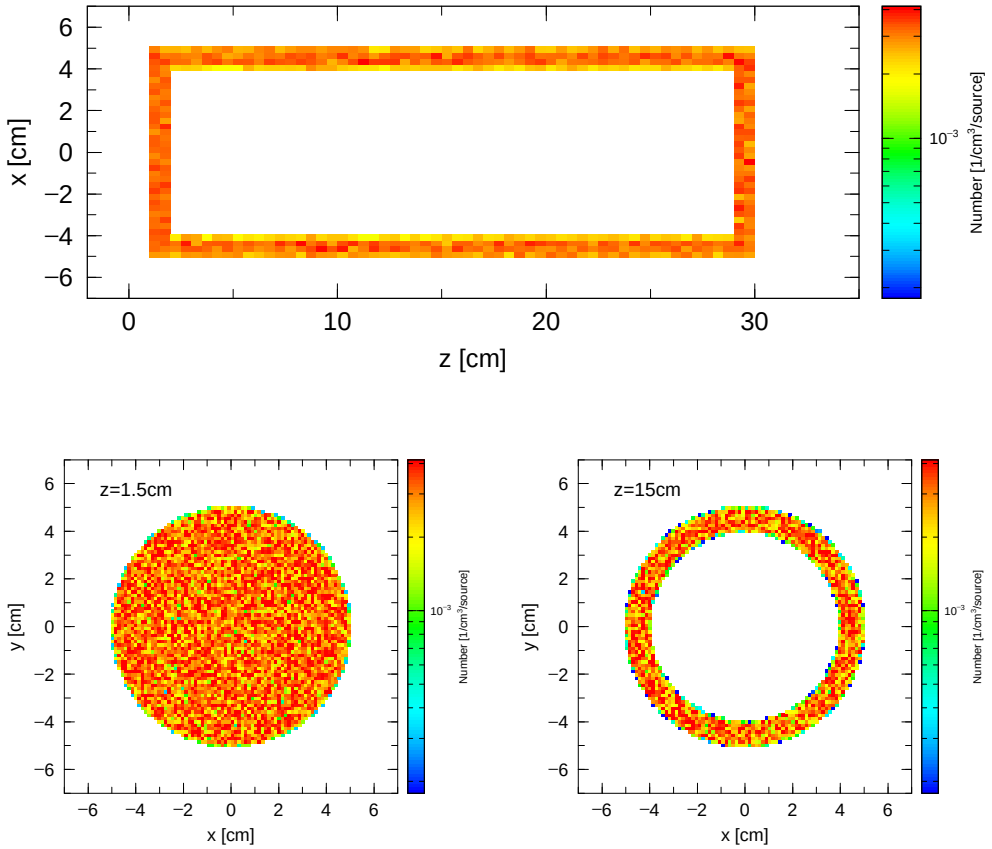


図 10: マルチソース、空間分布

次に、3 つのソースは、ソース粒子が、陽子、光子、そして中性子である。それぞれエネルギー分布が関数で定義してある。最初のものは、Maxwellian 分布、次がガウス分布、最後が任意の関数である。最初の Maxwellian 分布は、e-type=7 の

```
e-type = 7
   et0 = 1.e-4
   et1 = 1.e-6
   et2 = 1.e-3
```

と同義である。また、2 番目のガウス分布は、e-type=2 の

```
e-type = 2
   eg0 = 1.e-1
   eg1 = 1.e-1
   eg2 = 1.e-4
   eg3 = 5.e-1
```

と同義である。

これらのエネルギー分布を、やはり、[t-product] で output=source, icntl=6 で計算したものを以下に示す。結果は、粒子ごとにプロットしているので、各ソースごとのエネルギー分布が色別に表されている。

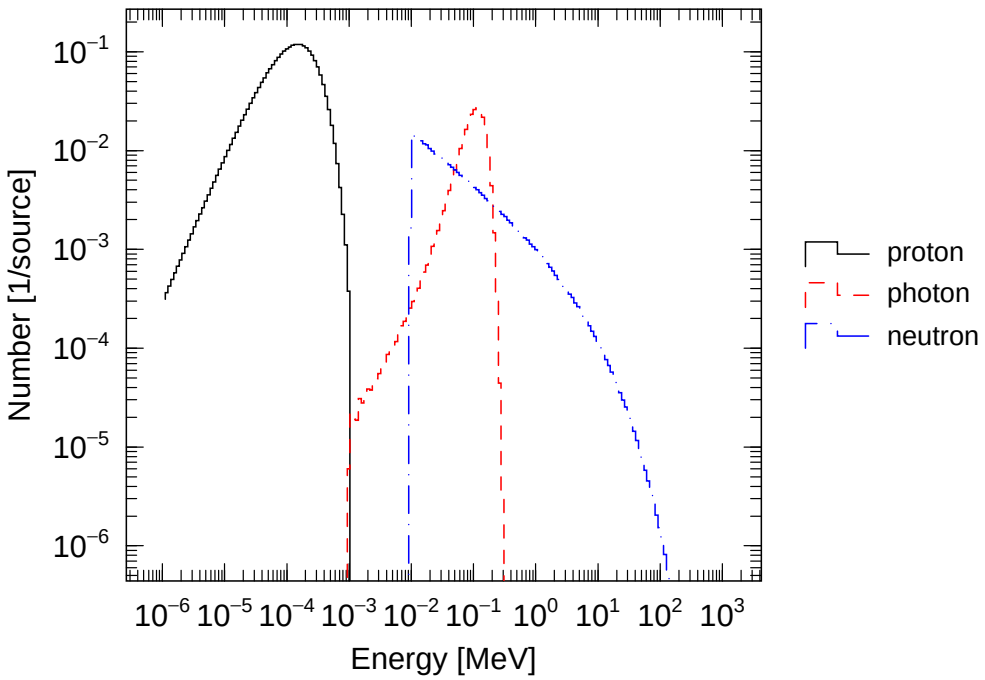


図 11: マルチソース、エネルギー分布

最初のソースは、dir=0, 即ち 90 度方向、2 番目が dir=-1, 180 度方向、3 番目が dir=data で角度分布を持つ。ここでは、0 度を中心としたガウス分布が関数形で定義されている。この結果は、[t-cross] を用いて次のように表される。

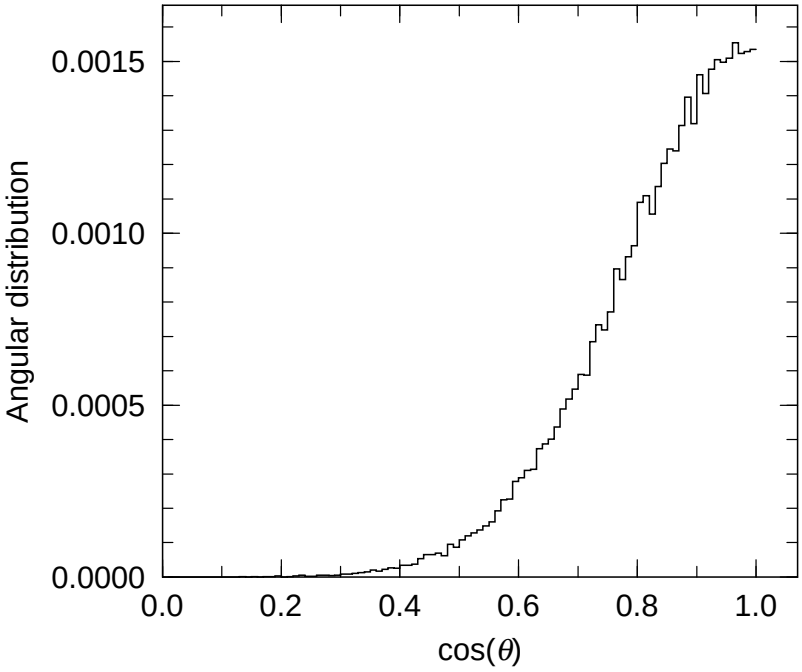


図 12: マルチソース、角度分布

4.3.20 ダクトソースオプション

中性子分光施設の非常に長い中性子導管を扱うために、特別なソースを生成するダクトソースオプションを作りました。等方的なソースを仮定すると、長いビームラインの中のカレントは、立体角の関係からソースの位置からの距離の二乗に反比例します。このことは、ビームラインの壁にぶつかるカレントは、”wall current”と呼びますが、例えば 100m 位置で、1m 位置に比べて 6 桁小さくなります。このような状況での計算の統計を上げるために、粒子のウエイトを変化させて、シミュレーション粒子の wall current が、ビームラインのどの場所でも同じになるようなオプションを設けました。

このダクトソースオプションは、`s-type = 1, 4, 2, 5,` の円柱と角柱のソースに `dom = -10` を指定することにより適応されます。各パラメーターとその意味を下にまとめます。

表 63: ダクトソースオプションのパラメータ

パラメータ	説明
`dom = -10`	ダクトソースの指定
`dl0 =`	ビームラインの始点、z0 からの距離 (cm)
`dl1 =`	ダクトソースの始点、z0 からの距離 (cm)
`dl2 =`	ダクトソースの終点、z0 からの距離 (cm)
`dpf =`	dl2 を通過する粒子の割合
`drd =`	s-type = 1, 4 円柱の時のビームラインの半径 (cm)
`dxw =`	s-type = 2, 5 角柱の時のビームラインの X サイズ (cm)
`dyw =`	s-type = 2, 5 角柱の時のビームラインの Y サイズ (cm)

ビームラインの形状は、`s-type = 1, 4` の時は、円柱、`s-type = 2, 5` の時は、角柱を仮定しています。また、`z1 = z0` と `dir = 1` も仮定しています。後者は、ビームラインの方向です。これを変更する時は、`trcl` を用いてください。ソース粒子は、`z0` 地点で、`s-type = 1, 4` の時は、`r0` の内部、もしくは、`s-type = 2, 5` の時は、`x0, x1, y0, y1` の内部で生成されます。

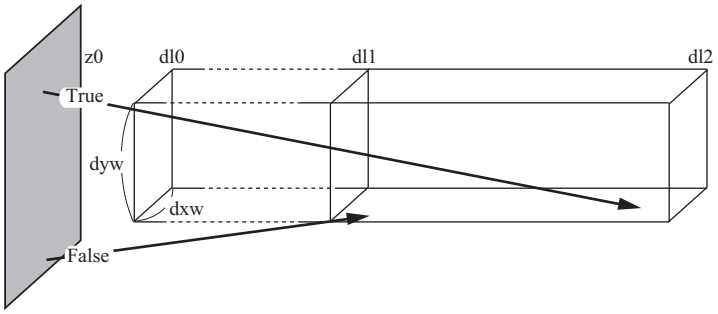


図 13: 角柱のダクトソースのイメージ図

ソース粒子の方向は、ダクトに当たる位置から決定されます。`dl1` から `dl2` の間のダクトに粒子が均一に当たるように壁の位置が決められます。そのとき、等方分布を仮定して粒子のウエイトが決められます。全体の規格化定数は、`z0` 地点で `dl0` 地点のダクト面積と同じ領域で発生した粒子が `dl0` 地点のダクトを通過する粒子数で規格化されます。通常、`dl0` から `dl2` のダクト壁から全てのソース領域が見える時、この規格化定数は 1 です。もし、`z0` 地点でのソース領域が、`dl0` 地点のダクト面積より大きい場合は、`z0` 地点でダクト面積より大きい領域からダクトに入った粒子は、ソース粒子の規格化には含まれません。このことは、`z0` 地点でダクト面積より大きい領域からダクトに入った粒子は、規格化定数を変えずにダクト管内のカレ

ントを増加させます。このダクトソースでは、粒子のソース位置での角分布は、ビームラインの方向のダクト面積の立体角で均一であるということを仮定しています。

次に、簡単な例題を示します。最初の例は、角柱のビームラインを仮定し、ソースサイズとガイド断面積は同じとしています。ダクトソースのセクションは、

List 4.3 ● ダクトソースオプション例題 1

```
[ S o u r c e ]

set: c1[200]    $dl0
set: c2[500]    $dl1
set: c3[5000]   $dl2
set: c4[5.0]    $x*2 at z0
set: c5[5.0]    $y*2 at z0
set: c10[5.0]   $dxw
set: c20[5.0]   $dyw
set: c30[0.001] $dpf

s-type = 2
proj   =  neutron
e0     =  20.0
x0     = -c4/2
x1     =  c4/2
y0     = -c5/2
y1     =  c5/2
z0     =  0.0
z1     =  0.0
dir    =  1.0
phi    =  0.0

dom = -10
dl0 = c1
dl1 = c2
dl2 = c3
dxw = c10
dyw = c20
dpf = c30
```

最初のパートでこのダクトソースに必要な定数、`dl0`, `dl1`, `dl2` と `dxw`, `dyw`, `dpf` を定義しています。次のパートで、ソースの位置、エネルギーを定義しています。最後のパートで、ダクトソースのパラメーターを定義しています。ここでは、5m から 50m のビームラインの中の current と wall current を [t-cross] タリーを用いて計算しています。結果を右の図に示します。同じ図に、理想的な場合の解析的な解、つまり、current が $1/L^2$ に比例し、wall current が $1/L^3$ に比例するグラフも描いてあります。十字の印が、`dl0` の位置です。ここの current が 1 です。ダクトソースオプションの結果は、理想的な場合と完全に一致しています。

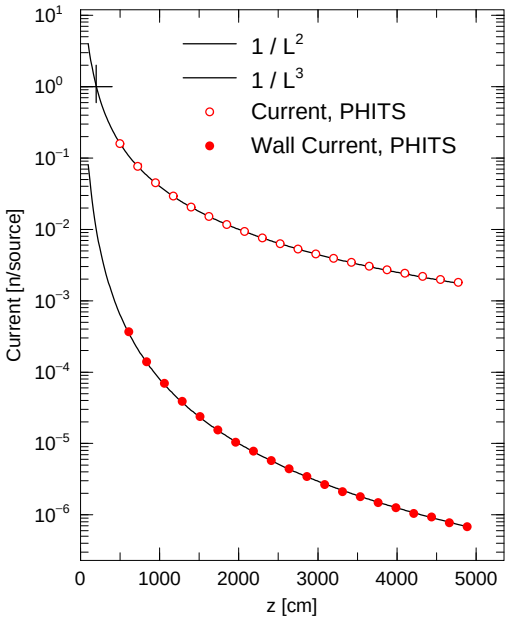


図 14: ダクトソースオプション例題 1

次の例題は、ソース領域の大きさを変えたものです。

List 4.4 ● ダクトソースオプション例題 2

```
[ S o u r c e ]

set: c1[200]    $dl0
set: c2[500]    $dl1
set: c3[5000]   $dl2
set: c4[10.0]   $x*2 at z0
set: c5[10.0]   $y*2 at z0
set: c10[5.0]   $dxw
set: c20[5.0]   $dyw
set: c30[0.001] $dpf

s-type = 2
proj   =  neutron
e0     =  20.0
x0     = -c4/2
x1     =  c4/2
y0     = -c5/2
y1     =  c5/2
z0     =  0.0
z1     =  0.0
dir    =  1.0
phi    =  0.0

dom = -10
dl0 = c1
dl1 = c2
dl2 = c3
dxw = c10
dyw = c20
dpf = c30
```

次のグラフが示すように、この例題では、current、wall current ともに増加しています。この増加分がソース領域のダクトより大きい部分からの寄与を示します。従って、このオプションによって、ソースのマージン領域からの寄与を自動的に扱うことができます。

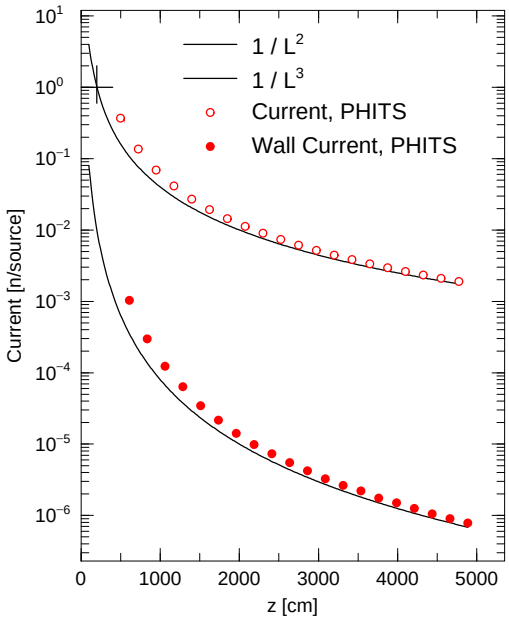


図 15: ダクトソースオプション例題 2

4.4 [Material] セクション

4.4.1 書式

このセクションでは、体系を構成する物質を定義します。書式は次の 2 種類があります。

最初は、従来の書式で、

```
[ M a t e r i a l ]
   MAT[ m ]
         nucleus    density
         nucleus    density
         nucleus    density
         nucleus    density
   MAT[ m ]
         keyword=value  .....
         nucleus    density
         nucleus    density
         MTm    S(a,b) identifier ....
   MAT[ m ]
            ......     .......
            ......     .......
```

この書式に加えて以下の入力ファイルの物質定義の書式も使えます。

```
[ M a t e r i a l ]
   Mm    nucleus    density        nucleus    density
         nucleus    density        nucleus    density
   Mm
         keyword=value  .....
         nucleus    density        nucleus    density
         nucleus    density        nucleus    density
         MTm    S(a,b) identifier ....
   Mm
           ......     .......
           ......     .......
```

ここで、*m* は、物質番号で重ならない限り 99999 まで自由です。

4.4.2 核種の定義

nucleus は、核種の定義で、`208Pb, Pb-208, 82208` どの形式でも可能です。水素は、`1H, H-1, 1001` と指定します。質量数をゼロにすると、即ち、`Pb, 82000` とすると、その原子の天然同位体比を仮定します。ただし、この仮定は、`dmax` を設定して核データライブラリーを使用する場合、天然同位体に関する情報が含まれていない限り使用できません。

核データを用いる時は、核種の記述の後のピリオドに続け、ライブラリーの番号 (2 桁の数字)、データのクラス (1 文字のアルファベット) で指定します。例えば、`208Pb.24c, Pb-208.24c, 82208.24c` です。指定しない場合は、断面積ディレクトリファイル `file(7) (D=xsdir)` 中のリストを上から検索し、その核種で最初に該当したライブラリーを使用します。核種毎ではなく、物質番号毎にデフォルトのライブラリーを指定したい場合は、表 64 の `NLIB, PLIB, ELIB, HLIB` を利用してください。ただし、これらで指定していても、上記の書式で核種毎に示してあるライブラリーが優先されますのでご注意ください。どの核データが使用されたのかを知りたい場合は、`[parameters]` セクションの `kmout` を 1 にすると、サマリーの出力ファイル `file(6) (D=phits.out)` にその情報が書き出されます。

炭素を定義するとき、“C” で始まるとコメント行とされてしまいます。“6000”, “6012” または “12C” で指定するようにしてください。

4.4.3 密度の定義

density は密度の定義で、単位は、*density* が正の時は、粒子密度 [10^{24} atoms/cm^3]、負の時は、質量密度 [g/cm^3] です。これらの密度は、`[cell]` セクションで密度が 定義されていれば、それらによって再規格化されます。従って、`[cell]` セクションで密度が定義されている時は、このセクションでの定義は、密度である必要はなく粒子数の比でもかまいません。

4.4.4 物質パラメータ

核データを使う領域に関して、物質毎のパラメータを *keyword=value* の形で指定します。挿入場所は、その物質のサブセクション中なら自由です。指定できるパラメータとその意味は以下の通りです。

表 64: 物質パラメータ

パラメータ	値	説明
GAS	D = 0	電子の阻止能の密度効果
	= 0	液体もしくは固体の計算に適する
	= 1	気体の計算に適する
ESTEP	= *n*	電子輸送のサブステップの数を *n* にする
		n が build-in のデフォルト値より小さい時は無視される
NLIB	= *id*	デフォルトの中性子ライブラリー番号を *id* にする
PLIB	= *id*	デフォルトの光子ライブラリー番号を *id* にする
ELIB	= *id*	デフォルトの電子ライブラリー番号を *id* にする
HLIB	= *id*	デフォルトの陽子ライブラリー番号を *id* にする
COND		伝導体設定
	< 0	絶縁体
	= 0	(Default) 1 個でも絶縁体があれば絶縁体、それ以外は伝導体
	> 0	1 個でも伝導体があれば伝導体

4.4.5 $S(\alpha,\beta)$ の指定

低エネルギー中性子の輸送の際、$S(\alpha,\beta)$ のライブラリーが必要になる場合があります。$S(\alpha,\beta)$ ライブラリーは

```
MTm 物質 ID
```

で指定できます。m は、その物質の物質番号です。物質 ID は、xsdir に書かれている ID 番号（lwtr.20t など）で、各データの物質や温度情報は \XS\tsl\tsl-table に書かれていますので、そちらをご参照下さい。

4.4.6 例題

幾つかの例題を下に示します。

List 4.5 ● material example (1)

```
 1:  [ M a t e r i a l ]
 2:  MAT[ 1 ]
 3:     1H        1.0000000E-04
 4:   208Pb       1.7238000E-02
 5:   204Pb       4.6801000E-04
 6:   206Pb       7.9430000E-03
 7:   207Pb       7.2838000E-03
 8:  MAT[ 2 ]
 9:     1H        1.0000000E-09
10:    14N        4.6801000E-05
11:    16O        7.9430000E-06
```

核種と密度の順番は、核種-密度がデフォルトで、これを逆にしたいときは、

List 4.6 ● material example (2)

```
 1:  [ M a t e r i a l ]
 2:     den            nuc      <------
 3:  MAT[ 1 ]
 4:  1.0000000E-04      1H
 5:  1.7238000E-02    208Pb
 6:  4.6801000E-04    204Pb
 7:  7.9430000E-03    206Pb
 8:  7.2838000E-03    207Pb
 9:  MAT[ 2 ]
10:  1.0000000E-09      1H
11:  4.6801000E-05     14N
12:  7.9430000E-06     16O
```

のように、den nuc を入れて定義します。

List 4.7 ● material example (3)

```
 1:  [ M a t e r i a l ]
 2:  m1     80196.49c  5.9595d-5
 3:         80198.49c  3.9611d-3
 4:         80199.49c  6.7025d-3
 5:         80200.49c  9.1776d-3
 6:         80201.49c  5.2364d-3
 7:         80202.49c  1.1863d-2
 8:         80204.49c  2.2795d-3
 9:  c      ...Be...
10:  m3     4009.37c  1.2362E-1
11:  mt3    be.01
12:  c      ...h2o (25C)...
13:  m4     1001.37c  6.6658d-2      8016.37c  3.3329d-2
14:  mt4    lwtr.01
15:  c      ...b4c (natural boron; 25%-density)...
16:  m5     6012.37c  6.8118d-3
17:         5011.37c  2.1825d-2
18:  c      ...liquid-h2 (20K)...
19:  m6     1001.49c  3.1371d-2      1011.49c  1.0457d-2
20:  mt6    orthoh.00   parah.00
```

4.5 [Surface] セクション

4.5.1 書式

このセクションでは、入力ファイルの面 (surface) の定義を行います。コメント文字として C も使えます。ただし、ただし、C 以外のコメント文字としては $ のみ用いることができます。ファイルのインクルード文、定数定義などは、セクションの途中でも用いることが出来ます。継続行には、先頭に 5 桁の空白が必要です。

書式は、面番号、座標変換番号、面記号、面定義数値の順番です。面定義数値では定数や数式が使えます。マクロボディーによる面の定義も可能です。面番号は *PHITS* の制約で 1 ~ 999999 までです。MCNP と同様、面番号の前に “*” をつけると全反射面、“+” をつけると白色反射面を定義できます。

書式と例題は以下の通りです。

```
[ S u r f a c e ]
面番号      座標変換番号      面記号      面定義数値
```

表 65: 面記述書式

項目	説明
面番号	1 ~ 999999
座標変換番号	座標変換無しなら入力しない。 座標変換がある場合は、[transform] セクションの TRn の番号 n
面記号	面カード表の面記号もしくはマクロボディーの記号
面定義数値	面により 1 個から 15 個の入力値

List 4.8 ● [surface] セクションの例題 (1)

```
 1:   [surface]
 2:     1   cz     5.0
 3:     2   cz    10.0
 4:     3   cz    15.0
 5:     4   cz    20.0
 6:     5   pz     0.0
 7:     6   pz     5.0
 8:     7   pz    10.0
 9:     8   pz    15.0
10:     9   pz    55.0
11:    10   pz    60.0
```

表 66: 面カード

面記号	型式	解説	方程式	入力数値
P	平面	汎用	$Ax + By + Cz - D = 0$	$A\ B\ C\ D$
PX		X 軸に垂直	$x - D = 0$	D
PY		Y 軸に垂直	$y - D = 0$	D
PZ		Z 軸に垂直	$z - D = 0$	D
SO	球	中心は原点	$x^2 + y^2 + z^2 - R^2 = 0$	R
S		汎用	$(x - \bar{x})^2 + (y - \bar{y})^2 + (z - \bar{z})^2 - R^2 = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ R$
SX		中心が X 軸上	$(x - \bar{x})^2 + y^2 + z^2 - R^2 = 0$	$\bar{x}\ R$
SY		中心が Y 軸上	$x^2 + (y - \bar{y})^2 + z^2 - R^2 = 0$	$\bar{y}\ R$
SZ		中心が Z 軸上	$x^2 + y^2 + (z - \bar{z})^2 - R^2 = 0$	$\bar{z}\ R$
C/X	円柱	X 軸に平行	$(y - \bar{y})^2 + (z - \bar{z})^2 - R^2 = 0$	$\bar{y}\ \bar{z}\ R$
C/Y		Y 軸に平行	$(x - \bar{x})^2 + (z - \bar{z})^2 - R^2 = 0$	$\bar{x}\ \bar{z}\ R$
C/Z		Z 軸に平行	$(x - \bar{x})^2 + (y - \bar{y})^2 - R^2 = 0$	$\bar{x}\ \bar{y}\ R$
CX		X 軸上	$y^2 + z^2 - R^2 = 0$	R
CY		Y 軸上	$x^2 + z^2 - R^2 = 0$	R
CZ		Z 軸上	$x^2 + y^2 - R^2 = 0$	R
K/X	円錐	X 軸に平行	$\sqrt{(y - \bar{y})^2 + (z - \bar{z})^2} - t(x - \bar{x}) = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ t^2\ \pm 1$
K/Y		Y 軸に平行	$\sqrt{(x - \bar{x})^2 + (z - \bar{z})^2} - t(y - \bar{y}) = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ t^2\ \pm 1$
K/Z		Z 軸に平行	$\sqrt{(x - \bar{x})^2 + (y - \bar{y})^2} - t(z - \bar{z}) = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ t^2\ \pm 1$
KX		X 軸上	$\sqrt{y^2 + z^2} - t(x - \bar{x}) = 0$	$\bar{x}\ t^2\ \pm 1$
KY		Y 軸上	$\sqrt{x^2 + z^2} - t(y - \bar{y}) = 0$	$\bar{y}\ t^2\ \pm 1$
KZ		Z 軸上	$\sqrt{x^2 + y^2} - t(z - \bar{z}) = 0$	$\bar{z}\ t^2\ \pm 1$
				±1 は 1 シートの円錐のみ指定
SQ	楕円 双曲面 放物面	X, Y, または Z 軸に 平行な軸	$A(x - \bar{x})^2 + B(y - \bar{y})^2 + C(z - \bar{z})^2 +$ $2D(x - \bar{x}) + 2E(y - \bar{y}) + 2F(z - \bar{z}) + G = 0$	$A\ B\ C\ D\ E$ $F\ G\ \bar{x}\ \bar{y}\ \bar{z}$
GQ	円柱 円錐 楕円 双曲面 放物面	X, Y, または Z 軸に平行 でない軸	$Ax^2 + By^2 + Cz^2 + Dxy + Eyz +$ $Fzx + Gx + Hy + Jz + K = 0$	$A\ B\ C\ D\ E$ $F\ G\ H\ J\ K$
TX	楕円環体 円環体	X, Y, または Z 軸に 平行	$(x - \bar{x})^2/B^2 +$ $(\sqrt{(y - \bar{y})^2 + (z - \bar{z})^2} - A)^2/C^2 - 1 = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ A\ B\ C$
TY			$(y - \bar{y})^2/B^2 +$ $(\sqrt{(x - \bar{x})^2 + (z - \bar{z})^2} - A)^2/C^2 - 1 = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ A\ B\ C$
TZ			$(z - \bar{z})^2/B^2 +$ $(\sqrt{(x - \bar{x})^2 + (y - \bar{y})^2} - A)^2/C^2 - 1 = 0$	$\bar{x}\ \bar{y}\ \bar{z}\ A\ B\ C$
P	平面	座標により 定義された面		$x_1, y_1, z_1,\ x_2, y_2, z_2,$ x_3, y_3, z_3

円錐の場合、$\bar{x}, \bar{y}, \bar{z}$ 等で指定する頂点を中心に、各軸上に沿って上下に２つの円錐面が定義されます。±1 の部分の入力を、1 とすると軸の上側に広がる面のみを、−1 とすると軸の下側に広がる面のみを、何の値も与えないとその両方を対象とします。ただし、片方の円錐面しか使用しない場合、`[cell]` セクションで領域を定義する際に、円錐の頂点を通る平面でも囲むようにしてください。すなわち、円錐を領域として定義する場合は、円錐の頂点を通る平面、片方の円錐面、円錐の底面の 3 つが必要です。

面記号 P を用いて、3 つの座標を与えることにより平面を定義できます。この場合、座標原点を含む領域がマイナス側として定義されますので、`[cell]` セクションで使用する際はご注意ください。

楕円環帯や円環体を定義する TX,TY,TZ に対しては `[transform]` による座標変換が利用できませんのでご注意ください。

4.5.2 マクロボディー

表 67: マクロボディーカード

記号	型式	入力数値	解説
BOX	任意の BOX 平面 (全ての角が 90 度)	V_x V_y V_z $A1x$ $A1y$ $A1z$ $A2x$ $A2y$ $A2z$ $A3x$ $A3y$ $A3z$	基点の座標 基点からの第 1 面のベクトル 基点からの第 2 面のベクトル 基点からの第 3 面のベクトル
RPP	直方体 (各面が xyz に垂直)	X_{min} X_{max} Y_{min} Y_{max} Z_{min} Z_{max}	x 座標の最小値、最大値 y 座標の最小値、最大値 z 座標の最小値、最大値
SPH	球 (汎用球 S と同じ)	V_x V_y V_z R	中心座標 球の半径
RCC	円柱	V_x V_y V_z H_x H_y H_z R	底面の中心座標 底面の中心からの上面中心へのベクトル 円柱の半径
RHP or HEX	任意の 6 角柱 プリズム	$v1$ $v2$ $v3$ $h1$ $h2$ $h3$ $r1$ $r2$ $r3$ $s1$ $s2$ $s3$ $t1$ $t2$ $t3$	基点の座標 基点から高さベクトル 基点からの第 1 面へのベクトル 基点からの第 2 面へのベクトル 基点からの第 3 面へのベクトル
REC	楕円柱	V_x V_y V_z H_x H_y H_z V_{1x} V_{1y} V_{1z} V_{2x} V_{2y} V_{2z}	シリンダーの底面の中心の座標 シリンダーの高さベクトル ($\vec{H}$) $\vec{H}$ に垂直な楕円の長軸ベクトル ($\vec{V}_1$) $\vec{H}$ と $\vec{V}_1$ に垂直な楕円の短軸ベクトル ($\vec{V}_2$)
TRC	カットされた円錐	V_x V_y V_z H_x H_y H_z R_1 R_2	円錐の底面の中心の座標 円錐の高さベクトル 底面の半径 上面の半径
ELL	楕円体	$Rm > 0$ のとき、 V_{1x} V_{1y} V_{1z} V_{2x} V_{2y} V_{2z} Rm $Rm < 0$ のとき、 V_{1x} V_{1y} V_{1z} V_{2x} V_{2y} V_{2z} Rm	 第 1 焦点の座標 第 2 焦点の座標 長軸の半径 この場合，長軸を中心に回転して楕円体を作りますので， 長軸 < 短軸となっても問題ありません 楕円体の中心 長軸のベクトル 短軸の半径
WED	くさび形	V_x V_y V_z V_{1x} V_{1y} V_{1z} V_{2x} V_{2y} V_{2z} V_{3x} V_{3y} V_{3z}	頂点の座標 底面の三角形においてある頂点から別の頂点へのベクトル ($\vec{V}_1$) $\vec{V}_1$ の頂点とは別の頂点へのベクトル ($\vec{V}_2$) 高さベクトル ($\vec{V}_3$)

カットされた円錐 (TRC) を用いる際、上面の半径 (R_2) を 0 にしないよう注意してください。例えば、TRC を用いてカットされていない（頂点をもつ）円錐を定義する場合は、R_2 の値に小さな値を入れてください。

くさび形 WED は、その底面が直角三角形である必要があります。任意の三角形は直角三角形を組み合わせて作成してください。

List 4.9 ● [surface] セクションの例題 (2) [cell] セクション例題 (2) に対応

```
[surface]
   1   rpp -15 15 -5 5  0 60
   2   rpp  -5  5 -5 5  0 20
   4   rpp -15 15 -5 5  0 20
```

```
5:     5   rpp -20 20 -5 5  0 40
6:     6   rpp -20 20 -5 5  0 20
7:     7   rpp -20 20 -5 5 40 60
8:     3   c/y   0 10  4
```

4.5.3 マクロボディーの面定義

マクロボディーで定義した面をセル定義文で使うときは、マイナス記号がマクロボディーの内側、プラス記号が外側を表します。マクロボディーを構成するそれぞれの面もセル定義文で使うことができます。その時は、マクロボディー番号の次にドットを入れ、それぞれの面の番号を付けます。それぞれの面の番号は以下の通りです。

表 68: マクロボディー面番号

記号	面番号	解説
BOX	1	$A1x$ $A1y$ $A1z$ の終点に垂直な面
	2	$A1x$ $A1y$ $A1z$ の始点に垂直な面
	3	$A2x$ $A2y$ $A2z$ の終点に垂直な面
	4	$A2x$ $A2y$ $A2z$ の始点に垂直な面
	5	$A3x$ $A3y$ $A3z$ の終点に垂直な面
	6	$A3x$ $A3y$ $A3z$ の始点に垂直な面
RPP	1	$X_{\max}$ の面
	2	$X_{\min}$ の面
	3	$Y_{\max}$ の面
	4	$Y_{\min}$ の面
	5	$Z_{\max}$ の面
	6	$Z_{\min}$ の面
SPH		通常の面と同じ
RCC	1	半径 R の円柱の側面
	2	H_x H_y H_z の終点に垂直な面
	3	H_x H_y H_z の始点に垂直な面
RHP or HEX	1	$r1$ $r2$ $r3$ の終点に垂直な面
	2	面 1 の対面
	3	$s1$ $s2$ $s3$ の終点に垂直な面
	4	面 3 の対面
	5	$t1$ $t2$ $t3$ の終点に垂直な面
	6	面 5 の対面
	7	$h1$ $h2$ $h3$ の終点に垂直な面
	8	$h1$ $h2$ $h3$ の始点に垂直な面
REC	1	楕円のシリンダー面
	2	H_x H_y H_z の終点に垂直な面
	3	H_x H_y H_z の始点に垂直な面
TRC	1	円錐面
	2	H_x H_y H_z の終点に垂直な面
	3	H_x H_y H_z の始点に垂直な面
ELL		普通の面として定義される
WED	1	底面と上面の斜辺を含む面
	2	$\vec{V}_2$ と $\vec{V}_3$ を含む面
	3	$\vec{V}_1$ と $\vec{V}_3$ を含む面
	4	$\vec{V}_1$ と $\vec{V}_2$ 及び $\vec{V}_3$ の終点を含む面
	5	$\vec{V}_1$ と $\vec{V}_2$ 及び $\vec{V}_3$ の始点を含む面

4.6 [Cell] セクション

4.6.1 書式

このセクションでは、[surface] セクションで定義した面を使ってセル (小部屋) の定義を行います。これは GG(General Geometry) の考え方に基づいています。基本的にセルは閉じた空間として定義し、様々な形状のセルを組み合わせることにより、粒子を輸送する仮想空間を構築します。*PHITS* では、粒子が飛ぶことが可能な内部空間とそれ以外の空間を分けて定義する必要があり、後者を外部ボイド (真空) として明示的に指定しなければなりません。

コメント文字として C の他に$が使えます。このセクションではセルの定義に#を用いるため、*PHITS* の入力ファイルでのコメント文字#は利用できません。ファイルのインクルード文、定数定義などは、セクションの途中でも用いることが出来ます。継続行には、先頭に 5 桁の空白が必要です。

書式は以下の通りです。セル番号、物質番号、物質密度、セル定義文、キーワード形式のセルパラメータの順番で各セルを定義します。各項目の意味は表 69 を参照してください。

```
[ C e l l ]
セル番号      物質番号      物質密度      セル定義文      セルパラメータ
```

また、LIKE *n* BUT セルパラメータの表式や lattice 構造も使うことができます。似たような構造のセルを複数並べる場合に効率良く設定できます。利用方法の詳細は 4.6.5 節をご覧ください。セルパラメータの種類については表 70 にまとめています。

表 69: セル記述書式

項目	説明
セル番号	1 ~ 999999 まで使用できます。
物質番号	[material] セクションで指定した物質番号を入力します。ただし、そのセルが内部空間にあるボイド (真空) のときは 0 を、外部ボイドのときは −1 を指定します。
物質密度	そのセルに含まれる物質の密度を与えます。単位は、[material] セクションで定義した density が正の場合は粒子密度 (10^{24} atoms/cm^3)、負の場合は質量密度 (g/cm^3) となります。ここで指定した原子数密度が計算に使用されますが、物質の組成比は [material] セクションで指定した密度を基に与えられます。従って、同じ組成で密度の異なる物質を定義可能となります。その際に、異なる物質番号を付加するパラメータ matadd を加えました。 セルがボイドのとき、すなわち物質番号が 0 か −1 の場合はこの項目は入力しません。
セル定義文	[surface] セクションで定義された面の面番号や集合代数の演算子を用いてセルの幾何形状を記述します。演算子の記号は ⊔ (空白), :, #でそれぞれ積 (and), 和 (or), 否定 (not) を表します。(,) も使用できます。記述方法は 4.6.2 節をご参照ください。
LIKE *n* BUT	セルの内容がセル番号 *n* とほとんど同じで、BUT 以下に記述されるセルパラメータの値だけ違うセルを定義します。
セルパラメータ	書式は keyword=value です。キーワードとして VOL(体積), TMP(温度), TRCL (transform), U(universe), LAT(lattice), FILL が使用できます。また LIKE *n* BUT の場合は、これらに加えて MAT(物質番号), RHO (密度) が使えます。各キーワードの意味は表 70 を参照してください。

同じ物質番号を指定して別の密度をもつセルを複数定義した場合はご注意ください。2 番目以降のセル番号の物質番号として別の数字が割り振られます。

表 70: セルパラメータ

項目	説明
VOL	そのセルの体積 (cm^3) を与えます。
TMP	そのセルの物質がもつ温度 (MeV) を与えます。
TRCL	そのセルを座標変換します。書式は [transform] セクションと同じです。
U	Universe 番号。1 ~ 999999 まで使用できます。Universe 構造を利用する場合に定義します。利用方法は 4.6.3 節を参照してください。
LAT	Lattice 番号。LAT=1 の場合は四角柱、LAT=2 の場合は六角柱を定義します。これらを単位とした繰り返し構造 (lattice 構造) を利用する場合に指定します。利用方法は 4.6.4 節を参照してください。
FILL	U で定義した universe 番号を入力し、そのセルを U で定義した宇宙 (universe) で満たします。利用方法は 4.6.3, 4.6.4 節を参照してください。
MAT	LIKE *n* BUT MAT=*m* の表式で用います。セル番号 *n* とは物質番号が違うセルを定義できます。
RHO	LIKE *n* BUT RHO=*x* の表式で用います。セル番号 *n* とは密度が違うセルを定義できます。密度の単位はセル番号 *n* と同じです。

4.6.2 セルの記述方法

任意の形状の構造物を仮想空間に構築する場合、最初にその構造物の表面を座標空間において定義し、次に対応する領域をその面で閉じることによってセル (小部屋) をつくる、という方法があります。この方法は General Geometry(GG) の考え方に基づいており、面の表側と裏側を区別して表現し、更に集合代数の演算子を利用することによって各セルを定義します。

考えている領域が定義した面の表側と裏側のどちらに属するかを区別するために、x, y, z で記述した面の方程式を利用します。具体的には、方程式 $f(x, y, z) = 0$ で定義された面があった場合に、ある領域の中の座標 (x_0, y_0, z_0) を代入した $f(x_0, y_0, z_0)$ が正であればその領域はプラス側、負であればマイナス側と表現します。例として、

List 4.10 ● [cell] セクションの例題 (1)

```
1:  [ C e l l ]
2:    1   0  -10
3:    2  -1   10
4:  [ S u r f a c e ]
5:    10  SZ  3  5
```

を考えてみます。5 行目の SZ を用いて定義される面番号 10 の面は中心を (0,0,3)、半径を 5cm とする球面であるため、方程式は $f(x, y, z) = x^2 + y^2 + (z - 3)^2 - 5^2 = 0$ となります。球の内はマイナス側、外はプラス側と定義されますが、これらは球の中心の座標 (0,0,3) や球の外の適当な座標 (0,0,10) を代入して、$f(0, 0, 3) = -25 < 0,\ f(0, 0, 10) = 24 > 0$ となることから確認できます。[cell] セクションではこの様な領域の区切り方を行った上で、必要に応じて各領域にセル番号を付与します。例題 (1) の 2 行目では、セル番号 1 のセルがボイド (真空) であることと、その領域が球面のマイナス側であることが定義されており、図 16 で示す仮想空間を構築します。図 16 は、例題 (1) でつくられる領域を xz 平面で切った断面図として見た

ものです。本来は球ですが、2 次元平面で見た結果円となっています。例題 (1) の 3 行目では外部ボイドが明示的に定義されており、そのセル番号を 2 と指定しています。

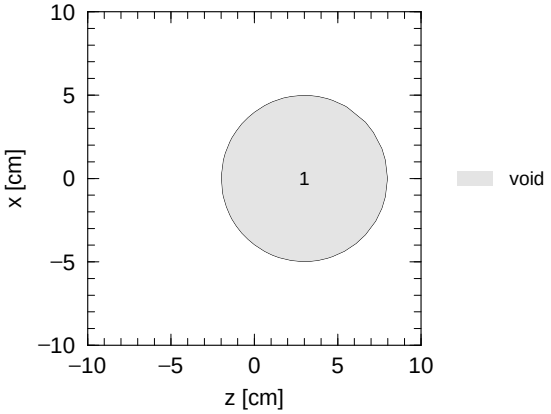


図 16: 例題 (1) の空間を *xz* 平面で切り取った断面図。セル番号 1 の領域はボイド (真空)。

例題 (1) の球面の様に 1 つの面だけで閉じた空間が定義できる場合は簡単ですが、[surface] セクションで定義できる面の多くは半無限領域をつくるため、幾つかの領域を組み合わせる必要があります。これを行うにあたって、*PHITS* では集合代数の考え方を導入しており、次の演算子を用いて閉じた領域を構成します。演算子は ⊔ (空白), :, #の 3 つで、それぞれ積 (and), 和 (or), 否定 (not) を表します。また、( ) で囲んだ範囲は 1 つの領域とみなします。ただし、⊔ (空白) と:は面番号同士を演算するのに対して、#はセル番号に掛かります。また、( ) を用いてまとめた領域に対しては、どの演算子でも掛けることができます。それでは、演算子を用いた次の例を考えてみましょう。

List 4.11 ● [cell] セクションの例題 (2)

```
 1:  [ C e l l ]
 2:    1   0  11  -12  13  -14  15  -16
 3:    2  -1  #1
 4:  [ S u r f a c e ]
 5:    11  PX  -6
 6:    12  PX   6
 7:    13  PY  -6
 8:    14  PY   6
 9:    15  PZ  -6
10:    16  PZ   6
```

この例題では半無限領域をつくる PX,PY,PZ の面を使用して、一辺が 12cm の立方体をつくっています。PX,PY,PZ はそれぞれ *yz*, *xz*, *xy* 平面に平行な面を定義する面記号であるため、例えば 5,6 行目で定義している面番号 11,12 の面は、それぞれ座標 (−6,0,0), (6,0,0) を通り *yz* 平面に平行な面となります。これらの方程式はそれぞれ $x + 6 = 0$, $x - 6 = 0$ ですから、原点 (0,0,0) を含む領域は、面番号 11 に対しては $0 + 6 = 6 > 0$ によりプラス側、面番号 12 に対しては $0 - 6 = -6 < 0$ よりマイナス側となります。よって、面番号 11 と 12 で囲まれた領域は “11 のプラス側かつ 12 のマイナス側” となり、演算子 ⊔ (空白) を用いて 11⊔-12 と記述します。面番号が 13~16 の面も同様ですから、x, y, z の 3 方向に関して閉じた空間を表現するには、2 行目の様に 6 つの面を ⊔ でつなげた書き方をします。例題 (2) ではこの閉じた領域をセル番号 1 でかつボイドであるとして定義しています。図 17 に、この例題でつくられる領域を *xz* 平面で切った断面図として示しました。本来は立方体ですが、2 次元平面で見た結果正方形となっています。また、2 行目ではセルに掛かる演算子#を用いて外部ボイドを定義しています。これはセル番号 1 の領域以外の範囲を対象にするという意味になります。

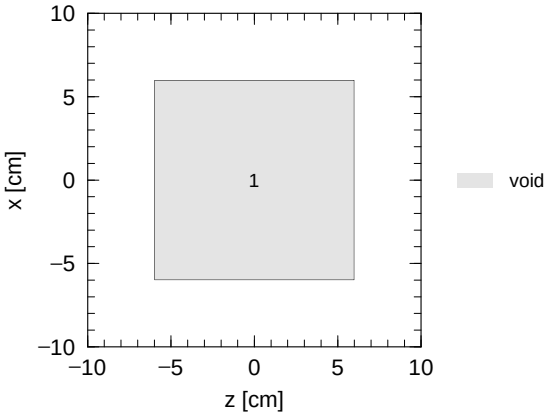


図 17: 例題 (2) の空間を *xz* 平面で切り取った断面図。セル番号 1 の領域はボイド (真空)。

次は ( ) と : の記号を使用した例です。例題 (1) で示した球と (2) の立方体を合わせた形状を考えています。

List 4.12 ● `[cell]` セクションの例題 (3)

```
 1:  [ C e l l ]
 2:    1   0  -10 : (11  -12  13  -14  15  -16)
 3:    2  -1  #1
 4:  [ S u r f a c e ]
 5:    10  SZ   3  5
 6:    11  PX  -6
 7:    12  PX   6
 8:    13  PY  -6
 9:    14  PY   6
10:    15  PZ  -6
11:    16  PZ   6
```

2 行目の ( ) で囲んだ部分が例題 (2) のセル番号 1 の領域に相当しており、これと例題 (1) の球面の内側を合わせた領域がこの例題ではセル番号 1 として定義されています。図 18 に示したのが例題 (3) の領域を *xz* 平面で切った断面図です。本来は立方体から球の一部が飛び出た形状をもつ立体ですが、2 次元平面では図の様になっています。本例題の様に、領域を足し合わせる場合に使用する演算子が : です。

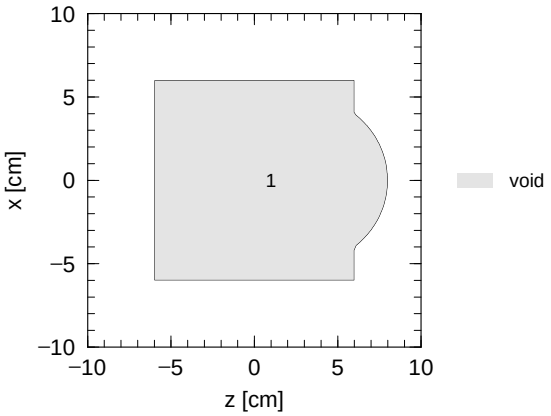


図 18: 例題 (3) の空間を *xz* 平面で切り取った断面図。セル番号 1 の領域はボイド (真空)。

例題 (4) では更に、立方体を球の内側と外側に分けた場合を考えます。

List 4.13 ● [cell] セクションの例題 (4)

```
 1:  [ M a t e r i a l ]
 2:    mat[1]  1H 2  16O 1
 3:  [ C e l l ]
 4:    1   0       -10
 5:    2   1  1.0   10 (11  -12  13  -14  15  -16)
 6:    3  -1        #1 #2
 7:  [ S u r f a c e ]
 8:    10  SZ   3  5
 9:    11  PX  -6
10:    12  PX   6
11:    13  PY  -6
12:    14  PY   6
13:    15  PZ  -6
14:    16  PZ   6
```

[surface] セクションは例題 (3) と同じです。5 行目で “立方体の内側でかつ球の外側” の領域を記号 ⊔ (空白) を用いて記述しており、これをセル番号 2 と定義しています。また、本例題では [material] セクションで “water” を物質番号 1 として定義しており、セル番号 2 の領域には粒子密度が 1.0×10^{24} atoms/cm^3 の “water” が満たされています。図 19 に結果を示します。ただし、*xz* 平面で切った断面図です。セル番号 1 は 4 行目で定義しており、この球の中はボイドとしています。外部ボイドの定義は 6 行目で行っており、セル番号 1 と 2 の領域を除いた他の全てが外部となっています。

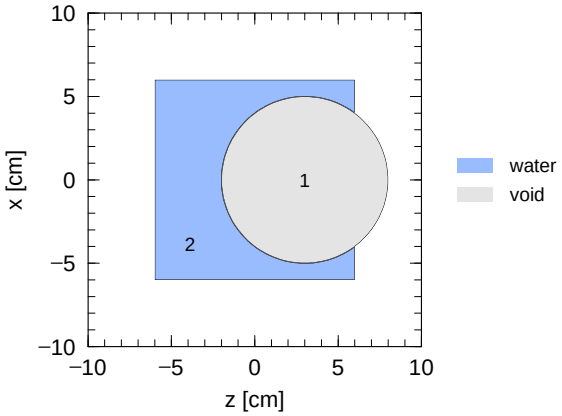


図 19: 例題 (4) の空間を *xz* 平面で切り取った断面図。セル番号 1 の領域はボイド (真空) とし、セル番号 2 の領域には水を満たした。

4.6.3 Universe 構造

PHITS では、セルパラメータ U を用いて、複数の宇宙 (universe) を定義できます。各宇宙に個別の仮想空間を構築することで、ある宇宙の一部として別の宇宙の内部構造を引用することが可能となります。この様に、多数の宇宙をそれぞれ定義し、相互に参照するための機能をここでは universe 構造と呼びます。ただし、実際に粒子輸送の舞台となる空間そのものは 1 つだけであり、この空間の一部を別の宇宙の内部構造で満たす (置き換える) 場合に universe 構造を利用します。単純に一部分だけを置き換える場合はあまり意味がありませんが、特に 4.6.5 節の繰り返し幾何形状を利用するときに有効となる機能です。

例題として、図 20 に示した 3 つの宇宙を組み合わせて仮想空間を構築する方法について考えます。図の (a) が実際に粒子輸送を行う空間で、半径 12cm の立方体を *yz* 平面で分割した 2 つの直方体で構成されて

います。ただし、示した図は *xz* 平面で切った断面図です。図の (b) と (c) が、(a) とは違う内部構造をもつ universe の 1 と 2 で、座標の中心に半径 5cm、高さ 12cm の円柱を配置しています。これらは、内部構造は同じですが、各領域を構成する物質が違っています。Universe 1 では円柱内部に水が満たされ、その外はボイド (真空) となっているのに対し、universe 2 では円柱内部は鉄で、その外部は水で満たされています。

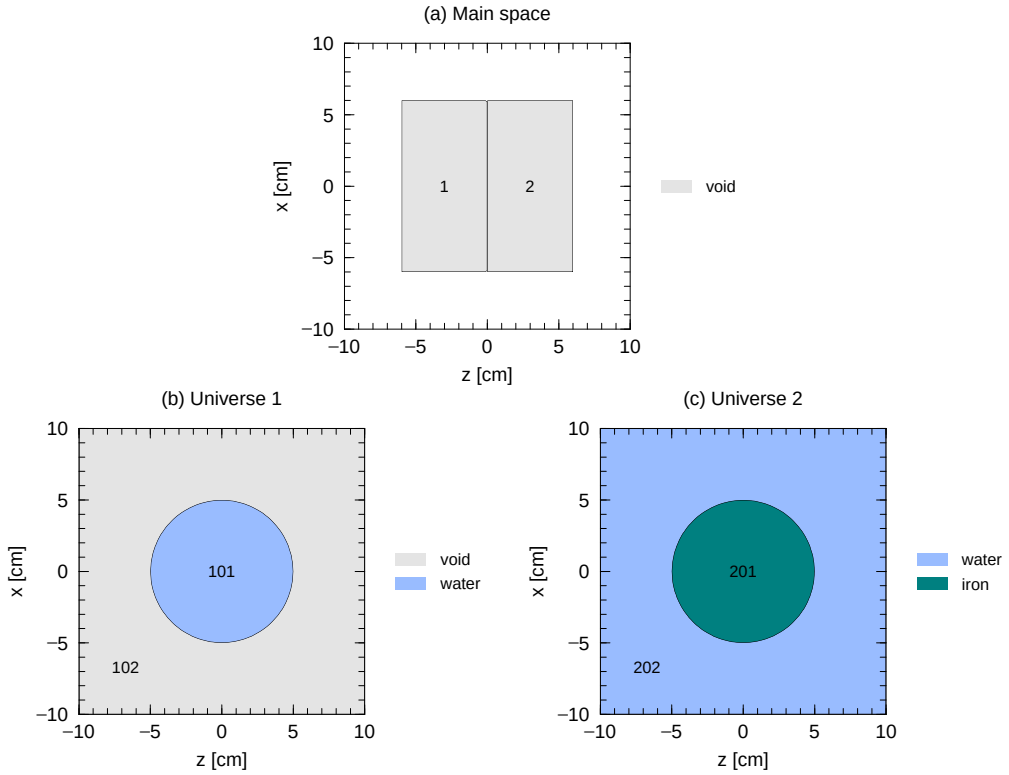


図 20: (a) 立方体を *yz* 平面で分割した 2 つの直方体。(b) 円柱内部に水を満たし、それ以外はボイドとした universe 1。(c) 鉄の円柱の周りを水で満たした universe 2。

(b) の universe 1 を (a) のセル番号 1 の領域に入れ、(c) の universe 2 をセル番号 2 の領域に入れた仮想空間を構築する例が以下の例題 (5) です。

List 4.14 ● `[cell]` セクションの例題 (5)

```
[ M a t e r i a l ]
  mat[1]  1H 2  16O 1
  mat[2]  Fe 1
[ C e l l ]
    1  0        11  -12  13  -14  15  -17  FILL=1
    2  0        11  -12  13  -14  17  -16  FILL=2
  101  1  1.0  -10  13  -14  U=1
  102  0       #101  U=1
  201  2 10.0  -10  13  -14  U=2
  202  1  1.0  #201  U=2
    9 -1       #1 #2
[ S u r f a c e ]
  10  CY   5
  11  PX  -6
  12  PX   6
  13  PY  -6
  14  PY   6
  15  PZ  -6
  16  PZ   6
  17  PZ   0
```

7,8 行目で universe 1 を、9,10 行目で universe 2 を定義しています。これらは共に面番号 10,13,14 を使って定義した円柱を中心に置いた宇宙となっています。Universe 1 では、セル番号 101 の円柱内部に物質番号 1 の水を入れ、その外部をボイドにしてセル番号 102 の領域と定義しています。Universe 2 では、円柱内部をセル番号 201 として鉄柱にし、セル番号 202 の外部には水を満たしています。7~10 行目の最後にある `U=1, 2` がそのセルがどの宇宙に属するかを指定するパラメータです。5, 6 行目で、それぞれ図 20(a) のセル番号 1,2 を定義しており、最後の `FILL=1, 2` によってそのセルをどの universe によって満たすか (置き換えるか) を決定します。この例題の結果を示したのが図 21 です。*xz* 平面の断面図ですが、セル番号 1 の領域が universe 1 に、セル番号 2 が universe 2 に置き換えられているのがわかります。

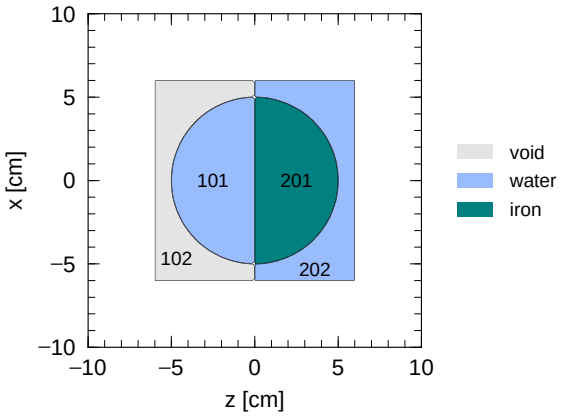


図 21: 例題 (5) の空間を *xz* 平面で切り取った断面図。セル番号 1 の領域に universe 1 のセル番号 101 と 102 が、セル番号 2 の領域に universe 2 のセル番号 201 と 202 が入っている。

Universe 構造を利用する際の注意点として、未定義の領域を引用しないことと、座標系が全ての宇宙で一致していることが挙げられます。例えば前者については、例題 (5) の 8 行目でセル番号 102 の領域をボイドとして定義する必要があり、これがないと 5 行目で引用した際にセル番号 102 の部分を適切に設定できません。各 universe の全ての領域を定義する必要はありませんが、引用する領域については何らかの物質 (かボイド) を指定しておかなければなりません。また後者については、座標原点の位置、x, y, z 軸の方向、空間のスケールがどの universe でも同じです。これは同じ `[surface]` セクションで定義した面を利用しているためです。別の宇宙を引用する場合は、対応する座標を確認してください。例えば、例題 (5) の 14,15 行目で `PX` の値を変えると、図 22 の様に円柱の一部が立方体の中に入らなくなります。

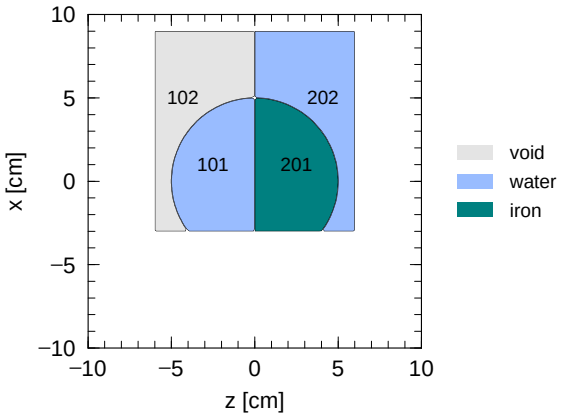


図 22: 例題 (5) において、セル番号 1 と 2 の範囲を *x* 方向に関してずらした場合。

4.6.4 Lattice 構造

本節では、lattice 構造の仕組みと定義の仕方、および基本的な利用方法について説明します。この機能は似た形状を多数繰り返す場合に非常に有効です。より実際的な設定方法については 4.6.5 節をご覧ください。

Lattice 構造は、図 23 に示す四角形と六角形の 2 種類の形状を基本単位として、格子状に繰り返し並べる機能です。U パラメータで指定して、基本格子を無限に並べた宇宙をつくります。ただし、その内部は別の U パラメータを用いて定義した宇宙で満たす必要があります。パラメータ LAT を用いて、四角形は LAT=1、六角形は LAT=2 として定義できます。図 23 に示した図形の各辺に付いた数字は各々の図形を定義する際の面の順番を意味しており、[surface] セクションで定義した面番号を図の数字の順に並べます。1 と 2、3 と 4、5 と 6 の面はそれぞれが平行な面として定義されている必要があります。もちろん、プラス側・マイナス側のどちらであるかも指定します。ただし、1 と 2 の長さと 3 と 4 の長さを変えることは可能で、正方形や正六角形でなくても構いません。

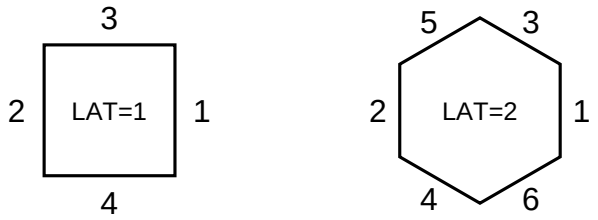


図 23: Lattice 構造の基本単位として使用できる四角形 (左) と六角形 (右)。各辺の番号は、LAT パラメータを定義する際の面の順番。

はじめに、四角形 (LAT=1) の lattice 構造を利用した例を考えます。

List 4.15 ● [cell] セクションの例題 (6)

```
 1:  [ M a t e r i a l ]
 2:    mat[1]  1H 2  16O 1
 3:  [ C e l l ]
 4:      1   0        11  -12   13  -14  15  -16  FILL=1
 5:    101   0       -26   25  -22   21  LAT=1  U=1  FILL=2
 6:    201   1  1.0  -90  U=2
 7:      2  -1       #1
 8:  [ S u r f a c e ]
 9:    11  PX  -6
10:    12  PX   6
11:    13  PY  -6
12:    14  PY   6
13:    15  PZ  -6
14:    16  PZ   6
15:    21  PX  -2
16:    22  PX   2
17:    23  PY  -2
18:    24  PY   2
19:    25  PZ  -2
20:    26  PZ   2
21:    90  BOX  -10 -10 -10  20 0 0  0 20 0  0 0 20
```

5 行目で lattice の基本格子を定義しています。LAT=1 で扱うことができる形状は四角形ですが、3 次元の形状を考えると実際には四角柱となります。図 23 で示した様に、定義には最低 4 面必要で、4 面の場合は高さが無限大の四角柱となります。したがって、本例題で定義しているのは断面が一辺 4cm の正方形となる無限の高さをもつ四角柱となり、これが y 軸に関して平行かつ隙間無く並んでいる宇宙を universe 1 としています。この格子状の集まり全てがセル番号 101 の領域です。これとは別に 6 行目において、面記号 BOX を用いて一辺 20cm の立方体が原点にある宇宙を定義しており、これは universe 2 としています。このセル

番号 201 の立方体には水が入っており、FILL=2 を用いることにより、セル番号 101 中の格子一つ一つが水で満たされることになります。4 行目でセル番号 1 を universe 1 で満たした領域として定義しており、これが一辺 12cm の立方体であることから、本例題の仮想空間は図 24 の様になります。左図が 3 次元的に見た立体図で、右図がこれを *xz* 平面で切り取った断面図です。図を見てわかる様に、同じ四角柱が全部で 9 つ並んでいます。これらは、universe 1 においては高さが無限大ですが、セル番号 1 の領域が *y* 軸方向にも制限されているため、結果的に有限の高さをもつ四角柱となっています。もし、有限の高さをもった四角柱を lattice の基本単位としたい場合は、例えば面番号 23, 24 による領域の制限を 5 行目で加える必要があります。

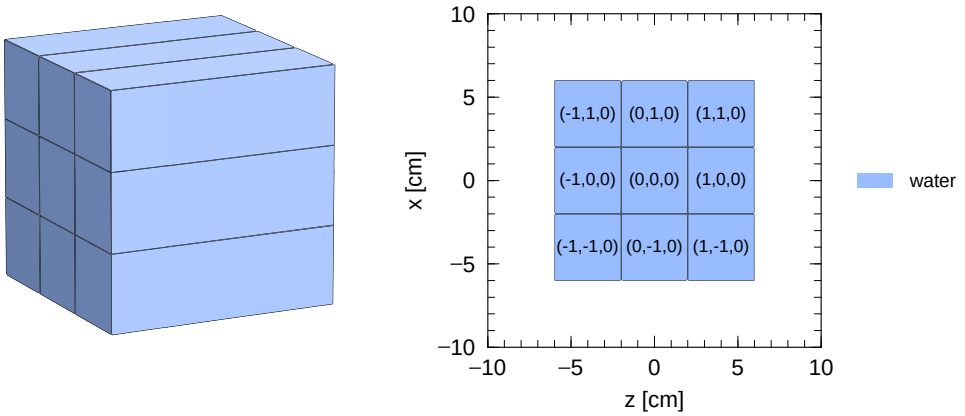


図 24: 例題 (6) の空間を立体的に見た図 (左) とこれを *xz* 平面で切り取った断面図 (右) 。各格子は、中に示された座標によって区別される。

図 24 の右図で示した様に、各円柱には lattice 座標 (s, t, u) がふられており、各々の領域がセル番号 101 の中のどれであるかを区別できるようになっています。ただし、この座標は必ずしも通常の座標 (x, y, z) とは対応しておらず、単位格子を定義する際の面の順番で決まります。すなわち、図 23 の左図において、辺 2→1 の方向が s、4→3 が t の方向となります。残りの u の向きは、s, t と同様にセル定義文の順番で決定されます。例題 (6) の 5 行目では、z 軸方向、x 軸方向の順番で定義しているため、s は z 軸、t は x 軸に対応付けがなされています。実際に図 24 の右図でも、z が増える方向に s が増加し、x が増える方向に t の値が増えていることがわかります。各タリーにおいて mesh=reg により領域指定する場合は、(201 < 101[-1 0 0] < 1) の書式を用います。これは “セル番号 1 を満たすセル番号 101 の lattice 座標 $(-1, 0, 0)$ に含まれているセル番号 201 の領域” という意味です。この書式に関しては 5.1.2 節をご参照ください。なお lattice 座標 (s, t, u) は、[t-gshow] タリーにおいて output=7 or 8 を指定することで表示させることができます。

次の例題は六角形 (LAT=2) の lattice 構造を利用した場合です。

List 4.16 ● [cell] セクションの例題 (7)

```
 1:  [ M a t e r i a l ]
 2:    mat[1]  1H 2  16O 1
 3:  [ C e l l ]
 4:      1   0        11 -12  13 -14  15 -16  FILL=1
 5:    101   0       -31  32 -33  34 -35  36 -24  23  LAT=2  U=1  FILL=2
 6:    201   1  1.0  -90  U=2
 7:      2  -1       #1
 8:  [ S u r f a c e ]
 9:    11  PX  -6
10:    12  PX   6
11:    13  PY  -6
12:    14  PY   6
13:    15  PZ  -6
14:    16  PZ   6
```

```
15:     23  PY  -2
16:     24  PY   2
17:     set: c1[2]
18:     31  PZ  [ c1*cos(pi/6)]
19:     32  PZ  [-c1*cos(pi/6)]
20:     33  P   1  0  [ 1/tan(pi/3)]  [ c1]
21:     34  P   1  0  [ 1/tan(pi/3)]  [-c1]
22:     35  P   1  0  [-1/tan(pi/3)]  [ c1]
23:     36  P   1  0  [-1/tan(pi/3)]  [-c1]
24:     90  BOX  -10 -10 -10  20 0 0  0 20 0  0 0 20
```

17~23 行目で正六角形をつくるための面の定義を行っています。これらを組み合わせると、*xz* 平面で見た場合に正六角形となり、*y* 軸に垂直な六角柱となります。ここで、正六角形の一辺の長さを 2cm とし、これを `c1` として定数定義しています。また、組み込み定数である `pi`(円周率 π) および面記号 `PZ, P` を用いて、60 度ずつ回転した平面を定義しています。Lattice 構造の基本格子は 5 行目で定義しており、セル定義文中の面番号の順番は図 23 の右図にある数字の通りとなっています。ただし、最後に面番号 23, 24 による *y* 軸方向の制限を加えています。したがって本例題では、高さが 4cm、一辺が 2cm の六角柱を無限に並べた宇宙が universe 1 となります。各六角柱には 6 行目で定義した universe 2 が入っており、実質的には水で満たされたセル番号 201 が入っていることになります。4 行目で、先の例題と同様に、セル番号 1 の領域を universe 1 で満たしており、本例題の結果は図 25 となります。左図が立体図、右図がこれを *xz* 平面で切った断面図です。左図からわかる様に、六角柱が詰められた universe 1 から立方体の領域だけを切り取った形状になっています。また、*y* 軸方向にも 3 列が並んでおりますが、その高さは揃えられており、いわゆる 3 次元のハニカム構造はつくりません。右図では各領域に lattice 座標 (s, t, u) が付けられています。図を見てわかる様に、*s* は左から右、*t* は左下から右上に向かって増加しており、これらはやはり 5 行目のセル定義文のところに並べる順番に依存します。図 23 の右図において、辺 2→1 の方向が *s*、4→3 が *t* の方向となります。*u* についても同様で、面番号 24→23 の順番であれば *y* 軸方向に、23→24 であれば $-y$ 軸方向に *u* の値は増加します。各タリーにおいて `mesh=reg` により領域指定する場合は、`(201 < 101[-2 0 0] < 1)` の書式を用います。これは “セル番号 1 を満たしたセル番号 101 の lattice 座標 $(-2, 0, 0)$ に含まれているセル番号 201 の領域” という意味です。この書式の詳細は 5.1.2 節をご覧ください。なお lattice 座標 (s, t, u) は、`[t-gshow]` タリーにおいて `output=7 or 8` を指定することで表示させることができます。

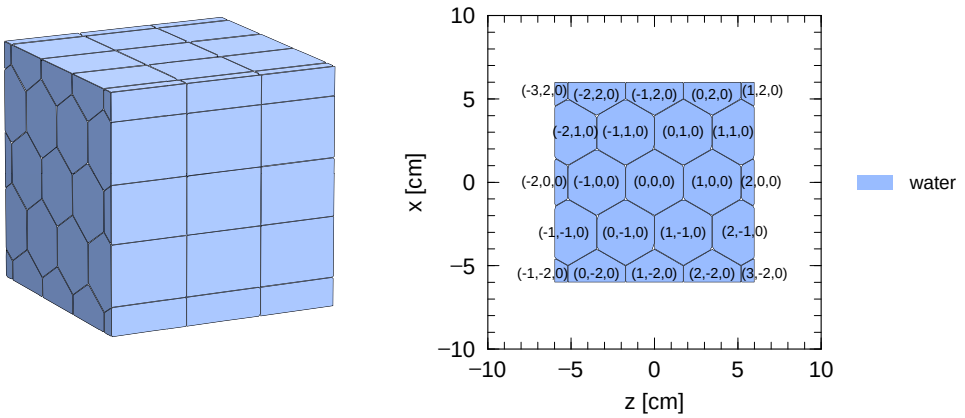


図 25: 例題 (7) の空間を立体的に見た図 (左) とこれを *xz* 平面で切り取った断面図 (右)。各格子は、中に示された座標によって区別される。

4.6.5 繰り返し幾何形状

多数の同じ、あるいは少し違うだけの構造のセルを並べる場合に、当然 1 つ 1 つのセルを定義することは可能ですが、基本となるセルを 1 つ用意しこれを繰り返すことで効率良く設定できます。4.6.4 節の lattice 構造の利用もその 1 つであり、他には LIKE *n* BUT セルパラメータ の表式も利用できます。

4.6.5.1 LIKE *n* BUT セルパラメータ

セル番号 *n* のセルと同様であるが、BUT 以下の内容だけが違うセルを定義する方法です。BUT の後に続くセルパラメータには表 70 に挙げたものがあります。以下の例題では、TRCL と MAT を使用した場合を考えています。

List 4.17 ● [cell] セクションの例題 (8)

```
 1:  [ M a t e r i a l ]
 2:    mat[1]  1H 2  16O 1
 3:    mat[2]  Fe 1
 4:  [ C e l l ]
 5:      1   0       -10  13 -14  #2 #3 #4
 6:      2   1  1.0   11 -12  13 -14  15 -16
 7:      3   LIKE 2 BUT  TRCL=1
 8:      4   LIKE 2 BUT  TRCL=2  MAT=2
 9:      5  -1        #(-10  13  -14)
10:  [ S u r f a c e ]
11:    10  CY   10
12:    11  PX  -2
13:    12  PX   2
14:    13  PY  -2
15:    14  PY   2
16:    15  PZ  -2
17:    16  PZ   2
18:  [T r a n s f o r m ]
19:  *tr1   3 0 -5
20:  *tr2   0 0  6  30 90 120  90 0 90  60 90 30  1
```

7, 8 行目で LIKE *n* BUT セルパラメータの表式を用いてセルを定義しています。この元となるセル番号 2 は 6 行目で定義しており、中に水が入った一辺 4cm の立方体です。セル番号 2 は原点を中心として配置されていますが、これを 19 行目で定義した座標変換番号 1 で座標変換したのがセル番号 3 です。座標変換番号 1 は、*x* 軸方向に 3cm、*z* 軸方向に −5cm 平行移動する変換ですから、*xz* 平面上で見ると図 26 の様になります。この図は本例題の結果を *xz* 平面で切り取った断面を示しています。次のセル番号 4 は TRCL と MAT のセルパラメータを使用しており、座標変換番号 2 による平行移動 (および回転) を行い、その上で物質番号 2 に中の物質を変更しています。回転は *xz* 平面上で 30 度行っており、図 26 でも確認できます。5 行目では、セル番号 1 を半径 10cm、高さ 4cm の円柱と定義しています。ただし、セル番号 2, 3, 4 の領域が含まれない様に、#を用いてこれらを除いています。また、これに伴い外部ボイドの定義も 9 行目に示す様な書き方が必要です。単純に #1 のみではセル番号 2, 3, 4 の領域も外部ボイドとなってしまいます。

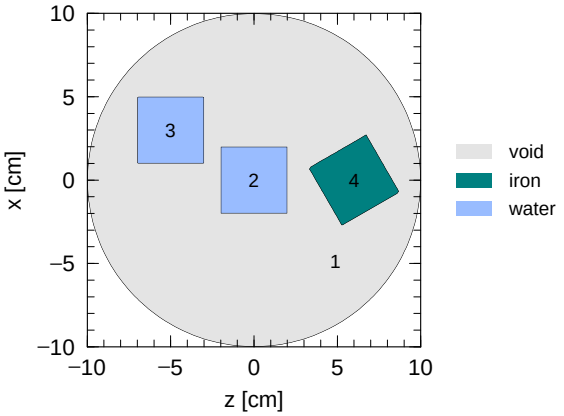


図 26: 例題 (8) の空間を *xz* 平面で切り取った断面図。セル番号 1 の円柱の中にセル番号 2, 3, 4 が入っている。

4.6.5.2 Lattice を用いた階層構造

4.6.4 節で見た四角柱や六角柱の lattice をより複雑なものにするために、universe を用いて階層構造を構築することができます。考え方は universe 構造に基づいており、universe 1 の一部を universe 2 で満たし、更に universe 2 の一部を universe 3 で満たし、といった手続きで各階層を定義していくことが可能です。*PHITS* では基本的に最大 10 (`param.inc` 中の `mxlv` で変更できます) の階層構造をつくることができます。

List 4.18 ● `[cell]` セクションの例題 (9)

```
 1:  [ M a t e r i a l ]
 2:    mat[1]  1H 2  16O 1
 3:    mat[2]  Fe 1
 4:  [ C e l l ]
 5:      1   0        11  -12   13  -14  15  -16  FILL=1
 6:    101   0       -26   25  -22   21  LAT=1  U=1
 7:                  FILL=-1:1  -1:1  0:0
 8:                  2 2 3  2 3 2  3 2 2
 9:    201   1  1.0  -90  U=2
10:    301   2 10.0  -10  U=3
11:    302   0        10  U=3
12:      2  -1       #1
13:  [ S u r f a c e ]
14:    10  CY   1.5
15:    11  PX  -6
16:    12  PX   6
17:    13  PY  -6
18:    14  PY   6
19:    15  PZ  -6
20:    16  PZ   6
21:    21  PX  -2
22:    22  PX   2
23:    25  PZ  -2
24:    26  PZ   2
25:    90  BOX  -10 -10 -10  20 0 0  0 20 0  0 0 20
```

まず 5, 6 行目にある、四角柱の lattice 構造とそれを引用するセル番号 1 の定義は例題 (6) と同じです。ただし、7, 8 行目におけるセルパラメータ `FILL` の使い方が違っており、ここでは各格子の状態を個別に指定しています。7 行目では取り扱う範囲を決めており、各格子に割り振られた lattice 座標 (s,t,u) を参照して `FILL=`の後ろで決定しています。本例題では、s に関して -1 から 1 まで (3 つ)、t に関して -1 から 1 まで (3 つ)、u に関して 0 から 0 まで (実質は 1 つ) をセル番号 101 の lattice 構造と定義しています。合計 $3\times3\times1 = 9$ の格子が対象となり、それぞれを満たす宇宙は 8 行目で指定しています。この順番は lattice 座標 (s,t,u) で表現して、$(-1,-1,0),(0,-1,0),(1,-1,0),(-1,0,0),\cdots$ の様になっており、2 であれば universe 2 で、3 であれば universe 3 でその格子を満たすことを意味します。

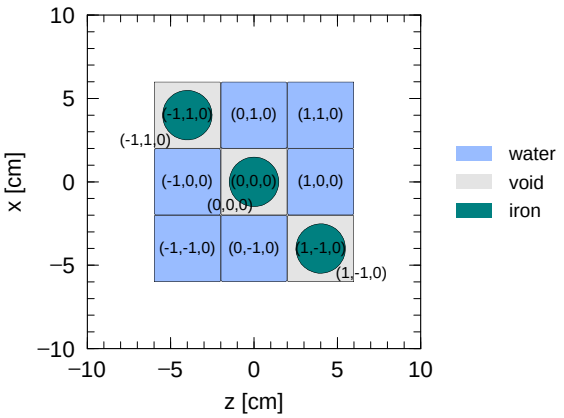


図 27: 例題 (9) の空間を xz 平面で切り取った断面図。9 つの四角柱の内、3 つが鉄柱を中心に配した内部構造をもっている。

Universe 2 を水が入った一辺 20cm の立方体を原点付近に配置した宇宙として、universe 3 を原点に半径 1.5cm の鉄柱を置いた宇宙として定義しており、9 つの四角柱の内 3 つが特別な構造をもつことになります。結果を示したのが図 27 で、xz 平面に関する断面図です。この図から $(s,t,u) = (1,-1,0),(0,0,0),(-1,1,0)$ の格子が内部構造をもっていることがわかります。各タリーにおいて `mesh=reg` により領域指定する場合は、`(302 < 101[0 0 0] < 1)` の書式を用います。これは “セル番号 1 を満たしたセル番号 101 の lattice 座標 (0,0,0)

に含まれているセル番号 302 の領域” という意味で、図 27 では中心の四角柱にある鉄柱周辺のボイドの領域を指します。この様な書式については 5.1.2 節を参照してください。

次は、階層構造をより複雑にした例です。

List 4.19 ● [cell] セクションの例題 (10)

```
 1:  [ M a t e r i a l ]
 2:    mat[1]  1H 2  16O 1
 3:    mat[2]  Fe 1
 4:  [ C e l l ]
 5:      1   0        11  -12   13  -14  15  -16  FILL=1
 6:    101   0       -26   25  -22   21  LAT=1  U=1
 7:                  FILL=-1:1  -1:1  0:0
 8:                  2 2 3(1 0 1)  2 3(1 0 1) 2  3(1 0 1) 2 2
 9:    201   1  1.0  -90  U=2
10:    301   0       -36   35  -32   31  LAT=1  U=3
11:                  FILL=-1:0  -1:0  0:0
12:                  4 2  2 4
13:    401   2 10.0  -10  U=4
14:    402   0        10  U=4
15:      2  -1       #1
16:  [ S u r f a c e ]
17:    10  CY   0.5
18:    11  PX  -6
19:    12  PX   6
20:    13  PY  -6
21:    14  PY   6
22:    15  PZ  -6
23:    16  PZ   6
24:    21  PX  -2
25:    22  PX   2
26:    25  PZ  -2
27:    26  PZ   2
28:    31  PX  -1
29:    32  PX   1
30:    35  PZ  -1
31:    36  PZ   1
32:    90  BOX  -10 -10 -10  20 0 0  0 20 0  0 0 20
```

この例題でつくられる仮想空間は、図 28 の様になります。ただし、*xz* 平面で切った断面を示しています。まず、一辺 12cm の立方体を 9 つの四角柱に分け、その内の 3 つに内部構造をもたせた点は例題 (9) と同じです。違うのは universe 3 として定義した宇宙を更に内部構造をもった格子で埋めたことで、9~11 行目でその状態を指定しています。ここでは $2 \times 2 \times 1 = 4$ の四角柱を定義しており、その内の 2 つが鉄柱をもつ格子となっています。また、8 行目の `(1 0 1)` は座標変換を表しており、引用する際に universe 3 を *x* 軸方向に 1cm、z 軸方向に 1cm 平行移動しなさい、という意味となります。セル番号 301 は、原点 (0,0,0) を中心にもつ四角柱を基本単位としているため、図の様な配置にする場合は少しずらす必要があります。また、lattice を用いて階層構造を作成した場合の `mesh=reg` による領域指定は、階層構造を増やした書式により行います。例えば図 28 において $x = -1, z = -1$ を中心とする微小四角柱のボイド部分は、`(402 < 301[-1 -1 0] < 101[0 0 0] < 1)` と記述します。もし対象とする階層が更に深くなった場合は、<と lattice 座標を用いて階層を増やすことで各領域を指定できます。この書式の詳細につ

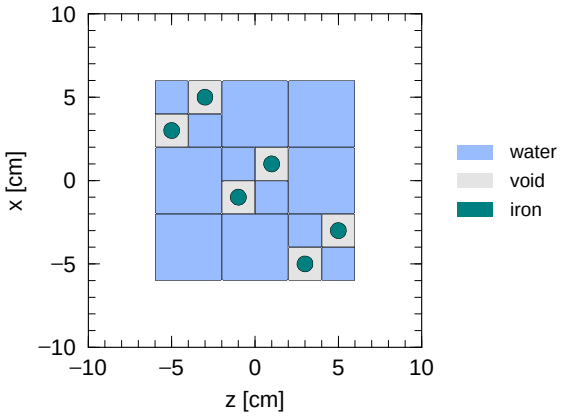


図 28: 例題 (10) の空間を *xz* 平面で切り取った断面図。9 つの四角柱の内 3 つが構造をもち、更に 4 つの四角柱で構成されている。

いては 5.1.2 節をご覧ください。

4.6.5.3 ボクセルファントム (voxel phantom) の利用

PHITS では lattice 構造を利用してボクセルファントム (voxel phantom) を仮想空間として設定できます。ボクセルファントムとは微小立方体を積み重ねて生物などの複雑な構造物を表現したものです。CT 等の画像データを元にして作成するボクセルデータを使用します。本節では、非常に簡単なボクセルデータを用意し、これを用いた設定方法について説明します。

基本的な考え方として、まず粒子輸送を行うためのある一定の大きさをもつ立方体 (直方体) を用意し、その中に微小立方体 (ボクセル) を並べます。その際、lattice 構造の `LAT=1` を利用し、各ボクセルの座標が指定できる様にセルパラメータ `FILL` を使います。`FILL` を使用した場合、各ボクセルがどの状態か、すなわちどの universe で満たされているか、が指定できるため、水や鉄 (あるいはボイド) で満たされた universe を引用することで、様々な構造物をつくることができます。したがって、ボクセルデータは各ボクセルの位置座標と構成物質の情報をもっている必要があり、ここでは $L_{000}\ L_{100}\ L_{200}\ \cdots\ L_{stu}\ \cdots$ の順番で並べたデータ群を想定しています。ここで、L_{stu} は lattice 座標 (s, t, u) にあるボクセルの構成物質 (universe 番号) です。

以下では、一辺が 10cm の立方体の中に一辺 2cm のボクセルを合計 $5 \times 5 \times 5 = 125$ 並べた場合を考えています。

List 4.20 ● `[cell]` セクションの例題 (11)

```
 1:  [ M a t e r i a l ]
 2:    mat[1]  1H 2  16O 1
 3:    mat[2]  Fe 1
 4:  [ C e l l ]
 5:      1   0   11  -12   13  -14  15  -16  FILL=1
 6:    101   0  -20  LAT=1  U=1
 7:             FILL=-2:2  -2:2 -2:2
 8:             2 2 2 2 2  2 2 2 2 2  2 2 3 2 2  2 2 2 2 2  2 2 2 2 2
 9:             2 2 2 2 2  2 3 3 2 2  2 3 4 3 2  2 3 3 2 2  2 2 2 2 2
10:             2 2 2 2 2  2 3 3 3 2  3 4 4 4 3  2 3 3 3 2  2 2 2 2 2
11:             2 2 2 2 2  2 2 3 3 2  2 3 4 3 2  2 2 3 3 2  2 2 2 2 2
12:             2 2 2 2 2  2 2 2 2 2  2 2 3 2 2  2 2 2 2 2  2 2 2 2 2
13:    201   0       -90  U=2
14:    301   2 10.0  -90  U=3
15:    401   1  1.0  -90  U=4
16:      2  -1       #1
17:  [ S u r f a c e ]
18:    11  PX  -5
19:    12  PX   5
20:    13  PY  -5
21:    14  PY   5
22:    15  PZ  -5
23:    16  PZ   5
24:    20  BOX  -1 -1 -1  2 0 0  0 2 0  0 0 2
25:    90  BOX  -10 -10 -10  20 0 0  0 20 0  0 0 20
```

5 行目でセル番号 1 の領域を一辺が 10cm の立方体として定義しています。これが次の行のセル番号 101 の lattice で満たされており、基本単位は面番号 20 で定義した一辺 2cm の立方体です。ここでは面記号 `BOX` を用いて微小立方体 (ボクセル) を定義しています。7 行目でボクセルを lattice 座標 (s, t, u) のそれぞれに 5 つずつ並べるとしており、8~12 行目で各ボクセルを満たす universe 番号を指定しています。ボクセルは $(-2, -2, -2), (-1, -2, -2), \cdots, (2, 2, 2)$ の順番で並んでおり、ボイドの場合は 2、鉄は 3、水は 4 の数字が対応しています。ボイドや鉄、水が入った空間の定義は 13~15 行目で行っており、それぞれが一辺 20cm の立方体に入った宇宙を universe 2, 3, 4 としています。本例題のボクセルで表現しているのは多少いびつな形状の鉄の箱に水が入っている構造物です。結果を図 29 に示しました。左図が立体的にそのまま見た結果、右図が表面の一部を透明にして見た結果です。少しずれていますが、図の左から右が s 軸、下から上が t 軸、奥

から手前が u 軸の方向に対応します。見てわかる様に、でこぼこした鉄の箱の中には水が入っており、この構造物を対象に粒子輸送計算を行うことができます。各タリーにおいて `mesh=reg` により領域指定する場合は、`(401 < 101[0 0 0] < 1)` の書式を用います。あるいは、セル番号 1 中の水の部分全てを指定したい場合は、`(401< 1)` とすることができます。これらの書式については 5.1.2 節をご覧ください。ただし、`(301 < 101[-2:2 -2:2 -2:2] < 1)` の様な指定方法はできません。セル番号 `101[-2:2 -2:2 -2:2]` の全てにセル番号 301 の領域が含まれているわけではないためです。

図 29: 例題 (11) の構造物を立体的に見た図 (左) と表面の一部を取り除き内部を見た図 (右)。

PHITS では、最初に読み取ったインプットデータを一時的にバイナリ形式で書き出し、それを再度読み込んで粒子輸送計算を行います。例題 (11) で扱ったのは簡単なボクセルですが、一般的にボクセルデータは大規模なものとなり、それを *PHITS* の実行の度に読み書きするのは非常に時間がかかります。これを緩和するために、`[parameters]` セクションの `ivoxel` パラメータを使う方法があります。`ivoxel` は、ボクセルデータ (実質的には例題 (11) の 8 から 12 行目) をバイナリとして出力させて保存し、2 度目からはそのバイナリを読んで計算するという機能をもちます。*PHITS* 実行時に `ivoxel=2` とすると `file(18)` にデータを書き出してそのまま終了します (ただし、バージョン 2.30 以前では、終了せずに引き続き輸送計算を行います) ので、以降 `ivoxel=1` と設定することで該当ファイルからバイナリデータを読み込んで計算できます。また、大量のボクセルデータを *PHITS* のインプットファイルとは別のファイルに保存して計算するときは、`infl` を使用します。この場合の `ivoxel` の使い方も同じです。

4.6.5.4 連続四面体形状の利用

本節では、連続四面体形状の仕組みと定義の仕方、および基本的な利用方法について説明します。連続四面体形状はポリゴンの一種で、様々な大きさの四面体を組み合わせることで、複雑な形状をもつ物体を表現できるようになります。

連続四面体形状は、

- Node ファイル: 各四面体の各々の頂点の xyz 座標
- Element ファイル: 各四面体要素を構成する 4 つの頂点

の 2 つのファイルで表現されます。Node ファイルおよび Element ファイルは、共通のファイル名に.node および.ele という拡張子が付いた形を想定しています。PHITS のインプットファイルでは、この共通ファイル名を指定することになります。この機能の具体的な利用方法については、`\phits\utility\TetraGEOM` フォルダにある資料やサンプルインプットをご参照ください。

次に、Node ファイルおよび Element ファイルの例を示します。

List 4.21 ● Node ファイルの例

```
1:   # Simple two elements
2:   5   3  0  0
3:   #  pointID x y z
4:   1    0.0  -2.5  -4.0
5:   2   -4.0  -2.5   0.0
6:   3    4.0  -2.5   0.0
7:   4    0.0  -2.5   4.0
8:   5    0.0   2.5   0.0
```

Node ファイルでは、まず始めにこのファイルに含まれる全ての頂点の数と扱っている次元数を設定します (2 行目)。PHITS で扱える次元は 3 次元のみです。2 行目の最後の 2 つの 0 は PHITS では使われずに無視されます。その下に (4 行目以下)、

`[頂点番号] [x] [y] [z]`

の順で頂点番号とその頂点の xyz 座標を指定します。

List 4.22 ● Element ファイルの例

```
1:   # Simple two elements
2:   2  4  1
3:   # elementID point(1:4) universe
4:   1     1  2  3  5    5001
5:   2     5  2  3  4    5002
```

Element ファイルでは、まず始めにこのファイルに含まれる全要素数と 1 つの要素を構成する頂点の数（PHITS では、1 次要素の四面体しか扱えないので、4 となります）と要素に付加する情報の数（PHITS では 1）を設定します (2 行目)。その下に (4 行目以下)、

`[要素番号] [頂点番号 1] [頂点番号 2] [頂点番号 3] [頂点番号 4] [要素ユニバース番号]`

の順で各要素の情報を指定します。要素番号は各四面体の ID 番号、頂点番号 1~4 はその四面体を構成する頂点の番号で Node ファイル内で定義した番号です。最後の要素ユニバース番号で、各四面体をどの universe 番号の空間で満たすかを指定します。各 universe の定義は PHITS のインプットファイルで行います (List 4.23 の 17,18 行目参照)。なお、Node ファイルおよび Element ファイル内では、任意の行で#から始まるコメント行を挿入することができます。

Node ファイルおよび Element ファイルは四面体メッシュ分割ソフトウェア TetGen を使って作成することができます。多少技術が必要ですが、一般的なポリゴンデータもこのソフトウェアを使って連続四面体に変換することが可能です。TetGen は以下のウェブサイト

`http://wias-berlin.de/software/tetgen/`

から無料でダウンロード可能です。TetGen の使い方については、TetGen のマニュアルをご参照ください。

次に、PHITS のインプットファイルにおける指定の仕方について説明します。

List 4.23 ● `[cell]` セクションの例題 (12)

```
 1:   [Material]
 2:   mat[1] 14N 78.1  16O 20.9  40Ar 0.93
 3:   mat[2] 1H 2  16O 1
 4:   mat[3] 56Fe 1
 5:   [Surface]
 6:   10  rpp   -5.0  5.0  -3.0  3.0  -5.0  5.0
 7:   20  rpp   -7.0  7.0  -5.0  5.0  -7.0  7.0
 8:   90  so     500.0
 9:   [Cell]
10:   101   1 -0.001205 -20  U=1  LAT=3  tfile=Tetra  TSFAC=1.0
11:     1   0           -10  FILL=1
12:     2  -1            10
13:   201   2 -1.0       -90  U=5000
14:   202   3 -7.874     -90  U=5001
```

PHITS で連続四面体形状を取り扱うためには、直方体（面記号 RPP）で指定する領域を定義し、その中に四面体連続形状を展開します。全ての四面体の頂点はこの直方体の中に含まれる必要があります。ただし、直方体の大きさを連続四面体形状の大きさからかけ離れて大きく取り過ぎると計算の都合で問題が生じるので、連続四面体形状が全て収まる適当な大きさの直方体としてください。10 行目では、セル番号 101 の領域として、面番号 20 で定義される 14cm×10cm×14cm の直方体の中にファイル名 Tetra で指定される連続四面体形状を展開しています。`LAT=3` は連続四面体形状を使用するオプションの指定で、`TFILE` に与えられるファイル名に対応する Node ファイルおよび Element ファイルが読み込まれることになります。Windows 以外の OS では、ファイル名に大文字小文字の区別があるので、ここでのファイル名の指定も大文字小文字を正しく入力してください。`TSFAC` の指定により、Node ファイルで与えられる四面体頂点の座標に対する規格化定数を与えることができます。ここでは、`TSFAC=1.0` なので、Node ファイルで与えられる座標をそのまま使用します。セル番号 101 の領域に対するマテリアルの指定は領域 101 内で連続四面体を除く領域で使用されることになります。11 行目でセル番号 1 の領域を 10cm×6cm×10cm の直方体として定義しています。`Fill` の指定により、この領域が 10 行目のセル番号 101 で定義される連続四面体を含む領域によって満たされることになります。このように連続四面体形状 (`LAT=3`) を使用する場合も、Lattice 構造の指定と同じように、Universe と Fill を利用した入れ子構造を定義してください。Element ファイルで指定される各四面体の要素ユニバースに対応する領域の指定は Lattice 構造での指定と同様で 13,14 行目で行っています。この例題では、四面体要素 1 には物質番号 2 の水 (H_2O) が、四面体要素 2 には物質番号 3 の鉄 (Fe) が含まれることになります。

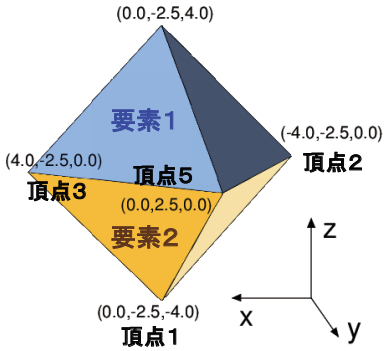


図 30: 例題 (12) で作られる連続四面体形状。

4.7 [Transform] セクション

4.7.1 書式

このセクションでは、座標変換カードの定義を行います。コメント文字として C も使えます。C 以外のコメント文字としては $ のみ用いることができます。ファイルのインクルード文、定数定義などは、セクションの途中でも用いることが出来ます。なお，[transform] を用いて可能な座標変換は回転と平行移動のみで，拡大・縮小はできない点にご注意ください。

このセクションで定義した座標変換は、ソース、[surface] セクション、[cell] セクション、タリーの r-z、xyz メッシュ、磁場の定義などで使うことができます。

書式は以下の通りです。

```
[ T r a n s f o r m ]
   TRn    O1   O2   O3   B1   B2   B3   B4   B5   B6   B7   B8   B9   M
```

表 71: 座標変換記述書式

項目	説明
n	座標変換番号 1 ~ 999999 ∗TRn は、B_i が角度の余弦ではなく、degree 単位の角度であることを意味する。
O_1 O_2 O_3	変換の転置ベクトル
B_1 ~ B_9	変換の回転行列 なお，定義できる行列は回転行列のみです。行列式が 1 でない場合は，1 となるように規格化されます。
M	= 1 は、転置ベクトルが主座標系で定義した補助座標系の位置であることを意味する。 = −1 は、転置ベクトルが補助座標系で定義した主座標系の位置であることを意味する。

デフォルト値は、

```
   TRn    0   0   0    1   0   0   0   1   0   0   0   1    1
```

4.7.2 座標変換の定義

回転行列、転置行列による座標変換の定義は以下の通りです。

$M = 1$ のとき

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} B_1 & B_2 & B_3 \\ B_4 & B_5 & B_6 \\ B_7 & B_8 & B_9 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix}$$

$M = -1$ のとき

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} B_1 & B_2 & B_3 \\ B_4 & B_5 & B_6 \\ B_7 & B_8 & B_9 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} B_1 & B_4 & B_7 \\ B_2 & B_5 & B_8 \\ B_3 & B_6 & B_9 \end{pmatrix} \begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix}$$

ここで、

$$\begin{aligned} B_1 &= \cos(x', x) \\ B_2 &= \cos(x', y) \\ B_3 &= \cos(x', z) \\ B_4 &= \cos(y', x) \\ B_5 &= \cos(y', y) \\ B_6 &= \cos(y', z) \\ B_7 &= \cos(z', x) \\ B_8 &= \cos(z', y) \\ B_9 &= \cos(z', z) \end{aligned}$$

です。直感的には、変換を行う対象を、$M = 1$ の場合は回転してから平行移動させ、$M = -1$ の場合は平行移動してから回転させます。回転は、どちらの場合も、xyz 座標系の原点に対して行われます。ただし、平行移動は $M = 1$ と -1 で逆向きとなるのでご注意ください。

List 4.24 ● `[transform]` セクションの例題 1

```
[ T r a n s f o r m ]
*tr1       0.0000000E+00  0.0000000E+00  1.4000000E+02
           1.3500000E+02  9.0000000E+01  4.5000000E+01
           9.0000000E+01  0.0000000E+00  9.0000000E+01
           2.2500000E+02  9.0000000E+01  1.3500000E+02    1

*tr2       0.0000000E+00  0.0000000E+00  2.5800000E+02
           3.0000000E+02  9.0000000E+01  2.1000000E+02
           9.0000000E+01  0.0000000E+00  9.0000000E+01
           3.0000000E+01  9.0000000E+01  3.0000000E+02    1
```

上の例題では、`tr1` は y 軸周りに 135 度回転し z 方向に 140cm 移動する座標変換、`tr2` は y 軸回りに 300 度回転し z 方向に 258cm 移動する座標変換となっています。ここでは $*$ が付いているため、$B_i, (i = 1, \ldots, 9)$ には余弦 (cos) の値ではなく回転角度 (単位: 度, degree) を入れることになります。

List 4.25 ● [transform] セクションの例題 2

```
[ T r a n s f o r m ]
set: c10[90]  $ angle of around Z (degree)
set: c20[30]  $ angle of around Y (degree)
set: c30[0]   $ angle of around X (degree)

tr1   0 0 0
      cos(c10/180*pi)*cos(c20/180*pi)
      sin(c10/180*pi)*cos(c30/180*pi)+cos(c10/180*pi)*sin(c20/180*pi)*sin(c30/180*pi)
      sin(c10/180*pi)*sin(c30/180*pi)-cos(c10/180*pi)*sin(c20/180*pi)*cos(c30/180*pi)
     -sin(c10/180*pi)*cos(c20/180*pi)
      cos(c10/180*pi)*cos(c30/180*pi)-sin(c10/180*pi)*sin(c20/180*pi)*sin(c30/180*pi)
      cos(c10/180*pi)*sin(c30/180*pi)+sin(c10/180*pi)*sin(c20/180*pi)*cos(c30/180*pi)
      sin(c20/180*pi)
     -cos(c20/180*pi)*sin(c30/180*pi)
      cos(c20/180*pi)*cos(c30/180*pi)
      1
```

上の例題では、z 軸周りに c10 度、次に y 軸周りに c20 度、最後に x 軸周りに c30 度回転させる回転行列を書き下したものです。c10, c20, c30 の定数に値を入れることにより、任意の回転を定義できるので便利です。

4.8 [Temperature] セクション

このセクションでは、GG の cell の Free-Gas Thermal Temperature を定義します。単位は MeV です。時間の定義はできません。この量は、GG の場合、[cell] セクションのセルパラメーターでも定義できますが、 領域が重なった場合は、この [temperature] で定義した値が優先します。両者で定義されない領域は、2.53×10^{-8}MeV にセットされます。書式は、

```
[ T e m p e r a t u r e ]
    reg                tmp
      1              1.0*1.e-8
     11              5.0*1.e-8
( { 2 - 5 } 8 9 )     2.0*1.e-8
  ( 11 12 15 )        3.0*1.e-8
     16               6.0*1.e-8
    ....              ........
    ....              ........
```

同じ値の領域をまとめて書く、({ 2 - 5 } 8 9) という書式も使えます。ただし、単一の数字で無い場合は必ず (　　) で括ってください。しかし、(6 < 10[1 0 0] < u=3) などの lattice, universe 構造は、 使えません。

領域番号 (reg) と温度 (tmp) の順番を変えたいときは、 tmp reg とします。読み飛ばしコラム用の non も使えます。GG の場合も、cell でなく reg を使ってください。

4.9 [Mat Time Change] セクション

このセクションでは、ある特定の物質をある時間で異なる物質に変化させる機能を定義します。時間の単位は nsec です。書式は、

```
[ Mat Time Change ]
            mat       time      change
              1       50.0       11
              2      100.0       12
              3     1000.0        0
            ....     ........
            ....     ........
```

物質番号 (mat) と (time change) の順番を変えたいときは、mat change time の様に定義行で変えます。読み飛ばしコラム用の non も使えます。この 3 つのコラムは全て必須です。mat で定義された物質が、time で定義された時間で change で定義された物質に変化します。時間によってシャッターの開閉がある場合などを想定しています。

4.10 [Magnetic Field] セクション

4.10.1 荷電粒子

PHITS では、指定した領域において磁場を設定することができます。ただし、電子・陽電子についてはこの機能が動作しないのでご注意ください。このセクションでは、磁場を及ぼす region もしくは cell の番号 (reg) と、磁場のタイプ (typ)、磁石の間隔の半分 (gap)、磁場の強さ (mgf)、座標変換 (trcl)、時間変化 (time) を定義します。このセクションで定義された磁場を有効にするには、パラメータセクションで、imagnf=1 を指定しなければなりません。書式は、

```
[ Magnetic  Field ]
  reg              typ    gap           mgf           trcl     time
    1               4     10.00000     -5.956540        3      non
    2               4     10.00000      6.416140        1      non
    3               2     10.00000     -7.611980        0      0.0
    4               2     10.00000      3.516000        0      pi/2
   ( 150 < 61 )     4     13.00000      7.880140        2      non
   ( 150 < 62 )     4     13.00000     -7.440800        2      non
   ( 150 < 63 )     4     13.00000      9.441010        2      non
   ( 150 < 64 )     4     13.00000     -8.295220        2      non
   ( 150 < 65 )     4     13.00000      3.694830        2      non
   ( 150 < 66 )     4     13.00000     -2.099350        2      non
   ...             ...    ........      ........       ...     ...
   ...             ...    ........      ........       ...     ...
```

ここで、time は、ユーザー定義の磁場の時間変化のパラメータです。non が、時間変化無しです。ユーザー定義サブルーティンは、usrmgt1.f と usrmgt2.f がサンプルルーティンとして含まれています。前者が Wobbler 磁石、後者が中性子用パルスマグネットです。これらを切り替えるには、パラメータセクションで、usrmgt=1,2 で指定します。Wobbler の時の time パラメータは、位相を指定し、パルスマグネット時は、立ち上がり時間を指定します。pi は、円周率の定数として使用できます。

reg は、region、typ は磁場のタイプ、二重極電磁石が 2、四重極電磁石が 4、gap は、磁石の間隔の半分 (cm)、mgf は磁場の強さ (kG)、trcl は、座標変換の定義番号です。座標変換は、[transform] セクションで定義します。

同じ値の領域をまとめて書く、({ 2 - 5 } 8 9) という書式も使えます。また、(6 < 10[1 0 0] < u=3) などの lattice, universe 構造も指定できます。 ただし、単一の数字で無い場合は必ず (　　) で括ってください。

この書式を用いれば lattice のひとつひとつに 異なる magnet field を定義することも可能です。上の定義で、同じセルが二重に定義された時は、初めに定義された値が採用されます。

これらの順番を変えたいときは、定義の文字列の順番を変えます。読み飛ばしコラム用の non も使えます。GG の場合も、cell でなく reg を使ってください。

二重極電磁石の場合、磁石の間隔 gap は意味がありませんが、何らかの数値を入力して下さい。磁場の働く領域は、真空の領域、即ち、mat が 0 のときに 限りません。磁場中でも減速、散乱が起こります。

trcl は、省略できます。trcl コラムがある場合で、座標変換のないときはゼロを書いてください。磁場中心の軸は z 軸上、磁場の方向は、二重極電磁石では、y 軸正方向、即ち正荷電粒子が z 軸正方向に進むときに、x 軸負方向に偏極します。四重極電磁石では、正荷電粒子が z 軸正方向に進むときに、x 軸方向に収

束、y 軸方向に分散の配置です。これらの位置関係を幾何形状に合わせるには、`trcl` による座標変換が必要です。座標変換で、z 軸を磁石の幾何形状の中心軸に変換します。

`[source]` セクションのパラメータ `izst=`を用いて入射粒子の電荷価数を指定した場合、磁場中の運動はその価数に応じたものになります。通常の *PHITs* では、荷電粒子、特に原子核は、磁場の中で電荷価数 Z の粒子として振る舞います。しかしながら、加速原子核では、電荷価数が Z、即ち全ての電子が剥ぎ取られた状態以外に、幾つかの電荷状態が存在するときがあります。`izst` を用いることで、その状態にある粒子の運動を模擬できるようになります。ここで指定された価数は、輸送の途中では変化しません。ただし、核反応を起こし、生成される原子核には影響を及ぼしません。即ち、原子核反応で生成された原子核の価数は、Z に戻ります。

4.10.2 中性子

中性子光学のための磁場の書式は、ほぼ荷電粒子の場合と同じですが、細かい点が異なりますので、ここにまとめて説明します。まず書式の例題を下に示します。

```
[ Magnetic  Field ]
  reg    typ    gap           mgf          trcl      polar     time
    1     60    0.00000      35000.0         3         non      non
    2     61    0.00000      35000.0         1          1       non
    3    106    5.00000       7130.0         0          0       non
    4    104    0.00000       3.5            0         non      5.0
    5    102    0.00000       0.20           0         non      non
    6    101    3.00000       7130.0         2          1       non
    7    103    0.00000      35000.0         0         -1       non
   ...   ...    ........      ........      ...       ...      ...
   ...   ...    ........      ........      ...       ...      ...
```

タイプは上の 7 種類あります。60、61 は、最もシンプルな 6 極磁場で、重力効果や追加 2 極磁場は入れられません。60 は、スピンとの相互作用なし、即ちスピンは磁場に平行か反平行かで記述されます。61 は、スピンとの相互作用が入ります。磁場の弱いところでのスピンの反転などが起こります。磁場の強さは、mgf のコラムに [T/m^2] の単位で指定します。

次に 100 番台は、全てスピンとの相互作用を考慮しています。また、重力効果、追加の 2 極磁場が入れられます。106 が 6 極、104 が 4 極、102 が 2 極磁場です。z 方向の追加 2 極磁場の強さは、gap のコラムに記述します。単位は T(テスラ) です。

101 は、磁場の値をユーザー定義ファイル usrmgf1.f で定義します。このユーザープログラムでは、原研の中性子光学グループで測定された 6 極磁場の値が格納された 4 つのファイルからデータを読み込み、それらを内挿して計算に用います。磁場の強さは、mgf で定義した値に規格化します。gap のコラムで与えた 2 極磁場は、定義した磁場領域全域に作用します。このサンプルを動かすインプットファイル sex03b.in を添付しますので、参考にして下さい。

103 も、ユーザー定義ファイルを参照するオプションです。103 では、usrmgf3.f を参照します。usrmgf3.f には、6 極磁場の式が書いてありますので、このまま動かせば、106 を指定した時と同じになります。ユーザー定義磁場を書く場合は、添付の usrmgf1.f と usrmgf3.f を参考にして下さい。

中性子のスピンは、ソースセクションで `sx, sy, sz` で、定義すれば、その方向で磁場に入射します。ソースセクションで定義しないか、`sx, sy, sz` 全てがゼロの場合は、最初に磁場に入った時の磁場の方向で初

期化されます。その時、このセクションの `polar` のコラムで定義される偏極率によりスピンの向きが磁場に平行か反平行かの比率が決定されます。non は、偏極率ゼロと同義です。偏極率 P の定義は、

$$P = \frac{\phi_+ - \phi_-}{\phi_+ + \phi_-}$$

ここで、ϕ_+ と ϕ_- は、スピンが磁場に平行な粒子数と反平行な粒子数です。

4.11 [Electro Magnetic Field] セクション

このセクションでは、指定した空間に一様な電場と磁場を設定するためのパラメータを定義します。[parameters] セクションで ielctf=1 とすることでこのセクションで定義したパラメータが有効となり、荷電粒子の電磁場中の運動を記述できるようになります。ただし、電子・陽電子についてはこの機能が動作しないのでご注意ください。

電場と磁場は任意に併存させることができます。ただし、設定できるのは共に一様な場のみです。[Magnetic Field] セクションで定義できる 4 極磁場は設定できません。

パラメータは、電磁場を指定する region もしくは cell の番号 (reg)、電場の強さ (elf)、磁場の強さ (mgf)、電場の方向 (trcle)、磁場の方向 (trclm) です。場の強さ elf と mgf の単位は、それぞれ kV/cm と kGauss です。電場と磁場の方向 trcle と trclm は、座標変換番号 trcl を用いて定義します。これらを 0 とした場合は、電場の向きが x 軸の +、磁場の向きが y 軸の+の方向となります。trcle と trclm は省略できません。場の強さを 0 とした場合や trcl が必要ない場合も 0 と入力してください。書式は以下のようになります。

```
[ Electro Magnetic Field ]
  reg    elf    mgf    trcle    trclm
    1    100      1        1        2
    2    100      1        1        2
```

[t-track] タリーを用いて電磁場中の粒子の軌跡をプロットする場合は、[parameters] セクションにおいて itstep=1 とすると、より正確なタリー結果を得られるようになります。

[source] セクションのパラメータ izst=を用いて入射粒子の電荷価数を指定した場合、電磁場中の運動はその価数に応じたものになります。通常の *PHITS* では、荷電粒子、特に原子核は、電場や磁場の中で電荷価数 Z の粒子として振る舞います。しかしながら、加速原子核では、電荷価数が Z、即ち全ての電子が剥ぎ取られた状態以外に、幾つかの電荷状態が存在するときがあります。izst を用いることで、その状態にある粒子の運動を模擬できるようになります。ここで指定された価数は、輸送の途中では変化しません。ただし、核反応を起こし、生成される原子核には影響を及ぼしません。即ち、原子核反応で生成された原子核の価数は、Z に戻ります。

4.12 [Delta Ray] セクション

このセクションでは、物質中を荷電粒子が通過した場合にその飛跡周辺に発生するノックアウト電子 (δ 線) を 2 次粒子として発生させる機能をコントロールします。荷電粒子が通過する物質に与えるエネルギーは、通常 LET(dE/dx) として評価され、その軌道上にのみ付与されます。しかし、高エネルギーの δ 線が発生した場合などは、その輸送によって 1 次粒子の軌道から離れた位置にエネルギーが付与されることが知られており、本セクションを利用することでその影響を調べることが可能となります。δ 線生成断面積は、Butts と Katz の式[29] より計算し、そのエネルギーや角度を決定する際は相対論を考慮しています

真空 (void) 以外の領域毎に、δ 線を発生させるしきい値エネルギー $E_{\rm th}$(単位は MeV) を決めることができ、この値より高いエネルギーの δ 線を実際に 2 次粒子として発生させます。$E_{\rm th}$ 以下の δ 線の寄与は通常の LET で評価しており、設定できる $E_{\rm th}$ の最小値は 0.001MeV(=1keV) です。ただし、$E_{\rm th}$ を 10keV 以下に設定したり、δ 線を発生させる領域の厚さをおおよそ 10μg/cm^2 以下に設定すると、荷電粒子の挙動が多少変化しますのでご注意下さい。具体的には、荷電粒子の実効的な阻止能が、δ 線を多く放出しすぎることにより高くなってしまいます。[delta ray] セクションを利用しない場合は、$E_{\rm th}$ にデフォルトの値として $1.e+10$ が入っており、事実上 δ 線は発生しません。領域番号と $E_{\rm th}$ はそれぞれ `reg`, `del` で指定します。その書式は以下の例の通りです。

```
[ delta ray ]
    reg     del
      1     0.1
     11     1.0
    ....   ....
    ....   ....
```

同じ値の領域をまとめて書く、`( { 2 - 5 } 8 9 )` という書式も使えます。ただし、単一の数字で無い場合は必ず `(  )` で括ってください。しかし、`( 6 < 10[1 0 0] < u=3 )` などの lattice, universe 構造は使えません。領域番号 (`reg`) としきい値エネルギー (`del`) の順番を変えたいときは、`del reg` とします。読み飛ばしコラム用の `non` も使えます。GG の場合も、`cell` でなく `reg` を使ってください。

[29] J. J. Butts and R. Katz, “Theory of RBE for Heavy Ion Bombardment of Dry Enzymes and Viruses”, Radiation Research 30, 855-871 (1967).

4.13 [Super Mirror] セクション

このセクションでは、低エネルギーの中性子のスーパーミラーによる反射の機能を定義します。ここでは、以下のような半経験的なスーパーミラーの反射率を仮定します。

$$R = \begin{cases} R_0 & \text{if} \quad Q \le Q_c \\ \frac{1}{2}R_0\,(1 - \tanh\,[(Q - mQ_c)/W])\,(1 - \alpha(Q - Q_c)) & \text{if} \quad Q > Q_c \end{cases}$$

ここで Q は、散乱ベクター (Å^{-1}) で、次のように定義されます。

$$Q = |\mathbf{k}_i - \mathbf{k}_f| = \frac{4\pi \sin\theta}{\lambda}$$

m は、ミラーの物質とレイヤーの数などに依存するパラメーターです。Q_c は、一層のレイヤーによる臨界散乱ベクター、これ以上の Q で、反射率は α の傾斜で直線的にカットオフ値 $Q = mQ_c$ まで減少する。そのときのカットオフの幅は W で表される。

これらのパラメーターは、このセクションで次のように定義される。

```
[ Super Mirror ]
   r-in          r-out    mm   r0    qc      am    wm
   {2001-2020}   3001     3    0.99  0.0217  3.0   0.003
   2500          3500     3    0.99  0.0217  3.0   0.003
   2600          3600     3    0.99  0.0217  3.0   0.003
   ....           ....     ..   ...   ....    ...   ...
   ....           ....     ..   ...   ....    ...   ...
   ....           ....     ..   ...   ....    ...   ...
```

反射の面は、`r-in` が入射領域、`r-out` が反射体領域の間の面で定義されます。

同じ値の領域をまとめて書く、`( { 2 - 5 } 8 9 )` という書式も使えます。また、`( 6 < 10[1 0 0] < u=3 )` などの lattice, universe 構造も 指定できます。

上の標識のその他のパラメーターは、m を `mm` で、R_0 を `r0`、Q_c を `qc` (Å^{-1})、α を `am` (Å)、W を `wm` (Å^{-1}) で定義します。

このスーパーミラーの反射は、10eV 以下の中性子、また、$\sin\theta > 0.001$ の時に制限されます。後者は、表面の粗さに因るものです。

4.14 [Elastic Option] セクション

このセクションでは、核データを用いる低エネルギー中性子の弾性散乱について、ユーザー定義の角分布を与えるパラメータを指定します。ユーザー定義のサブルーティンのサンプルとして、usrelst1.f と usrelst2.f が用意してあります。これらは、パラメータセクションで、usrelst=1, 2 で切り替えることができます。

書式は、以下のように、このオプションを適用する領域番号と、ユーザー定義サブルーティンで用いる 4 つの定数を定義することができます。

```
[ Elastic Option ]
            reg       c1       c2       c3       c4
              1        5        1      3.3      0.4
              2        1        1      1.1      0.7
              3        3        1      0.3      0.8
            ....     ...      ...      ...      ...
            ....     ...      ...      ...      ...
```

物質番号 (reg) と (c1 c2 c3 c4) の順番を変えたいときは、reg c3 c2 c1 c4 の様に定義行で変えます。読み飛ばしコラム用の non も使えます。同じ名前をまとめて書く、{ 4 - 7 } という書式も使えます。ただし、({ 4 - 7 } 9 10) という表式は使えません。

現在、サンプルサブルーティンとして入っている、usrelst1.f と usrelst2.f は、それぞれ、前者がデータベースを用いた Bragg 散乱の角分布、後者が、任意の関数形を用いた角分布が記述できるようになっています。

4.15 [Data Max] セクション

このセクションでは、マテリアルの核種ごとにライブラリー利用の上限エネルギー dmax を設定することができます。このセクションは、6 つまで定義することができます。

```
[Data Max]
  part = neutron proton
  mat   nucleus  dmax
   all   Fe      20
    5    all     50
    3    56Fe    150
```

最初の行に part=の書式で輸送粒子を指定します。省略はできません。また、ここで定義できる粒子は、version 2.86 において、neutron と proton だけです。

マテリアルごとに指定するときは mat コラムを用います。ここにはマテリアル番号、もしくは、all を使うことができます。nucleus コラムで核種を指定します。56Fe、もしくは、26056 の書式で指定します。Fe、26000 などの質量を指定しない場合は、その核種の全ての同位体を指定することになります。また、all も使うことができます。dmax のコラムでその核種のライブラリー利用の上限エネルギー (MeV) を指定します。(mat) (nucleus) (dmax) の順番は可変です。また、読み飛ばしのコラム non も使えます。

同じ核種が重ねて定義されている場合は、後出が優先されます。

[parameters] セクションで定義される、dmax(1) と dmax(2) は、このセクションで定義される dmax の最大値とすることが望まれます。

[parameters] セクションで、kmout=1 を指定すると、file(6) (D=phits.out) にマテリアル毎に各核種の dmax が出力されます。

4.16 [Frag Data] セクション

このセクションでは、ユーザー指定断面積読み込み機能について定義できます。この機能を使用することにより、特定の粒子間の核反応イベントをユーザーが指定した断面積データを用いて模擬できます。ただし、本機能で指定した粒子間、エネルギー領域にある反応に対しても、核データライブラリを使用する設定になっている場合は、データライブラリにある情報が優先的に使用されます。また、指定できる入射粒子は、陽子や中性子、原子核のみです。光子や電子は指定できないのでご注意ください。

書式は、以下のようになります。

```
[ Frag Data ]
  opt     proj     targ     file
    0     12C      16O      DDX_12C-16O.dat
    1     proton   63Cu     DDX_p-63Cu.dat
  ....     ...      ...      ...      ...
  ....     ...      ...      ...      ...
```

opt が 0 の場合は指定した断面積を使用しません。これが 1 の場合は、proj と targ で指定した粒子間の核反応を file で指定したファイルにある断面積データを基に模擬します。opt が 2,3 の場合は整備中です。opt が 4 の場合は、入射エネルギーや放出角度、放出エネルギーについて、与えらたデータを基に単純な外挿を行います。ただし、この外挿オプションは、以下に示す neo が正で、nag が 0 でない場合しか使用できません。

file で指定する断面積データファイルの書式は、以下のようになります。

projectile
target
nei
ein(1) ein(2) ein(3) ..... ein(nei+1)
totxs(1) totxs(2) totxs(3) ..... totxs(nei+1)
neo
eout(1) eout(2) eout(3) ..... eout(neo+1)
nag
angle(1) angle(2) angle(3) ..... angle(nag+1)
nfrg
frag(1) frag(2) frag(3) ..... frag(nfrg)

proxs(1,1) proxs(1,2) proxs(1,3) ..... proxs(1,nfrg)
ddx(1,1,1,1) ddx(1,1,1,2) ddx(1,1,1,3) ..... ddx(1,1,1,nag)
ddx(1,1,2,1) ddx(1,1,2,2) ddx(1,1,2,3) ..... ddx(1,1,2,nag)
. . . . . .
ddx(1,1,neo,1) ddx(1,1,neo,2) ddx(1,1,neo,3) ..... ddx(1,1,neo,nag)

（続き）

ddx(1,2,1,1) ddx(1,2,1,2) ddx(1,2,1,3) ..... ddx(1,2,1,nag)
ddx(1,2,2,1) ddx(1,2,2,2) ddx(1,2,2,3) ..... ddx(1,2,2,nag)
. . . . . .
ddx(1,2,neo,1) ddx(1,2,neo,2) ddx(1,2,neo,3) ..... ddx(1,2,neo,nag)

. . . . . .
. . . . . .

ddx(1,nfrg,1,1) ddx(1,nfrg,1,2) ddx(1,nfrg,1,3) ..... ddx(1,nfrg,1,nag)
ddx(1,nfrg,2,1) ddx(1,nfrg,2,2) ddx(1,nfrg,2,3) ..... ddx(1,nfrg,2,nag)
. . . . . .
ddx(1,nfrg,neo,1) ddx(1,nfrg,neo,2) ddx(1,nfrg,neo,3) ..... ddx(1,nfrg,neo,nag)

proxs(2,1) proxs(2,2) proxs(2,3) ..... proxs(2,nfrg)
ddx(2,1,1,1) ddx(2,1,1,2) ddx(2,1,1,3) ..... ddx(2,1,1,nag)
ddx(2,1,2,1) ddx(2,1,2,2) ddx(2,1,2,3) ..... ddx(2,1,2,nag)
. . . . . .
ddx(2,1,neo,1) ddx(2,1,neo,2) ddx(2,1,neo,3) ..... ddx(2,1,neo,nag)

. . . . . .
. . . . . .
. . . . . .
. . . . . .

proxs(nei+1,1) proxs(nei+1,2) proxs(nei+1,3) ..... proxs(nei+1,nfrg)
ddx(nei+1,1,1,1) ddx(nei+1,1,1,2) ddx(nei+1,1,1,3) ..... ddx(nei+1,1,1,nag)
ddx(nei+1,1,2,1) ddx(nei+1,1,2,2) ddx(nei+1,1,2,3) ..... ddx(nei+1,1,2,nag)
. . . . . .
ddx(nei+1,1,neo,1) ddx(nei+1,1,neo,2) ddx(nei+1,1,neo,3) ..... ddx(nei+1,1,neo,nag)

. . . . . .
. . . . . .

ddx(nei+1,nfrg,1,1) ddx(nei+1,nfrg,1,2) ddx(nei+1,nfrg,1,3) ..... ddx(nei+1,nfrg,1,nag)
ddx(nei+1,nfrg,2,1) ddx(nei+1,nfrg,2,2) ddx(nei+1,nfrg,2,3) ..... ddx(nei+1,nfrg,2,nag)
. . . . . .
ddx(nei+1,nfrg,neo,1) ddx(nei+1,nfrg,neo,2) ddx(nei+1,nfrg,neo,3) ..... ddx(nei+1,nfrg,neo,nag)

最初に、projectile と target において入射粒子と標的粒子のそれぞれを指定します。nei は入射エネルギーの分点数で、その次の行に nei+1 点の入射エネルギー（ein: 単位は MeV/u）を指定します。neo は放出粒子のエネルギー分点数で、その次の行に neo+1 点の放出エネルギー（eout: 単位は MeV）と、更にその次の行に同数の全反応断面積（totxs: 単位は mb）を指定します。ここで totxs が 0 の場合は、`icxsni` や `icrhi` で指定する全反応断面積モデルにより得られた値が使われます。 nag は放出角度の分点数で、その次の行に nag+1

点の角度（angle）を指定します。nag>0 の場合はラジアン単位 (rad)、nag<0 の場合は度数単位 (degree) で angle を与えます。nag=0 の場合は等方分布を仮定します。nfrg は放出粒子の数で、その次の行に nfrg 個の放出粒子を指定します。proxs は各放出粒子のある入射エネルギーに対する生成断面積 (mb) で、その次の行から 2 重微分断面積（ddx: 単位は mb/MeV/sr）を並べます。1 つの proxs に対して neo×nag 点の 2 重微分断面積のデータが必要で、それを 1 単位として nei+1 個のデータを並べます。放出粒子を複数指定する場合 (nfrg>1) は、proxs から始まるデータ群が nfrg で指定した数だけ必要となります。

また、neo を書くべき行で `model` と書いた場合は、PHITS に組み込まれている核反応モデルが使用されます。その場合は、微分断面積のデータを与える必要はありません。

List 4.26 ● 断面積データファイルの例

```
 1:  proton
 2:  63Cu
 3:  1
 4:  10.0    100.0
 5:  1000.0  500.0
 6:  3
 7:  1.0  10.0  50.0  100.0
 8:  -6
 9:  0.0  30.0  60.0  90.0  120.0  150.0  180.0
10:  1
11:  neutron
12:  300.0
13:  10.0  10.0  10.0  10.0  10.0  10.0
14:  15.0  13.0  12.0  11.0  10.0  10.0
15:  10.0  11.0  10.0  11.0  10.0  10.0
16:  0.0
17:   5.0   5.0   5.0   5.0   5.0   5.0
18:  10.0   8.0   7.0   6.0   5.0   5.0
19:   5.0   6.0   5.0   6.0   5.0   5.0
```

List 4.26 に断面積データファイルの一例を示します。1,2 行目にそれぞれ入射粒子として proton、標的粒子として ^{63}Cu を指定しています。3 行目がエネルギー分点数で、この例では nei=1 としています。4,5 行目において入射エネルギーと全反応断面積を nei+1 点ずつ与えています。6 行目は放出粒子のエネルギー分点数です。この例では neo=3 としており、7 行目で放出エネルギーとして、1.0, 10.0, 50.0, 100.0MeV の値を指定しています。8 行目は角度の分点数です。これをマイナスにし、9 行目で与える角度の情報を度数単位で与えています。10 行目で放出粒子として取り扱いたい数を決めており、11 行目でその粒子 (=neutron) を指定しています。12 行目に入射エネルギーが 10.0MeV の場合の neutron 生成反応断面積を与えており、13,14,15 行目でこの入射エネルギーにおける 2 重微分断面積を与えています。各行は 7 行目で決めたエネルギービンに対応しており、各列は 9 行目で決めた角度ビンに対応しています。基本的に、12 行目の生成反応断面積には、13,14,15 行目の微分断面積を積分した量を与えてください。一致しない場合は、生成反応断面積の値を使って規格化されます。16,17,18,19 行目は、入射エネルギーが 100.0MeV の場合の生成反応断面積と 2 重微分断面積です。16 行目の生成反応断面積は 0 にしていますが、この場合、生成反応断面積として、17,18,19 行目の微分断面積を内挿して積分した値を使用します。

neo をマイナスで指定することで、放出粒子のエネルギーを離散的に与えることができます。この場合、eout の個数は neo としてください。

neo を 0 にすると、放出粒子のエネルギースペクトルをガウス分布で与えることができます。この場合は、ddx の箇所にガウス分布の平均値と分散の値を MeV で与えてください。

4.17 [Importance] セクション

このセクションでは、GG の cell の importance を定義します。定義されない領域もしくは粒子は、1.0 にセットされます。

このセクションは、6 つまで定義することができます。書式は、

```
    [ I m p o r t a n c e ]
       part =  proton  neutron
        reg               imp
          1             1.000000
         11             5.000000
( { 2 - 5 } 8 9 )       2.000000
   ( 11 12 15 )         3.000000
( 6<10[1 0 0]<u=3 )     6.000000
       ....             ........
       ....             ........
```

最初の行に `part =` の書式で粒子を指定します。省略した時のデフォルトは、`part = all` で全粒子です。`part =` の書式の詳細はタリーの粒子指定と同じ書式です。しかし、指定できるのは itype としての区別だけです。それぞれの原子核などは、個別には指定できません。

領域番号 (`reg`) と (`imp`) の順番を変えたいときは、 `imp reg` とします。読み飛ばしコラム用の `non` も 使えます。GG の場合も、`cell` でなく `reg` を使ってください。

同じ値の領域をまとめて書く、`( { 2 - 5 } 8 9 )` という書式も使えます。また、`( 6 < 10[1 0 0] < u=3 )` などの lattice, universe 構造も 指定できます。ただし、単一の数字で無い場合は必ず `(  )` で括ってください。

階層構造のある場合の importance の指定の仕方は、 各階層の importance の積を最下層の importance とします。ただし、最下層の特定のセルの importance の指定が、上の書式で可能です。この書式を用いれば lattice のひとつひとつに 異なる importance を定義することも可能です。上の定義で、同じセルが二重に定義された時は、初めに定義された値が採用されます。

透過性のある粒子に、具体的にはニュートリノですが、importance を強く設定されていると、計算時間が膨大になります。この危険を避けるために、part=all で指定してもニュートリノは省かれていますので、注意して下さい。

階層のあるセルの importance の指定は、幾つかの書き方が とり得ます。例えば、最下層のセル 5 6 7 が共に上位のセル 11 12 13 に含まれる時、

```
1:   [ Importance ]
2:      reg            imp
3:    ( 5 6 7 < 11 )   2.0
4:    ( 5 6 7 < 12 )   4.0
5:    ( 5 6 7 < 13 )   8.0
6:    ( 11 12 13 )     1.0
```

で定義される importance と、

```
1:   [ Importance ]
2:      reg            imp
3:    ( 5 6 7 )        1.0
4:      11             2.0
5:      12             4.0
6:      13             8.0
```

は、同じ結果を与えます。ただし、後者の場合、importance のサマリーでセル 5 6 7 の importance の値は、1.0 と表示されます。

4.18 [Weight Window] セクション

このセクションでは、ウエイトウインドウを定義します。このセクションは、6 つまで定義することができます。書式は、

```
    [ Weight Window ]
       part =  proton  neutron
        eng =  5
      ( tim =  5 )
               6.00e-7  3.98e-1  1.00e+0  7.00e+0  5.00e+4
      reg                ww1        ww2        ww3
        1              0.010000   0.100000   0.001000
       11              0.005000   0.050000   0.000300
( { 2 - 5 } 8 9 )      0.001000   0.010000   0.000100
   ( 11 12 15 )        0.000500   0.005000   0.000030
( 6<10[1 0 0]<u=3 )    0.000010   0.001000   0.000010
      ....             ........   ........   ........
                         ww4        ww5
                       0.010000   0.100000
                       0.005000   0.050000
                       0.001000   0.010000
                       0.000500   0.005000
                       0.000010   0.001000
                       ........   ........
```

最初の行に `part =` の書式で粒子を指定します。省略した時のデフォルトは、`part = all` で全粒子です。`part =` の書式の詳細はタリーの粒子指定と同じ書式です。しかし、指定できるのは itype としての区別だけです。それぞれの原子核などは、個別には指定できません。

次に、エネルギーもしくは、時間のメッシュを定義します。`eng =` もしくは `tim =` でメッシュの個数を、次行に、メッシュの値 $(e_1, e_2, e_3, ...)$ を定義します。このメッシュ毎のウエイトウインドウの下限値を以下に定義します。各下限値は、ww1, ww2, ww3, ... でエネルギーもしくは時間メッシュを指定します。ここで wwi は、$e_{i-1} < E < e_i$ のメッシュのウインドウ下限値です。$e_0 = 0, t_0 = -\infty$ を仮定しています。`eng =` もしくは `tim =` の定義がない場合は、エネルギーもしくは、時間のメッシュを仮定せず、ひとつの領域とします。この場合は、ww1 だけを定義します。

領域番号 (`reg`) は、最初のコラムに固定です。上の例題の様に、ひとつの表に、ww1, ww2, ... が書き切れない時は、次の表に書き足すことができます。この時、最初の領域番号 (`reg`) のコラムは、省略できます。また、領域番号の並びは、上の表と同じと仮定しています。 読み飛ばしコラム用の non は、使えます。GG の場合も、`cell` でなく `reg` を使ってください。

同じ値の領域をまとめて書く、`( { 2 - 5 } 8 9 )` という書式も使えます。また、`( 6 < 10[1 0 0] < u=3 )` などの lattice, universe 構造も 指定できます。ただし、単一の数字で無い場合は必ず `(  )` で括ってください。

透過性のある粒子に、具体的にはニュートリノですが、importance を強く設定されていると、計算時間が膨大になります。この危険を避けるために、part=all で指定してもニュートリノは省かれていますので、注意して下さい。

4.19 [Forced Collisions] セクション

強制衝突 (forced collisions) は、衝突の確率を人為的に上げてタリーの効率を上げたり、衝突の確率の低い薄いターゲットなどを解析するときに用います。強制衝突の領域に入った指定された粒子は、ふたつに分離されます。ひとつはその領域の透過確率をその重みにかけた粒子で、次の領域まで透過させます。もうひとつは、(1 - 透過確率) を乗じた重みを持った粒子で、この強制衝突の領域で強制的に衝突を起こさせます。衝突位置は、断面積に従って確率的に決定します。

強制衝突の領域と強制衝突係数をこのセクションで指定します。定義されない領域の強制衝突係数は、ゼロにセットされます。

このセクションは、6 つまで定義することができます。その書式は、

```
    [ Forced Collisions ]
        part =  proton  neutron
        reg               fcl
          1             1.000000
         11             0.500000
( { 2 - 5 } 8 9 )       0.200000
   ( 11 12 15 )         0.300000
( 6<10[1 0 0]<u=3 )    -0.500000
        ....            ........
        ....            ........
```

最初の行に part = の書式で粒子を指定します。省略した時のデフォルトは、part = all で全粒子です。ただし、電子、陽電子は強制衝突はできません。part = の書式の詳細はタリーの粒子指定と同じ書式です。しかし、指定できるのは itype としての区別だけです。それぞれの原子核などは、個別には指定できません。

領域番号 (reg) と (fcl) の順番を変えたいときは、 fcl reg とします。読み飛ばしコラム用の non も使えます。GG の場合も、cell でなく reg を使ってください。

同じ値の領域をまとめて書く、({ 2 - 5 } 8 9) という書式も使えます。

また、(6 < 10[1 0 0] < u=3) などの lattice, universe 構造も 指定できます。ただし、単一の数字で無い場合は必ず ( ) で括ってください。

この書式を用いれば lattice のひとつひとつに 異なる forced collision factor を定義することも可能です。上の定義で、同じセルが二重に定義された時は、初めに定義された値が採用されます。

強制衝突の係数 fcl は次の様な意味を持ちます。まず、ゼロは強制衝突をさせません。|fcl| > 1 はエラーです。|fcl| ≤ 1 の時は、強制衝突の確率を |fcl| 倍します。その分、強制衝突粒子の重みを 1/|fcl| 倍します。

強制衝突領域での多重散乱と weight cutoff との関係では、次のふたつのオプションがあります。fcl < 0 の時、強制衝突による生成粒子は通常の衝突をさせます。その時 weight cutoff は行いません。fcl > 0 の時、強制衝突による生成粒子も更に強制衝突をさせます。この時、そのたびに weight cutoff を行います。また、タリーは weight cutoff の前に行います。この関係を良く理解して weight cutoff と強制衝突の係数を決めないと、強制衝突はさせたが出てきた粒子は全て殺されてしまうということもあり得ます。例えば、強制衝突で発生させた 2 次粒子の軌跡や強制衝突領域から離れた場所における粒子の情報をタリーする場合が該当します。この場合、[parameters] セクションにおいてウェイトカットオフ wc1(i) を低く設定する必要があります。

4.20 [Brems Bias] セクション

このセクションでは、物質毎に bremsstrahlung 過程のエネルギーバイアスを定義します。bremsstrahlung 過程では、大量の低エネルギー光子が生成されますが、しばしば、高エネルギー光子により関心がある場合があります。そのような時に、エネルギー毎のバイアスを定義して、目的のエネルギー領域の光子の統計を上げることができます。

書式は、

```
[ Brems Bias ]
    material =  all or number of materials
       m1   m2   m3   ........
    num              bias
    {1-45}            1.0
      46             2.0
      47             3.0
      48             4.0
      49             5.0
```

まず、バイアスを設定する物質を定義します。all の場合は次の行は必要ありません。数を指定した場合は、その数だけの物質番号を次の行に定義します。

バイアスは、1 から 49 のグループについて定義します。大きい番号が高エネルギーを表します。バイアス値は相対値です。

グループ番号 (num) とバイアス (bias) の順番を変えたいときは、 `bias num` とします。読み飛ばしコラム用の non も使えます。

4.21 [Photon Weight] セクション

このセクションでは、GG の cell の光子生成ウエイトを定義します。

セル i の光子生成ウエイトの値 W_i が正の時は、生成 photon のウエイトが $W_i \times I_s/I_i$ 以上の時に生成されます。それ以下の時はロシアンルーレットが施されます。ここで I_s, I_i は、ソースセルと生成場所の中性子のインポータンスです。

セル i の光子生成ウエイトの値 W_i が負の時は、生成 photon のウエイトが $W_i \times W_s \times I_s/I_i$ 以上の時に生成されます。それ以下の時はロシアンルーレットが施されます。ここで W_s は、生成する中性子の反応前のウエイトです。

もし、$W_i = 0$ なら、1 個の光子が中性子反応から生成されます。もし、$W_i = -1.0e6$ であれば、このセルでは光子の発生は無視されます。デフォルト値は、$W_i = -1$ です。

```
    [ Photon Weight ]
        reg             pwt
          1             0.1
         11             0.3
( { 2 - 5 } 8 9 )       0.5
   ( 11 12 15 )         0.1
         16             0.9
        ....            ........
        ....            ........
```

同じ値の領域をまとめて書く、`( { 2 - 5 } 8 9 )` という書式も使えます。ただし、単一の数字で無い場合は必ず `(  )` で括ってください。しかし、`( 6 < 10[1 0 0] < u=3 )` などの lattice, universe 構造は、使えません。

領域番号 (`reg`) とウエイト (`pwt`) の順番を変えたいときは、`pwt reg` とします。読み飛ばしコラム用の `non` も使えます。GG の場合も、`cell` でなく `reg` を使ってください。

4.22　[Volume] セクション

このセクションでは、GG の cell の体積 (cm^3) を定義します。この量は、以下のタリーセクションで用いられます。両者で定義されない領域は、1.0 cm^3 にセットされます。書式は、

```
[ V o l u m e ]
    reg                vol
     1              1.000000
    11              5.000000
( { 2 - 5 } 8 9 )   2.000000
  ( 11 12 15 )      3.000000
    16              6.000000
    ....            ........
    ....            ........
```

同じ値の領域をまとめて書く、`( { 2 - 5 } 8 9 )` という書式も使えます。ただし、単一の数字で無い場合は必ず `(  )` で括ってください。しかし、`( 6 < 10[1 0 0] < u=3 )` などの lattice, universe 構造は、使えません。タリーで異なる universe でのセルの体積を細かく定義する時は、タリーセクションの中の volume 定義を用いてください。

領域番号 (`reg`) と体積 (`vol`) の順番を変えたいときは、 `vol reg` とします。読み飛ばしコラム用の non も使えます。GG の場合も、`cell` でなく `reg` を使ってください。

4.23 [Multiplier] セクション

このセクションで定義した係数は、[t-track] セクションの multiplier サブセクションで利用することができます。エネルギーの関数として係数を定義し、[t-track] タリーの結果にその係数を乗じます。例えば、任意の線量換算係数を用意し、その値を用いた場合の線量を評価する、といったことに利用できます。

ひとつの [multiplier] セクションにひとつの係数のセットを定義でき、用意できる [multiplier] セクションの数の上限は 100 です。書式は以下の様になります。

```
[ Multiplier ]
  number = -201
  interpolation = log
  ne = 10
       20.0       2.678
       30.0       7.020
       50.0       18.50
       100.0      24.26
       200.0      16.13
       500.0      10.51
       1000.0     10.55
       2000.0     10.98
       5000.0     12.10
       10000.0    12.45
```

最初に、この係数をタリーの中で使うときの ID 番号を number で指定します。指定できる ID 番号は −299 から −200 までの負の数です。次に、この係数のテーブルから係数を抽出するときの内挿方法を interpolation で決定します。log の場合は log-log 内挿、lin のときは linear-linear 内挿です。そして、エネルギーの群数を ne で指定し、その下に ne の数だけエネルギー点と係数の値をそれぞれ定義します。ただし、エネルギーのデータは昇順で書く必要があります。

定義した係数をタリーで利用する場合は、[t-track] セクションの multiplier サブセクションを使います。基本的な表式は (C k) です。C を規格化定数、k を ID 番号 (ただし $k < 0$) として指定します。具体的には、[t-track] セクションにおいて、以下の様な書式で使用します。

```
multiplier =  number of material
      part = neutron
      emax = 1000
    mat     mset1          mset2
      1   ( 1    -201 )  ( 2 -202 )
      2   ( 1.2  -201 )  ( 3 -202 )
    ....   ....     ....   ....
    ....   ....     ....   ....
```

multiplier =で指定する物質の数を決定します。all も可能ですが、この時は、下の mat の項でも all を用います。part =で係数を乗じる粒子を指定します。6 個まで複数指定可能で、all も利用できます。省略すれば all を指定したことになります。ただし、指定した粒子以外の寄与はゼロになります。emax =で係数を乗じるエネルギーの上限を設定します。省略した場合は、[multiplier] セクションで指定したエネルギーの最大値がセットされます。それ以上では、その最大値に対する値が入ります。mat が係数を乗じる物質番号です。mset1, mset2 は multiplier セットの指定です。このセットは全部で 6 つまで指定可能で、それぞれのセット毎に結果が出力されます。また、ひとつの [t-track] セクション内に複数の multiplier サブセクションを指定できます。ただし、その時の multiplier セットの数は等しくなければなりません。

4.24 [Mat Name Color] セクション

このセクションでは、gshow タリー、3dshow タリーで領域を表示する時の、物質番号毎に物質の名前と色、表示の文字の大きさを指定します。デフォルトは、名前は物質番号、色は物質の出現順に割り振ります。書式は、

```
[ Mat Name Color ]
            mat       name        size                color
              0       void           1            lightgray
              1        air         0.5          yellowgreen
              2     {mat 2}           2         orangeyellow
              3     {mat 3}           2  { 0.067 0.600 1.00 }
      { 4 - 7 }          Fe           3            mossgreen
     ....           ........
     ....           ........
```

物質番号 (mat) と (name size color) の順番を変えたいときは、mat color size name の様に定義行で変えます。読み飛ばしコラム用の non も使えます。name color の何れかが定義されなければなりません。省略された場合は、デフォルトと見なされます。同じ名前、色の物質をまとめて書く、{ 4 - 7 } という書式も使えます。ただし、({ 4 - 7 } 9 10) という表式は使えません。文字の大きさは、デフォルトを 1 とした相対値で指定します。

名前に空白が入る時は、{ } で括ってください。また、名前の中で括弧 ( ) を使うときは、¥(¥) としてください。名前の中で括弧 { } は使えません。名前として定義できる文字数の上限は 30 です。

[cell] セクションにおいて同一の物質番号で密度の違う領域を複数定義した場合、デフォルト設定 (matadd=1) では、最初のもの以外の密度の物質に対して新しい物質番号を付与します。この番号は、file(6) (D=phits.out) の [material] セクションに出力されるので、その値を mat としてご使用ください。また、[parameters] セクションにおいて matadd=0 とすることで、同じ物質番号が割り振られるようになりますが、この機能は本セクションで指定する物質番号には影響がありません。この番号は file(6) の最初に Warning として出力されますので、その値をご参照ください。

色の指定は、$A_N G_E L$ の書式に準じます。記号 (r bbb yy)、名前 (red orange blue)、また、HSB 数値 H(色相) S(彩度) B(明度) で指定できます。HSB 数値指定の時は { } で括ってください。HSB 数値が 1 個しか指定されない場合は、彩度、明度は 1 に設定されます。

次頁に、色指定の記号、名前、HSB 数値の表を示します。

表 72: グレースケール

記号	HSB定義	出力	名前
W	–1.0		white
O	–0.8		lightgray
K	–0.6		gray
J	–0.4		darkgray
F	–0.2		matblack
E	–0.0		black

表 73: 記号による色指定

記号	HSB定義	出力	名前
R	1.000		red
RR	0.933		orange
RRR	0.867		–
Y	0.800		yellow
YY	0.733		–
YYY	0.667		–
G	0.600		green
GG	0.533		–
GGG	0.467		–
C	0.400		cyan
CC	0.333		–
CCC	0.267		–
B	0.200		blue
BB	0.133		violet
BBB	0.067		magenta

表 74: 名前、数値による色指定

名前	出力	HSB定義		
darkred		1.000	1.000	0.600
red		1.000	1.000	1.000
pink		1.00	0.500	1.000
pastelpink		0.900	0.500	1.000
orange		0.933	1.000	1.000
brown		0.900	1.000	0.500
darkbrown		0.900	1.000	0.300
pastelbrown		0.900	0.600	0.500
orangeyellow		0.867	1.000	1.000
camel		0.800	0.700	0.700
pastelyellow		0.800	0.700	1.000
yellow		0.800	1.000	1.000
pastelgreen		0.700	0.600	1.000
yellowgreen		0.700	1.000	1.000
green		0.600	1.000	1.000
darkgreen		0.600	1.000	0.600
mossgreen		0.500	1.000	0.300
bluegreen		0.500	1.000	1.000
pastelcyan		0.400	0.400	1.000
pastelblue		0.250	0.400	1.000
cyan		0.400	1.000	1.000
cyanblue		0.400	1.000	0.500
blue		0.200	1.000	1.000
violet		0.133	1.000	1.000
purple		0.100	1.000	0.500
magenta		0.067	1.000	1.000
winered		0.002	0.800	0.700
pastelmagenta		0.067	0.600	1.000
pastelpurple		0.100	0.400	0.500
pastelviolet		0.133	0.400	1.000

4.25 [Reg Name] セクション

このセクションでは、[t-gshow]、[t-gshow]、[t-3dshow] で出力されるセル番号（数字）の表示を任意の名前に変更し、またその文字の大きさも変更できます。 他のタリーで gshow や rshow オプションを使用した場合の出力結果にも反映されます。 書式は、

```
[ Reg Name ]
            reg       name        size
              1       cover          1
              2        body        0.5
              3     {cell 2}          2
              4     {cell 3}          2
      { 5 - 8 }        tube           3
    ....            ........
    ....            ........
```

です。セル番号 (reg) と名前 (name) や文字の大きさ (size) の順番を変えたいときは、reg size name の様に定義行で変えます。読み飛ばしコラム用の non も使えます。name と size の何れかが定義されなければなりません。省略された場合は、デフォルトと見なされます。セル番号をまとめる { 4 - 7 } という書式も使えます。ただし、({ 4 - 7 } 9 10) という表式は使えません。名前に空白が入る時は、{ } で括ってください。このため、名前の中で括弧 { } は使えません。名前の中で括弧 (,) を使うときは、それぞれ ¥(, ¥) としてください。また、名前として定義できる文字数の上限は 30 です。文字の大きさは、デフォルトを 1 とした相対値で指定します。

4.26 [Counter] セクション

このセクションでは、タリーで用いるカウンターの設定をします。カウンターは 3 つ用意され、それぞれのカウンターで領域ごとに、その動作を定義します。カウンターの動作契機は 5 つあり、領域に入った時 (in)、領域を出た時 (out)、散乱（核反応）を起こした時 (coll)、核分裂を起こした時 (fiss)、境界で反射を起こした時 (ref) です。それぞれに、カウンターの進度 (-9999~9999) もしくは、ゼロセット (10000) を定義できます。カウンターは粒子に付随して、散乱によって生成される粒子は、親のカウンターを引き継ぎます。カウンターの容量は、-9999 から 9999 まででそれ以下、以上になる時は変化しません。カウンターは、`part =` を使って粒子毎に指定できます。また、`*part =` とすれば、カウンターを動作させない粒子を指定できます。書式は以下のようになります。、

```
[ C o u n t e r ]
   counter = 1
      part = neutron proton
    reg     in     out    coll    fiss     ref
      1      1   10000       0       1       0
     11      1   10000       0       0       0
   counter = 2
     *part = proton deuteron triton 3he alpha nucleus
    reg                        in     out    coll
     ( { 2 - 5 } 8 9 )         -1       0       1
   counter = 3
      part = 208Pb
    reg                        coll
     ( 11 12 15 )                 5
     ( 6<10[1 0 0]<u=3 )        100
    ....            ........
    ....            ........
```

領域番号 (`reg`) と (`in out coll fiss ref`) の順番を変えたいときは、`reg coll in out fiss ref` の様に定義行で変えます。読み飛ばしコラム用の `non` も使えます。`in out coll fiss ref` の内、少なくともひとつは定義されなければなりません。省略された場合は、ゼロ、つまりカウントなしと見なされます。数字は、カウンターの進度を表します。この数だけカウンターに加えられます。10000 は、ゼロセットを意味します。ソース粒子のカウンター初期値はゼロです。

同じ値の領域をまとめて書く、`( { 2 - 5 } 8 9 )` という書式も使えます。また、`( 6 < 10[1 0 0] < u=3 )` などの lattice, universe 構造も 指定できます。ただし、単一の数字で無い場合は必ず `(  )` で括ってください。

`part =` での粒子指定では、粒子は 20 個まで指定できます。原子核は、`208Pb` のように質量数を指定すればその核、`Pb` のように質量数を指定しなければ、`Pb` の同位体全体を指定することになります。

fiss を設定すると、散乱（核反応）が起こり、更に核反応として核分裂が起こった時だけカウンターが呼ばれます。核分裂が起こったかどうかの判定は、核データを使う時は核データの fission チャンネルが選択された時、反応モデルを使う時は統計崩壊過程において fission チャンネルが選ばれた時です。

4.27 [Timer] セクション

このセクションでは、各粒子が持っている時計のリセット、ストップをコントロールします。タイマーは、領域ごとに、その動作を定義します。タイマーの動作契機は 4 つあり、領域に入った時、領域を出た時、散乱を起こした時、境界で反射を起こした時です。それぞれに、ゼロリセット (-1)、ストップ (1)、何もしない (0) の動作が定義できます。書式は、

```
[ T i m e r ]
    reg     in     out    coll     ref
      1      0      -1       0       0
     11      1       0       0       0
    ....   ....    ....    ....    ....
    ....   ....    ....    ....    ....
    ....   ....    ....    ....    ....
```

領域番号 (reg) と (in out coll ref) の順番を変えたいときは、 reg coll in out ref の様に定義行で変えます。読み飛ばしコラム用の non も使えます。in out coll ref の内、少なくともひとつは定義されなければなりません。省略された場合は、ゼロ、つまり何もしません。

同じ値の領域をまとめて書く、({ 2 - 5 } 8 9) という書式も使えます。また、(6 < 10[1 0 0] < u=3) などの lattice, universe 構造も 指定できます。ただし、単一の数字で無い場合は必ず ( ) で括ってください。

5　タリー共通パラメータの書式

タリーセクションには、以下のセクションがあります。

表 75: タリーセクションの種類

name	説明
[t-track]	粒子の飛跡長 (track length) やフルエンスを導出するタリー。
[t-cross]	粒子の面横断回数やフルエンスを導出するタリー。
[t-point]	ある点や線上のフルエンスを導出するタリー。
[t-heat]	物質におけるエネルギー付与を導出するタリー。
[t-deposit]	荷電粒子の物質におけるエネルギー付与を導出するタリー。
[t-deposit2]	2 つの領域でのエネルギー付与の相関を出力するタリー。
[t-yield]	残留核の生成量を導出するタリー。
[t-product]	線源や核反応による生成粒子を導出するタリー。
[t-dpa]	原子あたりのはじき出し数 (DPA) を導出するタリー。
[t-let]	LET の関数として飛跡長や線量を導出するタリー。
[t-sed]	微小領域におけるエネルギー付与分布を導出するタリー。
[t-time]	Energy cut off と escape 粒子の個数を導出するタリー。
[t-star]	Star density を導出するタリー。
[t-dchain]	DCHAIN-SP 用入力ファイルを作成するタリー。
[t-wwg]	[weight window] のパラメータを出力するタリー。
[t-userdefined]	ユーザー定義による任意の物理量を導出するタリー。
[t-gshow]	ジオメトリ（仮想空間）を 2 次元で表示するタリー。
[t-rshow]	ジオメトリを 2 次元で物理量による色分けをして表示するタリー。
[t-3dshow]	ジオメトリを 3 次元で表示するタリー。

以下では、これらのタリーセクションで共通に使われるパラメーターの定義の仕方を説明します。

5.1　形状メッシュ

上に示した各タリーでは、データを収集する幾何形状の種類として、GG 領域メッシュ(reg)、r-z スコアリングメッシュ(r-z)、xyz スコアリングメッシュ(xyz) が全て使えます。 この 3 種類のメッシュを

```
mesh =  [ reg,  r-z,  xyz  ]
```

で定義します。

5.1.1　領域メッシュ

領域番号、もしくはセル番号で指定する領域メッシュは次の様に定義します。

```
mesh = reg
 reg = 1 2 3 4 5 ( 10 11 ) 50
```

領域番号、もしくは、セル番号は、空白区切りで指定します。

幾つかの領域をまとめたい時は、( ) を用います。このまとめた領域の識別番号はエコーに表示されます。

連続した領域番号では、次の様な書式も使えます。

```
mesh = reg
 reg = { 1 - 5 } ( 10 11 ) ( 6 < 10[1 0 0] < u=3 )
```

(n1 - n2) は、使えません。({ }) は可能ですが、{ ( ) } は不可です。(all) も可能です。また、(6 < 10[1 0 0] < u=3) などの lattice, universe 構造も 指定できます。この書式を用いれば lattice のひとつひとつを 個別にタリーすることも可能です。また、reg = all という書式で領域を定義すると全ての領域を対象とします。ただし、最下層でないセルは除かれます。

5.1.2 階層構造の領域と体積の定義

階層構造を含む領域を指定する際は、必ず ( ) で書式を括らなければなりません。階層は < で表します。中間の階層が lattice 構造を持つ場合は、そのセル番号の後に [ ] を用いてどの lattice か指定することができます。例えば、160[1:2 3:6 1:1] の様に指定した場合は、lattice 座標 (s, t, u) で表現した $1 \le s \le 2, 3 \le t \le 6, u = 1$ の範囲に含まれる $2 \times 4 \times 1 = 8$ の lattice を意味します。個別に幾つか指定する時はコンマを用い、160[1 3 4, 2 3 4, 3 3 4] とします。ひとつの階層の中で ( ) を用いると、それは括弧内の領域をまとめることを意味します。次の例題を見てみましょう。

List 5.1 ● mesh = reg の例題 (1)

```
mesh = reg
reg = (all)
      ({ 201 - 205 })
      ( 161 < 160[1:2 3:6 1:1] )
      ( (201 202 203 204) < (161 162 163 ) )
      ( ( 90 100 ) 120 < 61 ( 62 63 ) )
```

このメッシュ定義文をタリーで用いてエコーを取ると次のものが得られます。

List 5.2 ● mesh = reg の例題 (2)

```
mesh =  reg            # mesh type is region-wise
 reg = ( all ) ( { 201 - 205 } ) ( 161 < 160[ 1:2 3:6 1:1 ] ) ( (
       { 201 - 204 } ) < ( { 161 - 163 } ) ) ( ( 90 100 ) 120 < 61
       ( 62 63 ) )
volume                 # combined, lattice or level structure
non   reg      vol     # reg definition
 1  10001   8.1000E+01 # ( all )
 2  10002   5.0000E+00 # ( { 201 - 205 } )
 3  10003   1.0000E+00 # ( 161 < 160[ 1 3 1 ] )
 4  10004   1.0000E+00 # ( 161 < 160[ 2 3 1 ] )
```

```
11:        5  10005   1.0000E+00 # ( 161 < 160[ 1 4 1 ] )
12:        6  10006   1.0000E+00 # ( 161 < 160[ 2 4 1 ] )
13:        7  10007   1.0000E+00 # ( 161 < 160[ 1 5 1 ] )
14:        8  10008   1.0000E+00 # ( 161 < 160[ 2 5 1 ] )
15:        9  10009   1.0000E+00 # ( 161 < 160[ 1 6 1 ] )
16:       10  10010   1.0000E+00 # ( 161 < 160[ 2 6 1 ] )
17:       11  10011   4.0000E+00 # ( ( { 201 - 204 } ) < ( { 161 - 163 } ) )
18:       12  10012   2.0000E+00 # ( ( 90 100 ) < 61 )
19:       13  10013   1.0000E+00 # ( 120 < 61 )
20:       14  10014   2.0000E+00 # ( ( 90 100 ) < ( 62 63 ) )
21:       15  10015   1.0000E+00 # ( 120 < ( 62 63 ) )
```

入力では 5 つの領域の様に見えますが、エコーを見るとタリーを取る領域として 15 の領域が定義されていることが分かります。それぞれ単一の領域番号で表されないので、新たに 10000 番台の番号を付けてその体積と領域の定義文がエコーされます。この場合、[volume] セクションが無いので各セルの体積は 1 と 定義されています。

まず、`( all )` は、この例題では 81 個の最下層のセルが定義されているので、この領域の体積は 81 となっています。もし最下層のセルの体積が正確に `[volume]` セクションで 定義されていれば、この領域の体積は入力しなくても正確に与えられます。次に `({ 201 - 205 })` は、201 から 205 までの領域をまとめたものですから、体積として 5 がエコーされています。これも `[volume]` セクションで正確に体積が 定義されていれば、この領域の体積は入力しなくてもかまいません。

次に、`( 161 < 160[1:2 3:6 1:1] )` ですが、これは、160 の領域の中に 161 の領域が lattice として組まれています。ここでは、lattice 座標の s に関して 1 から 2、t に関して 3 から 6、u に関して 1 の各 lattice に含まれているセル番号 161 の計 8 個の領域でタリーを取ることを意味します。エコーでは、最下層の領域の数 1 が体積としてエコーされています。この様な場合は、次に示す体積定義文で体積を与えなければなりません。

次は、`( (201 202 203 204) < (161 162 163 ) )` ですが、これは、各階層に幾つかの領域が記述されていますが、全て括弧で括られてひとつの領域にされているので、全体としてもひとつの領域を表します。エコーの体積は、最下層の領域の和としているので正しくはありません。これも volume 定義文で与えなければなりません。

最後の `( ( 90 100 ) 120 < 61 ( 62 63 ) )` は、各階層に 2 つずつ独立な領域があるので計 4 個の領域に分割されます。それぞれの体積はやはり volume 定義文で与える必要があります。

volume 定義文は、インプットエコーの書式をそのまま使えます。書式は、

```
mesh = reg
 reg = 1 2 3 4 ( 5 < 12 ) ( {13 - 17} )
volume
 reg       vol
   1       1.0000
   2       5.0000
   3       6.0000
   4       1.0000
10001      6.0000
10002      5.0000
```

上の例題では、`reg` として、1 から 4 まではそのまま、`( 5 < 12 ) ( {13 - 17} )` は、10001 と 10002 を用います。この番号は、一度インプットエコーを取ると得られます。上の例題より分かるように、インプッ

トエコーの部分をそのままコピーし、体積の値だけを変更すれば済みます。

領域番号 (reg) と体積 (vol) の順番を変えたいときは、 vol reg とします。読み飛ばしコラム用の non も使えます。インプットエコーでは、non として通し番号が付けられています。axis = reg の時の タリーの出力の際にはこの番号が X 軸の値として出力されます。GG の場合も、cell でなく reg を使ってください。

最下層の領域である場合、その領域を指定する時は、(3000 < 3000[1:2 3:6 1:1]) の様に、同じ領域を階層構造の様に 2 度記述します。

5.1.3 r-z メッシュ

r-z スコアリングメッシュの場合、まず、円柱の中心の x, y 座標のオフセットを

```
mesh = r-z
  x0 = 1.0
  y0 = 2.0
```

の書式で定義します。これは省略可能でデフォルト値はゼロです。

次に、

```
mesh   = r-z
r-type = [1-5]
   ..........
   ..........
z-type = [1-5]
   ..........
   ..........
```

で始まるメッシュ定義文によって r と z とのメッシュを定義します。メッシュ定義文は後述します。

[t-track] タリーに関しては、既存の r, z メッシュに加え、θ メッシュを指定することができます。ここで、r, θ, z はそれぞれ円柱座標系における

r: 円の半径

θ: 円面における角度

z: 円柱の中心軸上の位置

であり、θ を定義する方位角の基線は x 軸です。[t-track] において θ に関するメッシュを指定する際は、a-type を用いてください。a-type=1,2,-1,-2 のタイプがありますが、これらの数字が正の場合は角度の単位を余弦で与え、負の場合は degree 単位で与えることになります。ただし、円柱の定義であるので、0 度から 360 度までしか指定できません。また、角度のメッシュを加えたことにより、角度を横軸にして出力できるようになりました。そのために axis として、rad と deg を指定できます。rad は radian で、deg は degree の単位で出力します。

5.1.4 xyz メッシュ

xyz スコアリングメッシュの場合、

```
mesh   = xyz
x-type = [1-5]
   ..........
   ..........
y-type = [1-5]
   ..........
   ..........
z-type = [1-5]
   ..........
   ..........
```

で始まるメッシュ定義文によって x, y, z のメッシュを定義します。メッシュ定義文は後述します。

5.2 エネルギーメッシュ

タリーのエネルギーメッシュは

```
e-type = [1-5]
   ..........
   ..........
```

で始まるメッシュ定義文によって定義します。DEPOSIT2 では、e1-type, e2-type で二つのエネルギーメッシュを定義します。メッシュ定義文は後述します。

5.3 LET メッシュ

タリーの LET メッシュは

```
l-type = [1-5]
   ..........
   ..........
```

で始まるメッシュ定義文によって定義します。メッシュ定義文は後述します。

5.4 時間メッシュ

タリーの時間メッシュは

```
t-type = [1-5]
   ..........
   ..........
```

で始まるメッシュ定義文によって定義します。メッシュ定義文は後述します。

5.5 角度メッシュ

cross, product タリーで用いる角度メッシュは

```
a-type = [1, 2, -1, -2]
   ..........
   ..........
```

で始まるメッシュ定義文によって定義します。この定義により、余弦 (cos) もしくは角度 (degree) を定義します。a-type を正で定義したときは余弦 (cos)、負で定義したときは角度 (degree) を表します。メッシュ定義文は後述します。

5.6 メッシュ定義文

メッシュ定義文には、e-type, t-type, x-type, y-type, z-type, r-type, a-type, l-type で始まる 8 種類がありますが、最初の一文字を除くと全て共通なので、ここでは e-type を例に解説します。ne を nx, ny, nz, ...、emin を xmin, ymin, zmin, ...、の様に読み換えてください。

5.6.1 メッシュタイプ

メッシュの指定の仕方は 5 種類あり、それを e-type = [1-5] で指定します。1 から 5 番までのそれぞれのメッシュタイプは以下のようなもです。

ただし、a-type 即ち、角度のメッシュでは、1, 2 (-1, -2) のメッシュタイプ しか使えません。以下に、それぞれのメッシュタイプのデータの定義の書式を示します。

5.6.2 e-type = 1 の場合

書式は、

表 76: メッシュタイプ

メッシュタイプ	内容
1	群数、分点をデータで与える。
2	群数と最小値、最大値を与え、線形で等分点が与えられる。
3	群数と最小値、最大値を与え、対数で等分点が与えられる。
4	メッシュ幅と最小値、最大値を与え、線形で等分点が与えられる。このとき、分点の最大値は与えた最大値もしくは最大値を超える最小の値となるように群数が決定される
5	最小値、最大値、メッシュ幅の対数値とを与え、対数で等分点が与えられる。このとき、分点の最大値は与えた最大値もしくは最大値を超える最小の値となるように群数が決定される

```
e-type = 1
    ne =  number of group
          data(1) data(2) data(3) data(4)
          data(5) data(6) data(7) data(8)
          .........
          .........
          data(ne+1)
```

です。この際、継続行のシンボル無しに複数行にまたがっても自動判別します。

5.6.3 e-type = 2, 3 の場合

この次に群数、最小値、最大値を次の様な書式で与えます。

```
e-type = 2, 3
    ne =  number of group
  emin =  minimum value
  emax =  maximum value
```

5.6.4 e-type = 4 の場合

この次にメッシュ幅、最小値、最大値を次の様な書式で与えます。

```
e-type = 4
  edel =  width of mesh
  emin =  minimum value
  emax =  maximum value
```

5.6.5 e-type = 5 の場合

この次にメッシュ幅、最小値、最大値を次の様な書式で与えます。

```
e-type = 5
  edel = log(  width of mesh  )
  emin =  minimum value
  emax =  maximum value
```

この場合、メッシュ幅は対数上での幅、即ち edel = log(M_{i+1} / M_i) です。

5.7 他のタリー定義文

5.7.1 粒子定義文

タリーの中で粒子を指定する時は、

```
part = proton neutron pion+ 3112 208Pb
```

のように空白で区切って定義するか、

```
part = proton
part = neutron
part = pion+
part = 3112
part = 208Pb
```

のように、定義文を繰り返すことも出来ます。粒子名の表式は表 4 を参照してください。kf コード番号での指定も可能です。

```
part = all
```

と定義すると、粒子すべての和を表わします。ひとつのタリーで粒子 6 個まで指定できます。それ以上の粒子を指定したい時は、複数のタリーセクションを用いてください。

複数の粒子をひとつのグループとしてタリーしたいときは、次のように ( ) が使えます。( ) の中の粒子は、最大 6 個まで指定できます。

```
part = ( proton neutron ) all pion+ 3112 208Pb
```

この場合、proton と neutron を合わせたものが、最初のグループとして出力されます。2 番目は、全ての粒子の和です。全体として 5 種類の粒子についての出力になります。原子核は、208Pb のように質量数を指定すればその核、Pb のように質量数を指定しなければ、Pb の同位体全体を指定することになります。

5.7.2 axis 定義文

出力データの x 軸を定義します。axis の種類は、タリーの種類とタリーの形状メッシュによりますが、

```
eng, reg, x, y, z, r, t, xy, yz, zx, rz,
cos, the, mass, charge, chart, dchain,
let, t-eng, eng-t, t-e1, e1-t, t-e2, e2-t, e12, e21
```

があります。

```
axis = eng
```

のように定義します。

ひとつのタリーで複数の axis が定義できます。

```
axis = eng  x  y
```

とするか、

```
axis = eng
axis = x
axis = y
```

のように表わすことも出来ます。ひとつの axis に対して、ひとつのファイルに結果が 出力されます。従って、複数の axis を定義した場合は、次のファイル定義文で同数のファイル名を定義する必要があります。なお、[t-yield] では、1 つのタリーに 1 つの axis しか指定できません。複数の axis に対して残留核収率を計算したい場合は、複数の [t-yield] タリーを作成してください。これは、統計誤差を正しく計算するためにバージョン 2.50 以降についた制限です。

5.7.3 file 定義文

出力ファイル名を定義します。書式は、

```
file = file.001  file.002  file.003
```

のように、パス名を含めてファイル名を書きます。axis を複数指定したときは、その数だけファイル名を指定します。このとき

```
file = file.001
file = file.002
file = file.003
```

のように一行にひとつずつ書くこともできます。

5.7.4 resfile 定義文

再開始計算時の過去タリーファイル名を指定します。書式は

```
resfile = file.001
```

のようにパス名を含めてファイル名を書きます。file と違うのは、複数タリーを指定しても、resfile は 1 つで十分です。2 つ以上指定した場合、先に書いたファイルが優先されます。resfile のデフォルトは、file です。その場合、過去タリーファイルから結果を読み込んで、新しい結果を加えて同じファイルに上書きします。

5.7.5 unit 定義文

出力の単位を定義します。通常番号で次の様に定義します。

unit = *number*

番号とその単位の内容は、各タリーの説明で解説します。

5.7.6 factor 定義文

出力の規格化定数を定義します。[t-gshow] タリーでは、境界線の太さを指定します。次の様に定義します。

factor = *number*

出力の物理量にこの定数が掛けられます。

5.7.7 output 定義文

出力する情報の種類を定義します。次の様に定義します。

```
output = name of output
```

詳細は、各タリーの説明で行います。

5.7.8 info 定義文

出力するタリー出力で、詳細情報を出力するかどうかのオプションです。通常、0 か 1 で定義します。

```
info = 0, 1
```

5.7.9 title 定義文

タリー出力に表示されるタイトルを定義します。

```
title = title of the tally
```

省略可能です。省略された場合、デフォルト値が入ります。

5.7.10 ANGEL パラメータ定義文

タリー出力で、*ANGEL* のパラメータを追加します。

```
angel = xmin(1.0) ymin(1.3e-8)
```

ここで定義したパラメーターは、タリー出力の中で

```
p: xmin(1.0) ymin(1.3e-8)
```

と記載されます。*ANGEL* の使い方は、*ANGEL* のマニュアルをご覧下さい。

5.7.11 2d-type 定義文

タリー出力で、axis = xy などの 2 次元表示を選択したときの、2 次元データの表示のオプションです。番号で指定します。ただし、rshow オプションがあるタリーでは、意味を持ちません。

```
2d-type =  1, 2, 3, 4, 5, 6, 7
```

- 2d-type = 1, 2, 3, 6, 7
  これは、データの並びが Fortran の書式で書くと

  ```
  ( ( data(ix,iy), ix = 1, nx ), iy = ny, 1, -1 )
  ```

  で、1 行に 10 個のデータが入り、*AnGeL* 用のヘッダーが付きます。ヘッダーは、1 は、等高線、2 は、クラスタープロット、3 は、カラープロットです。6 は、クラスターと等高線です。7 は、カラープロットと等高線です。

- 2d-type = 4
  これは、データの並びが Fortran の書式で書くと

  ```
  do iy = ny, 1, -1
  do ix = 1, nx
     ( x(ix), y(iy), data(ix,iy) )
  end do
  end do
  ```

  で、1 行に x(ix), y(iy), data(ix,iy) の 3 個のデータが入ります。

- 2d-type = 5
  これは、データの並びが Fortran の書式で書くと

  ```
  y/x  ( x(ix), ix = 1, nx )
    do iy = ny, 1, -1
       ( y(iy), data(ix,iy), ix = 1, nx )
    end do
  ```

  で、1 行に nx + 1 個のデータが並び、全部で ny + 1 行です。Excel 等に取り込むデータとして便利です。

5.7.12 gshow 定義文

[t-gshow], [t-rshow] 以外のタリーで用いることができます。タリーのメッシュが xyz メッシュ、axis が xy, yz, xz で、かつ 2d-type = 1,2,3,6,7 の *AnGeL* の出力を 想定したタリーで、これを指定すると、出力の画面に領域の境界線、また物質番号、領域番号、LAT 番号が表示されます。

```
gshow =  0, 1, 2, 3, 4
```

0 で表示無し、1 は境界線の表示、2 は境界線と共に物質番号、3 は境界線と共に領域番号、4 は境界線と共に領域番号と LAT 番号、を表示します。物質番号、領域番号、LAT 番号を表示させるときに、resol を使って分解能を resol 倍にすると、番号の表示が乱れますので、番号を表示させるときは、resol でなくメッシュの個数を増やして分解能を上げてください。パラメータセクションで、icntl = 8 とすることにより、 実際の計算をすることなしに、タリーを設定した領域を事前に見ることができます。icntl = 8 とすると、輸送計算はせずに、xyz メッシュ、xy, yz, zx,　axis のタリーで gshow = 1, 2, 3, 4 が指定してあるものの領域を指定のファイルに書き出します。その時、物質毎に色分けします。この機能を使い、大きい計算の前に、タリーの領域の確認、xyz メッシュの分解能の適否を確認することをお勧めします。

5.7.13　rshow 定義文

`[t-cross]` と `[t-gshow]` 以外のタリーで用いることができます。タリーの `mesh` が reg メッシュの場合に利用でき、領域毎に求めた物理量の大きさに応じた色を幾何形状表示させた体系に付与します。`axis` は `xy, yz, xz` のどれかを指定する必要があり、更に rshow 定義文の下に `xyz` のメッシュ定義文が必要となります。`rshow` の値を変えることで、境界線、物質番号、領域番号を表示させることができます。

```
rshow =  1, 2, 3
x-type = [2,4]
   ..........
   ..........
y-type = [2,4]
   ..........
   ..........
z-type = [2,4]
   ..........
   ..........
```

0 で表示無し、1 は境界線の表示、2 は境界線と共に物質番号、3 は境界線と共に領域番号、を表示します。0 の時は、xyz メッシュ定義文は、不必要なのでコメントアウトして下さい。 物質番号、領域番号を表示させるときに、resol を使って分解能を resol 倍にすると、番号の表示が乱れますので、番号を表示させるときは、resol でなくメッシュの個数を増やして分解能を上げてください。

reg メッシュでこのオプションを付けると、領域ごとの出力はありません。従って、計算の後に図の体裁や最大値最小値の調整などは、元のデータがありませんからできません。このオプションを付ける時は、axis = xy,yz,zx で用いるわけですが、その他に axis = reg も加えて　別ファイルに領域の物理量のデータを保存することをお勧めします。そのデータと、[t-rshow] タリーを用いることにより、再度加工したデータを元に表示することが可能となります。

パラメータセクションで、icntl = 10 とすることにより、実際の計算をすることなしに、タリーを設定した領域を事前に見ることができます。icntl = 10 とすると、輸送計算はせずに、reg メッシュ、xy, yz, zx, axis のタリーで rshow = 1, 2, 3 が指定してあるものの領域を 指定のファイルに書き出します。その時、物質毎に色分けします。この機能を使い、大きい計算の前に、タリーの領域の確認、xyz メッシュの分解能の適否を確認することをお勧めします。

5.7.14 x-txt, y-txt, z-txt 定義文

A_NG_EL 表示の x, y, z 軸のテキストをデフォルトから変えたい時に用います。これらのテキストは、A_NG_EL パラメータ定義文では、変えられません。

```
x-txt = テキスト
y-txt = テキスト
z-txt = テキスト
```

5.7.15 volmat 定義文

volmat パラメータは、xyz メッシュでメッシュが領域境界を またいでいる時の体積補正をするものです。これが有効になるのは、xyz メッシュでかつ material の指定がある場合です。領域をまたいだメッシュの体積を、メッシュサイズから与えるのではなく、指定された material を含む体積をモンテカルロ的に計算します。その時のスキャンは、各軸に平行な 1 辺当たり volmat 数の軌跡で計算します。この数をあまり大きく取るとメッシュ数にもよりますが、計算時間が膨大になることがありますので注意して下さい。volmat を負の数で指定すると、強制的に全ての xyz メッシュをスキャンします。正の場合は、メッシュの 8 頂点が同じ物質ならスキャンしません。

5.7.16 epsout 定義文

`epsout = 1` を指定すると、出力ファイルを A_NG_EL で処理した eps ファイルを作成します。ファイル名は出力ファイルの拡張子を eps に変えたファイル名です。パラメータセクションで `itall = 1` を指定して、出力の eps ファイルを ghostview を 用いて画面に表示し、`w` キーをを押して eps ファイルの更新に伴う画面の書き換えを指定していれば、バッチ毎にタリーの結果の変化を自動的に画面上で確認できます。

5.7.17 カウンター定義文

`[counter]` セクションで定義したカウンターを用いて、タリーで集める物理量に制限を加えることができます。各カウンターごとに、最小値と最大値を `ctmin(i), ctmax(i)` で定義します。`i` は、カウンター番号、1 から 3 です。デフォルト値は最小値が-9999、最大値が 9999 です。複数のカウンターを用いる時は、それらの条件の共通部分となります。

5.7.18 resol 分解能、width 線太さ定義文

`resol` を使い、gshow、rshow、3dshow の表示の時、指定した xyz のメッシュを固定したまま、境界線を求める分解能を上げることができます。デフォルトは 1 で、xyz のメッシュの分解能と同じです。`resol = 2` とすると各辺 2 倍のメッシュになります。xyz メッシュのタリーで、境界線の精度だけ上げるのに便利です。また、3dshow の時も、荒い精度で表示を確認してから、`resol` を大きくして最終的なきれいな図を得ることができます。`resol` を大きくしてもメモリーは変化しません。ただし、物質番号、領域番号、LAT 番号を同時に表示させるときには、resol を使って分解能を resol 倍にすると、番号の表示が乱れますので、番号を表示させるときは、resol でなくメッシュの個数を増やして分解能を上げてください。

`width` は、gshow、rshow、3dshow の表示の時の線の太さを定義します。デフォルトは、0.5 です。

5.7.19 trcl 座標変換

r-z、xyz メッシュの座標を trcl により座標変換します。書式は以下のように 2 通りあります。

```
trcl = number
trcl = O1  O2  O3  B1  B2  B3  B4  B5  B6  B7  B8  B9  M
```

最初の書式は、[transform] セクションで定義した座標変換番号です。次の書式は、この定義の中で座標変換を定義します。[transform] と同じように 13 個の数字で定義します。一行に収まらない場合は、複数の行にわたって定義できます。その時は自動認識しますので、行末の継続行の記号 ¥は必要ありません。ただし、継続行は先頭に 12 個以上の空白が必要です。

r-z、xyz メッシュの他に、3dshow のタリーで box を定義するときにも、trcl による座標変換が使えます。書式は [t-3dshow] のところで示します。

5.7.20 dump 定義文

[t-cross], [t-time], [t-product] タリーで、粒子の情報をファイルにダンプすることができます。その時、ダンプする物理量とそのデータ並びを dump 定義文で指定できます。

dump データの情報は、まず、dump = でデータの個数を指定します。正で与えた時は、バイナリーファイル、負で与えた時は、アスキーファイルへの書き込みをします。次の行にひとつのレコードのデータの並びを指定します。その時の番号と物理量の関係は、以下の通りです。

表 77: dump データの種類と番号 (1)

物理量	kf	x	y	z	u	v	w	e	wt	time	c1	c2	c3	sx	sy	sz
番号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

表 78: dump データの種類と番号 (2)

物理量	name	nocas	nobch	no
番号	17	18	19	20

ここで、kf は粒子を識別する kf-code (表 4 を参照してください)、x, y, z は座標 (cm)、u, v, w は運動量の単位ベクトル、e はエネルギー (MeV)、原子核の場合は核子当たりのエネルギー、wt は粒子ウエイト、time は時間 (nsec)、c1, c2, c3 はカウンターの値、sx, sy, sz はスピンの方向ベクトルです。name は粒子の衝突回数、nocas はバッチの中の現在のイベント数、nobch は現在のバッチ数、no は現在のイベントでのカスケード ID です。これらはバイナリーファイルの時は real*8 のデータで、アスキーファイルの時は、*n*(1p1d24.15) のフォーマットで格納します。

例えば、9 つのデータを次の順番で書き出させたいとき、

```
kf  e  wt  x  y  z  u  v  w
```

このデータを書き込みたいときは、

```
dump = 9
  1  8  9  2  3  4  5  6  7
```

と指定します。

dump 定義文を用いてダンプさせる時は、axis、file 数ともにひとつに制限されます。また、unit は 1 に固定されます。dump するデータを書き込むファイルは、file 定義文で指定した名前に “_dmp” を付けたファイルです。file 定義文で定義された file 名には、通常のタリーの結果が書き出されます。この結果からダンプされたデータの規格化定数を読み取ってください。（バージョン 2.66 以前では、file 定義文で指定したファイルに dump するデータが書き込まれ、通常のタリー結果を出力するファイルとして `.cfg` をつけたファイルが作成されていました。）メモリ分散型並列計算時には使用する並列 PE (Processor Element) 数 −1 個のファイルを作成します。データを dump するファイルの名前の最後に PE 番号をつけたファイルが作られ、各 PE は対応するファイルにのみデータの書き出しと読み込みを行います。

5.8 複数のタリー結果の統合機能

Sumtally 機能を使うことにより、複数のタリー結果を足し合わせることができます。足し合わせの方法は 2 種類あり、手動で並列計算を実行するために各タリー結果のヒストリー数を考慮して足し上げる方法と、各タリー結果に任意の重み付けをして加重平均を求める方法があります。前者は、並列計算が利用できず、限られた計算資源を有効に利用したい場合に役立ちます。後者は、例えば複数の線源による計算を行う際、それらの強度を任意に変えた場合の結果を、計算をやり直さずに求めることができます。

Sumtally 機能が動作する条件は次の 3 つです。

- [parameters] セクションにおいて `icntl=13` とする
- 足しあわせたいタリー結果の条件（`mesh, axis, part` など）が一致していること
- 足しあわせたいタリー結果の条件が書かれたタリーセクションにおいて、sumtally サブセクションを設定する[30] [31]

Sumtally サブセクションが設定されていても、`icntl=13` となっていなければその内容は無視されます。（バージョン 2.88 以降では、全てのタリーで本機能を利用できます。）この機能の具体的な利用方法については、`\phits\utility\sumtally` フォルダにある資料やサンプルインプットをご参照ください。

Sumtally サブセクションは、“sumtally start” と “sumtally end” で挟んだ領域で設定します。

Sumtally サブセクションで使用できるパラメータは表 79 の通りです。

表 79: sumtally サブセクションパラメータ

name	値	説明
isumtally =	1(省略時), 2	Sumtally 機能の計算方法 1: 手動並列計算用 2: 加重平均計算
nfile = (次行)	数 file name 数値	タリーファイル数 タリーファイル名、重み付けの値 isumtally=2 の場合は自動的に規格化されます
sfile	file name	sumtally 機能で足しあわせた結果の出力ファイル名
sumfactor	（省略可、D=1.0）	規格化定数

`isumtally=1` の場合は手動で並列計算を実行することができます。例えば、`maxcas=100, maxbch=10` で計算した結果 result-1.out と `maxcas=100, maxbch=20` で計算した結果 result-2.out がある場合、次のように sumtally サブセクションを設定することで `maxcas=100, maxbch=30` の条件で計算した結果 result-s.out を得ることができます。

List 5.3 ● sumtally 機能を用いた例題（`isumtally=1`）の場合）

```
1:  sumtally start
2:  isumtally = 1      $(D=1) sumtally option, 1:integration, 2:weighted sum
3:  nfile = 2          $ number of tally files
4:     result-1.out  1.0
5:     result-2.out  1.0
```

[30] バージョン 2.81 以前では、sumtally サブセクションが設定されたタリーセクションにおいて `file` パラメータが設定されていなくても動作しましたが、バージョン 2.82 以降ではこのパラメータがないとエラーになりますのでご注意ください。

[31] バージョン 2.85 以前では、1 つのインプットファイルにおいて 1 つの sumtally サブセクションしか設定できませんのでご注意ください。

```
6:  sfile=result-s.out $ file name of output by sumtally option
7:  sumfactor = 1.0    $ (D=1.0) normalization factor
8:  sumtally end
```

ただし、乱数の重複を避けるために、result-2.out の計算を実行する際は [parameters] セクションにおいて `irskip=-1000` を設定してください。また、足し合わせるファイル名の右で設定する重み付けの値は、基本的に 1 としてください。`isumtally=1` の場合、重み付けの値を 1 より変更することで、各タリー結果を求めた際のソースウエイトの平均値を変えることになります（[source] セクションで `wgt` を変更した場合と同じ）。なお、`isumtally=1` で得られたタリー結果に対しては再開始計算を実行することが可能です。

`isumtally=2` の場合、次の計算式により加重平均値 $\bar{X}$ を求めます。

$$\bar{X} = F \sum_{j=1}^{N} \frac{r_j}{r} \bar{X}_j \tag{2}$$

ここで、F が `sumfactor` で指定する規格化定数、N が `nfile` で指定するタリーファイル数、$\bar{X}_j$ が j 番目のタリーファイルの結果、r_j が j 番目のタリーファイルの重み付けの値です。r は $r = \sum_{j=1}^{N} r_j$ で計算しており、重み付けの値 r_j を変えることで、各タリー結果の相対的な強度を任意に与えることができます。標準偏差 σ_X は、j 番目のタリー結果の標準偏差を σ_{X_j} とし、

$$\sigma_X = F \sqrt{\sum_{j=1}^{N} \left(\frac{r_j}{r}\right)^2 \sigma_{X_j}^2} \tag{3}$$

により求めています。`isumtally=2` は、例えば、ある標的に対して左右から違う量の放射線を照射した結果を求めたい時に利用できます。左から照射した結果が result-l.out、右から照射した結果が result-r.out であった場合、2:3 の割合で照射した結果は次のような設定で計算することができます。

List 5.4 ● sumtally 機能を用いた例題（`isumtally=2`）の場合）

```
1:  sumtally start
2:  isumtally = 2      $(D=1) sumtally option, 1:integration, 2:weighted sum
3:  nfile = 2          $ number of tally files
4:     result-l.out  2.0
5:     result-r.out  3.0
6:  sfile=result-s.out $ file name of output by sumtally option
7:  sumfactor = 5.0    $ (D=1.0) normalization factor
8:  sumtally end
```

この条件と同じシミュレーションはマルチソースの機能を利用することで実行可能です。しかし、照射割合を変えるなど、色々な条件の計算を調べたい場合は、`isumtally=2` を利用するのが便利です。なお、重み付けの値は、足し合わせるタリー結果の相対的な割合にしか影響しませんのでご注意ください。絶対値を変える場合は `sumfactor` の値を変更します。また、`isumtally=2` の場合は、乱数等の再開始計算に必要な情報は出力されず、再開始計算ができないようになっています。

6 タリー入力書式

6.1 [T-Track] セクション

指定した任意の空間における粒子の fluence を出力します。このタリーでは、指定した空間を粒子が通過した際に、その空間における飛跡長 (track length) を計算しており (図 31 参照)、飛跡長の和を空間の体積で割ることによって、単位面積あたりの粒子の流量が得られるようになっています。

例えば、このタリーは指定した空間に置いた測定機器の応答の状況を調べるのに利用できます。各測定機器がもつ応答性能を ([cm^2] の単位をもつ) 断面積の形で計算しておけば、その値を本タリーで求めた fluence と掛け合わせることにより、シミュレートした状況においてその測定機器が何回応答するかを評価できます。

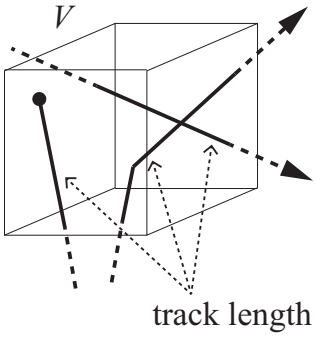


図 31: Track タリー: 空間内の飛跡 (実線) の長さを計算する。

表 80: [t-track] パラメータ (1)

name	値	説明
mesh =	reg, r-z, xyz	形状メッシュ、形状メッシュサブセクションが必要
part =	all(省略時), 粒子名	ひとつの [t-track] セクション最大 6 個まで
material =	(省略可) all, 数 (次行)	スコアする material を限定する。複数定義可 all : デフォルト、この場合省略した場合と同じ 数を指定した場合、その数だけの material を次の行に記述する。 負の数にした場合は、それらの material を対象から外すことを意味する。 material 番号
e-type =	1, 2, 3, 4, 5	エネルギーメッシュ エネルギーメッシュサブセクションが必要
t-type =	1, 2, 3, 4, 5 (省略可)	時間メッシュ 時間メッシュサブセクションが必要
a-type =	1, 2, -1, -2 (省略可)	mesh=r-z とした場合の θ に関する角度メッシュ 角度メッシュサブセクションが必要

表 81: [t-track] パラメータ (2)

name	値	説明
unit =	1, 2, 3, 4 11, 12, 13, 14	1: [1/cm^2/source] 2: [1/cm^2/MeV/source] 3: [1/cm^2/Lethargy/source] 4: [cm/source] 11:[1/cm^2/nsec/source] 12:[1/cm^2/nsec/MeV/source] 13:[1/cm^2/Lethargy/nsec/source] 14:[cm/nsec/source]
axis =	eng, reg, x, y, z, r, rad, deg, xy, yz, xz, rz, t	出力データの x 軸 出力データの x 軸（mesh=r-z で、θ に関する自由度） 2 次元表示 時間軸
file =	file name	axis の数だけ定義する
resfile =	(省略可, D=file)	再開始計算時の過去タリーファイル名。複数 axis の場合でも 1 つのみ指定。
multiplier =	物質数 (省略可)	multiplier を物質毎に指定します multiplier サブセクションの書式は下に示します
factor =	(省略可、D=1.0)	normalization factor
title =	(省略可)	出力ファイルヘッダーのタイトル
angel =	(省略可)	angel パラメータ
2d-type =	1,2,3,4,5,6,7 (省略可)	2 次元表示のオプション
x-txt = y-txt = z-txt =	(省略可) (省略可) (省略可)	x-軸テキスト y-軸テキスト z-軸テキスト
gshow =	0(省略時), 1, 2, 3, 4	mesh=xyz, axis=xy,yz,xz の時、領域境界 (1)、物質番号 (2)、領域番号 (3)、LAT 番号 (4) を表示
rshow =	0(省略時), 1, 2, 3	mesh=reg, axis=xy,yz,xz の時、領域境界 (1)、物質番号 (2)、領域番号 (3) を表示 この下に xyz 形状メッシュセクションが必要

unit = 3, 13 の場合の Lethargy はエネルギーに関する自然対数目盛を表しています。各エネルギービンの上限と下限がそれぞれ $E_{\rm high}, E_{\rm low}$ のときに、各 Lethargy の幅を $\ln(E_{\rm high}/E_{\rm low})$ で与えます。

unit = 1, 2, 3, 11, 12, 13 を指定すると、fluence、すなわち入射粒子あたりの飛跡長の和をその空間の体積で割った値を出力します。ただし、reg メッシュを用いている場合は、その体積を [Volume] セクションで与える必要があります。これが与えられていないときは、体積が 1cm^3 であるとして、入射粒子あたりの飛跡長の和をそのまま出力します。r-z, xyz メッシュの場合は体積が自動的に計算できますので、その値が用いられます。unit=4,14 を指定すると、体積で割らずにそのまま飛跡長の和を出力します。

本タリーに関しては、mesh=r-z とした場合に、既存の r, z メッシュに加え、θ メッシュを指定することができます。θ に関するメッシュを指定する際は、a-type を用いてください。また、角度を横軸にして出力する場合は、axis として rad か deg を指定してください。rad は radian で、deg は degree の単位で出力します。

表 82: [t-track] パラメータ (3)

name	値	説明
ginfo =	0(省略時), 1 2	gshow, rshow オプションの時領域エラーチェックを行わない。 領域エラーチェックを行い 2 次元ジオメトリ図上にエラー領域を表示する。 エラー領域表示に加えて領域エラーファイル (.err) を出力。
resol = width =	1(省略時) 0.5(省略時)	gshow, rshow オプションの時、領域境界を求める分解能を各辺 resol 倍します。 gshow, rshow オプションの時、領域境界を表示する線の太さを定義します。
volume reg vol	(省略可)	reg メッシュの時、 各領域の体積を定義します。 省略した場合、インプットエコーにデフォルト値が表示されます。この行の下に volume 定義文が必要 volume 定義文。書式は 5.1.2 を参照。
iechrl =	72(省略時)	volume のインプットエコーの時の最大コラム数
volmat =	(省略可、D=9)	xyz メッシュで material が指定されている時、 各メッシュの volume correction を行う。(0 で無し) volmat の値は、xyz メッシュ 1 辺のスキャン数
epsout =	0(省略時), 1, 2	1 で出力ファイルを ANGEL で処理した eps ファイルを作成。ファイル名は出力ファイルの拡張子を eps に変えたファイル名。 2 とすることで更に誤差棒を表示する。ただし、 2 次元表示 (axis= xy, yz, xz, rz) の場合は無効。
bmpout =	0(省略時), 1	二次元ビットマップ画像ファイルを出力する。 ファイル名は出力ファイルの拡張子を bmp に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkout =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルを出力する。 ファイル名は出力ファイルの拡張子を vtk に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkfmt =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルのフォーマット。 テキスト形式（0）またはバイナリ形式（1）を選択する。
ctmin(i) = ctmax(i) =	(省略可、D=-9999) (省略可、D= 9999)	i-th カウンターの最小値 i-th カウンターの最大値
trcl =	(省略可)	r-z, xyz メッシュの座標変換番号もしくは座標変換定義
gslat =	1(省略時) 0	1 で gshow で lattice の境界を表示 0 で gshow で lattice の境界を非表示

multiplier サブセクションを用いて、本タリーの結果にエネルギーに依存した係数を乗じることができます。任意の関数を用意する場合は [multiplier] セクションを使用します。係数のセットを表す ID 番号を k として (ただし $k < 0$)、基本的に ($C \quad k$) の様な表式を用いて指定します。ここで、C は規格化定数です。

multiplier サブセクションの書式は以下の通りです。

```
multiplier =  number of material
      part = neutron
      emax = 1000
    mat     mset1          mset2
      1   ( 1    -201 )  ( 2 -202 )
      2   ( 1.2  -201 )  ( 3 -202 )
    ....   ....     ....   ....
    ....   ....     ....   ....
```

multiplier =で指定する物質の数を決定します。all も可能ですが、この時は、下の mat の項でも all を用います。part =で係数を乗じる粒子を指定します。6 個まで複数指定可能で、all も利用できます。省略すれば all を指定したことになります。ただし、指定した粒子以外の寄与はゼロになります。emax =で係数を乗じるエネルギーの上限を設定します。省略した場合は、[multiplier] セクションで指定したエネルギーの最大値、もしくはライブラリを使う場合はその上限値がセットされます。それ以上では、その最大値に対する値が入ります。mat が係数を乗じる物質番号です。mset1, mset2 は multiplier セットの指定です。このセットは全部で 6 つまで可能で、それぞれのセット毎に結果が出力されます。また、ひとつの [t-track] セクション内に複数の multiplier サブセクションを指定できます。ただし、その時の multiplier セットの数は等しくなければなりません。

[multiplier] セクションで定義せずに使用できる係数もあります。$k = -1$ の場合は 1/weight を、$k = -2$ の場合は 1/velocity を乗じます。$k = -120$ の場合は、物質の密度を乗じますので、icntl=5 と組み合わせれば領域内の質量が計算できます。他に、*PHITS* に内蔵されている線量換算係数のセットが利用できます。$k = -101, -102, -112, -114$ でそれぞれ陽子、中性子、電子、光子のセットが選択されます。これらの換算係数は、AP(前方-後方) 照射条件で評価された実効線量換算係数です。[32] 線量換算係数の単位は (μSv/h)/(n/sec/cm^2) です。(*PHITS* ver.2.00 から、線量換算係数のデータの内挿の仕方を linear-linear から log-log に変えましたので、注意してください。)

[32] Y. Sakamoto, O. Sato, S. Tsuda, N. Yoshizawa, S. Iwai, S. Tanaka, and Y. Yamaguchi, ”Dose conversion coefficients for high-energy photons, electrons, neutrons and protons”, JAERI-1345, (2003)

また、以下に示す MCNP の FM カードと同様の書式も利用できます。

```
multiplier =  number of material
      part = proton
      emax = 150
    mat             mset1    mset2
      1   ( 0.1236 1 1 -4 )  ( 0.0 )
      2   ( 0.0060 2 1 -4 )  ( 0.0 )
      3   ( 0.0032 3 1 -4 )  ( 0.0 )
    ....   ....     ....   ....
    ....   ....     ....   ....
multiplier =  number of material
      part = neutron
      emax = 150
    mat                     mset1                mset2
      1   ( 0.1236 1 1 -4 : -6 -8 )  ( 1.0 -1 33 0.543 )
      2   ( 0.0060 2 1 -4 : -6 -8 )  ( 1.0 -1 34 0.321 )
      3   ( 0.0032 3 1 -4 : -6 -8 )  ( 1.0 -1 35 0.678 )
    ....   ....     ....   ....
    ....   ....     ....   ....
```

この例題の `mset1` は、heat をとるセットです。また、`mset2` は、陽子がゼロで、中性子は attenuator セットとなっています。

6.2 [T-Cross] セクション

指定した任意の面における粒子の current、または flux を出力します。粒子が面を通過する度にそのままカウントするものが current で、面の法線ベクトルに対して粒子が角度 θ で入射した場合に $1/\cos\theta$ の重みを付けてカウントするものが flux（ただし、単位時間あたりの量ではないので、正確には fluence）です。これらは共に面を通過する単位面積あたりの粒子の数を意味する物理量ですが、current が単に指定した面を通過する粒子の総量を評価するのに対し、flux は粒子の流れの方向に垂直な面を通過する量を評価します。したがって、単位面積あたりの値を求める際に必要となる面積が違うこととなり (図 32 参照)、flux の場合は $1/\cos\theta$ の重みを付けてカウントすることになります。なお、図 32 の S は、`mesh=reg` を選択している場合は形状メッシュサブセクションにおいて `area` として与えます。`mesh=r-z, xyz` を選択している場合は自動的に計算された値が使用されるため必要ありません。

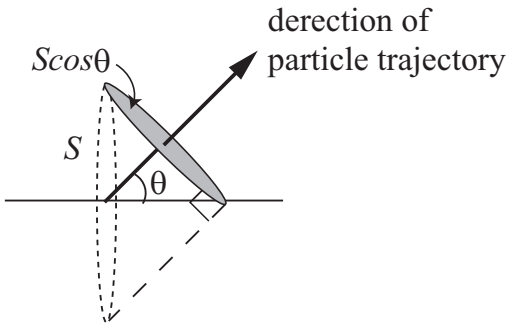


図 32: 指定した面の面積 S と粒子の運動の方向に対して有効な面素 $S\cos\theta$ の関係。

本タリーの flux は $1/\cos\theta$ の重みを考慮した量となるため、`[t-track]` で指定する領域の厚さを無限に薄くした場合と同等の結果が得られます。これにより例えば、指定した面に沿って設置した測定機器の応答の状況を調べるのに利用でき、前もって ([cm^2] の単位をもつ) 断面積の形で計算しておいた各測定機器の応答性能と掛け合わせることにより、その測定機器が応答する回数を見積もることができます。

表 83: [t-cross] パラメータ (1)

name	値	説明
mesh =	reg, r-z, xyz	形状メッシュ、形状メッシュサブセクションが必要
part =	all(省略時), 粒子名	ひとつの [t-cross] セクション最大 6 個まで
e-type =	1, 2, 3, 4, 5	エネルギーメッシュ エネルギーメッシュサブセクションが必要
a-type =	1, 2, -1, -2 a-curr, oa-curr の時に必要	角度メッシュ (1, 2 :cos, -1, -2 :degree) 角度メッシュサブセクションが必要
t-type =	1, 2, 3, 4, 5 (省略可)	時間メッシュ 時間メッシュサブセクションが必要
unit =	1, 2, 3, 4, 5, 6 11, 12, 13, 14, 15, 16	1: [1/cm^2/source] 2: [1/cm^2/MeV/source] 3: [1/cm^2/Lethargy/source] 4: [1/cm^2/sr/source] 5: [1/cm^2/MeV/sr/source] 6: [1/cm^2/Lethargy/sr/source] 11: [1/cm^2/nsec/source] 12: [1/cm^2/MeV/nsec/source] 13: [1/cm^2/Lethargy/nsec/source] 14: [1/cm^2/sr/nsec/source] 15: [1/cm^2/MeV/sr/nsec/source] 16: [1/cm^2/Lethargy/sr/nsec/source]
axis =	eng, reg, x, y, z, r, cos, the, t xy, yz, xz	出力データの x 軸 角度 (cos, the), 時間 (t) 2 次元表示

r-z メッシュの時にメッシュの側面が円柱状のあるセルの側面と一致する場合や、xyz メッシュの時に座標変換によって存在するセルの境界面とタリー面が一致する場合は、計算結果がおかしくなる可能性があるので注意してください。

角度メッシュを用いることで、面の法線ベクトルとのなす角 θ 毎の current もカウントできます。図 33 はこの測定をイメージしたものです。unit=4, 5, 6, 14, 15, 16 を指定すると、角度メッシュサブセクションで与えられる角度のビンの幅から対応する立体角を計算し、単位立体角あたりの量を出力します。

unit = 3, 6, 13, 16 の場合の Lethargy はエネルギーに関する自然対数目盛を表しています。各エネルギービンの上限と下限がそれぞれ $E_{\mathrm{high}}, E_{\mathrm{low}}$ のときに、各 Lethargy の幅を $\ln(E_{\mathrm{high}}/E_{\mathrm{low}})$ で与えます。

unit = 4, 5, 6, 14, 15, 16 の場合の sr は、立体角の単位ステラジアン (steradian) を表しています。

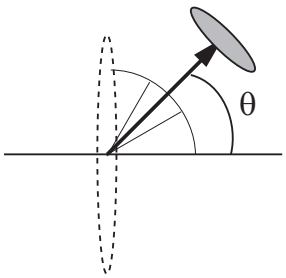


図 33: 角度メッシュを用いた current の角度分布の評価。

表 84: [t-cross] パラメータ (2)

name	値	説明
file =	file name	axis の数だけ定義する
resfile =	(省略可, D=file)	再開始計算時の過去タリーファイル名。複数 axis の 場合でも 1 つのみ指定。
multiplier =	物質数 (省略可)	multiplier を物質毎に指定します multiplier サブセクションの書式は下に示します
factor =	(省略可、D=1.0)	normalization factor
title =	(省略可)	出力ファイルヘッダーのタイトル
angel =	(省略可)	angel パラメータ
2d-type =	1,2,3,4,5,6,7 (省略可)	2 次元表示のオプション
output =	flux current f-curr b-curr o-curr of-curr ob-curr a-curr oa-curr	面横断の flux を出力 面横断の current を出力 面横断の forward current を出力 面横断の backward current を出力 面横断の omni current を出力 omni は、energy integrated を意味します 面横断の omni forward current を出力 面横断の omni backward current を出力 面横断の角度メッシュ current を出力 面横断の角度メッシュ omni current を出力
x-txt = y-txt = z-txt =	(省略可) (省略可) (省略可)	x-軸テキスト y-軸テキスト z-軸テキスト
gshow =	0(省略時) 1, 2, 3, 4	mesh=xyz, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3)、LAT 番号 (4) を表示
ginfo =	0(省略時), 1 2	gshow オプションの時領域エラーチェックを行わない。 領域エラーチェックを行い 2 次元ジオメトリ図上にエラー領域 を表示する。 エラー領域表示に加えて領域エラーファイル (.err) を出力。
resol = width =	1(省略時) 0.5(省略時)	gshow オプションの時、領域境界を求める 分解能を各辺 resol 倍します。 gshow オプションの時、領域境界を表示する 線の太さを定義します。
epsout =	0(省略時), 1, 2	1 で出力ファイルを ANGEL で処理した eps ファイル を作成。ファイル名は出力ファイルの拡張子を eps に 変えたファイル名。 2 とすることで更に誤差棒を表示する。ただし、 2 次元表示 (axis= xy, yz, xz) の場合は無効。

output=f-curr, b-curr, of-curr, ob-curr は、xyz メッシュと r-z メッシュで利用できます。ただし、xyz メッシュの場合は、z 軸の方向に対してのみタリーが機能します。

表 85: [t-cross] パラメータ (3)

name	値	説明
ctmin(i) = ctmax(i) =	(省略可、D=-9999) (省略可、D= 9999)	i-th カウンターの最小値 i-th カウンターの最大値
trcl =	(省略可)	r-z, xyz メッシュの座標変換番号もしくは座標変換定義
dump = (次行)	データ数 データ定義	ファイルに粒子の情報をダンプします。mesh=reg のみ。 正で与えた時はバイナリー、負の時はアスキーファイル。 データの順番を定義します。ただし、idmpmode=1 の使用のためには、nocas および nobch の情報が必要です。
gslat =	1(省略時) 0	1 で gshow で lattice の境界を表示 0 で gshow で lattice の境界を非表示

[t-cross] で、dump を用いることができるのは、reg メッシュだけです。 また、axis は reg に固定されます。dump を指定したときは、e-type, a-type, t-type の指定は、上限と下限の意味しかもちません。 unit は 1 に固定、output は current もしくは、a-curr, oa-curr になります。 dump 定義文を用いてダンプさせる時は、axis、file 数ともにひとつに制限されます。dump するデータを書き込むファイルは、file 定義文で指定した名前に "_dmp" を付けたファイルです。file 定義文で定義された file 名には、通常のタリーの結果が書き出されます。（バージョン 2.66 以前では、file=で指定したファイルに dump するデータが書き込まれ、通常のタリー結果を出力するファイルとして.cfg をつけたファイルが作成されていました。）dump ファイルを使用した接続計算を idmpmode=1 で実行するには、nocas および nobch の情報を出力しておく必要があります。また、idmpmode=1 を使用するためには、ここで出力される "_dmp" が付いた dump ファイルと共に file= で指定した名前の通常のタリー出力ファイルが必要になります。メモリ共有型並列計算を実行する場合、本機能による dump ファイルの作成と dumpall によるイベント情報の書き出しは共存できませんのでご注意ください。

形状メッシュで、mesh = reg を選択した場合、 それに引き続く形状メッシュサブセクションは、領域横断面を定義するため、領域番号を用いて出て行く領域 (r-in)、入ってくる領域 (r-out)、またその面積 (cm^2) を次の書式で定義する必要があります。出て行く領域（r-in) と入ってくる領域 (r-out）の in と out が逆になっている点にご注意下さい。

```
mesh = reg
 reg =  number of crossing surfaces
  r-in     r-out     area
    2        8       10.0
    3        8        5.0
  ( 4 5 ) ( 4 5 )    2.0
  (13<5)  (14<5)     7.0
  (13<6)  (14<6)     7.0
  (13<7)  (14<7)     7.0
   ...     ...      ....
   ...     ...      ....
```

このとき、データの並びは r-in r-out area がデフォルトです。この定義文は省略できますが、順番を変えたいときは area r-in r-out のように定義文を明示します。読み飛ばしコラム用の non も使えます。領域をまとめる、({ 2 - 5 } 8 9) という書式も使えます。また、(6 < 10[1 0 0] < u=3) などの

lattice, universe 構造も指定できます。ただし、単一の数字で無い場合は必ず ( ) で括ってください。これらの順番を変えたいときは、定義の文字列の順番を変えます。読み飛ばしコラム用の non も使えます。

mesh = reg の時は、上の様に入る領域と出る領域を指定しますので、flux を得る時は注意が必要です。上の例題において、上から 3 番目に示した指定方法以外は、片方向の flux になります。従って、flux を取る時は、この指定方法のように、隣り合う領域の両方を r-in と r-out に書いてください。

形状メッシュで、mesh = r-z を選択した場合は、$r_i - r_{i+1}$ で定義される領域の nz+1 個の z 横断面と、$z_i - z_{i+1}$ で定義される領域の nr+1 個の r 横断面の 2 種類の横断面が定義されます。r-surface と外部ボイドの面が同じ場合は、この面での cross は、タリーされませんからご注意ください。

形状メッシュで、mesh = xyz を選択した場合は、 $x_i - x_{i+1}, y_j - y_{j+1}$ で定義される領域の nz+1 個の z 横断面が定義されます。このとき、x, y 横断面は定義されません。

mesh = r-z, xyz の時は、面を横断する粒子の両方向が自動的に入ります。 また、これらの時の forward は、z 面では、正方向、r 面では中心から外方向を意味します。

バージョン 2.86 より、[t-cross] でも multiplier の機能が使えるようになりました。

multiplier サブセクションを用いて、本タリーの結果にエネルギーに依存した係数を乗じることができます。任意の関数を用意する場合は [multiplier] セクションを使用します。係数のセットを表す ID 番号を k として (ただし $k < 0$)、基本的に $(C \quad k)$ の様な表式を用いて指定します。ここで、C は規格化定数です。

multiplier サブセクションの書式は以下の通りです。

```
multiplier =  number of material
      part = neutron
      emax = 1000
    mat     mset1          mset2
      1   ( 1    -201 )  ( 2 -202 )
      2   ( 1.2  -201 )  ( 3 -202 )
    ....   ....     ....   ....
    ....   ....     ....   ....
```

multiplier =で指定する物質の数を決定します。all も可能ですが、この時は、下の mat の項でも all を用います。part =で係数を乗じる粒子を指定します。6 個まで複数指定可能で、all も利用できます。省略すれば all を指定したことになります。ただし、指定した粒子以外の寄与はゼロになります。emax =で係数を乗じるエネルギーの上限を設定します。省略した場合は、[multiplier] セクションで指定したエネルギーの最大値、もしくはライブラリを使う場合はその上限値がセットされます。それ以上では、その最大値に対する値が入ります。mat が係数を乗じる物質番号です。mset1, mset2 は multiplier セットの指定です。このセットは全部で 6 つまで可能で、それぞれのセット毎に結果が出力されます。また、ひとつの [t-track] セクション内に複数の multiplier サブセクションを指定できます。ただし、その時の multiplier セットの数は等しくなければなりません。

[multiplier] セクションで定義せずに使用できる係数もあります。$k = -1$ の場合は 1/weight を、$k = -2$ の場合は 1/velocity を乗じます。$k = -120$ の場合は、物質の密度を乗じますので、icntl=5 と組み合わせれば領域内の質量が計算できます。他に、PHITS に内蔵されている線量換算係数のセットが利用できます。$k = -101, -102, -112, -114$ でそれぞれ陽子、中性子、電子、光子のセットが選択されます。これらの換算係数は、AP(前方-後方) 照射条件で評価された実効線量換算係数です。[33] 線量換算係数の単位は $(\mu\mathrm{Sv/h})/(n/\mathrm{sec/cm}^2)$

[33] Y. Sakamoto, O. Sato, S. Tsuda, N. Yoshizawa, S. Iwai, S. Tanaka, and Y. Yamaguchi, ”Dose conversion coefficients for high-energy photons, electrons, neutrons and protons”, JAERI-1345, (2003)

です。(PHITS ver.2.00 から、線量換算係数のデータの内挿の仕方を linear-linear から log-log に変えましたので、注意してください。)

また、以下に示す MCNP の FM カードと同様の書式も利用できます。

```
multiplier =  number of material
      part = proton
      emax = 150
    mat             mset1    mset2
      1   ( 0.1236 1 1 -4 )  ( 0.0 )
      2   ( 0.0060 2 1 -4 )  ( 0.0 )
      3   ( 0.0032 3 1 -4 )  ( 0.0 )
    ....   ....     ....   ....
    ....   ....     ....   ....
multiplier =  number of material
      part = neutron
      emax = 150
    mat                     mset1                mset2
      1   ( 0.1236 1 1 -4 : -6 -8 )  ( 1.0 -1 33 0.543 )
      2   ( 0.0060 2 1 -4 : -6 -8 )  ( 1.0 -1 34 0.321 )
      3   ( 0.0032 3 1 -4 : -6 -8 )  ( 1.0 -1 35 0.678 )
    ....   ....     ....   ....
    ....   ....     ....   ....
```

この例題の `mset1` は、heat をとるセットです。また、`mset2` は、陽子がゼロで、中性子は attenuator セットとなっています。

6.3 [T-Point] セクション

[t-point] タリーは，ある点 (point detector) やリング状の線分 (ring detector) におけるフルエンスを計算することができます。PHITS は，通常，設定した領域に確率的に入ってくる放射線をタリーするため，タリー領域が小さくなると，十分な統計精度を得るためには極めて長い計算時間を要する場合があります。極端な例として，タリー領域を点や線分に設定した場合，どれだけ計算時間を費やしても結果を得ることはできません。[t-point] タリーは，[t-track] や [t-cross] タリーの手法とは異なり，線源位置，もしくは，散乱によって粒子が発生した地点で，タリー位置への方向確率と透過確率を計算して評価することができるため、粒子の存在確率が小さい領域での測定に有効に機能します。 ただし，散乱点での生成粒子の角度分布，エネルギー分布があらかじめ分かっていないと評価できません。また，物質中での荷電粒子のエネルギー損失があると透過確率の評価が難しいので，対象粒子は，核データの存在する中性子と光子に限られます。したがって，[t-point] タリーの利用条件は,

(1) 輸送計算の上限エネルギーは，ライブラリを使う上限エネルギー (dmax) とする

(2) 検出可能な粒子は中性子・光子のみとする

(3) Event Generator mode 及び EGS5 mode を使用しない

となります。本タリーの計算原理や具体的な利用方法については、\phits\utility\tpoint フォルダにある readme やサンプルインプットをご参照ください。

表 86: [t-point] パラメータ (1)

name	値	説明
point =	数	指定する point detector の数。サブセクションが必要
ring =	数	指定する ring detector の数。サブセクションが必要
part =	all(省略時)，粒子名	ひとつの [t-point] セクションに最大 6 個まで
e-type =	1，2，3，4，5	エネルギーメッシュ エネルギーメッシュサブセクションが必要
t-type =	1，2，3，4，5 (省略可)	時間メッシュ 時間メッシュサブセクションが必要
unit =	1，2，3 11，12，13	1: [1/cm^2/source] 2: [1/cm^2/MeV/source] 3: [1/cm^2/Lethargy/source] 11:[1/cm^2/nsec/source] 12:[1/cm^2/nsec/MeV/source] 13:[1/cm^2/Lethargy/nsec/source]

unit = 3，13 の場合の Lethargy はエネルギーに関する自然対数目盛を表しています。各エネルギービンの上限と下限がそれぞれ $E_{\rm high}, E_{\rm low}$ のときに、各 Lethargy の幅を $\ln(E_{\rm high}/E_{\rm low})$ で与えます。

表 87: [t-point] パラメータ (2)

name	値	説明
axis =	eng, t	出力データの x 軸
file =	file name	axis の数だけ定義する
resfile =	(省略可, D=file)	再開始計算時の過去タリーファイル名。複数 axis の場合でも 1 つのみ指定。
multiplier =	物質数 (省略可)	multiplier を物質毎に指定します multiplier サブセクションの書式は下に示します
factor =	(省略可、D=1.0)	normalization factor
title =	(省略可)	出力ファイルヘッダーのタイトル
angel =	(省略可)	angel パラメータ
epsout =	0(省略時), 1	1 で出力ファイルを $A_N G_E L$ で処理した eps ファイルを作成。ファイル名は出力ファイルの拡張子を eps に変えたファイル名
ctmin(i) = ctmax(i) =	(省略可、D=-9999) (省略可、D= 9999)	i-th カウンターの最小値 i-th カウンターの最大値

通常のタリーで指定する mesh パラメータの代わりに、[t-point] タリーでは、point もしくは ring パラメータを指定します。まず、point detector の場合、入力書式は次のようになります。

```
[ T-point ]
point =    1     # number of point detectors
non     x        y       z        r0
  1     10.0     0.0     50.0     1.0
```

point=n で point detector の個数 n を指定します。ひとつの [t-point] タリーセクションで定義できるポイントの上限数は 20 です。これ以上のポイントを定義するときは、複数のセクションを用意してください。次の行には、ポイントを指定する座標 (x,y,z)、ポイント周辺の特異領域の半径 (r0) のデータ順を定義します。読み飛ばしのコラムの指標 (non) も使えます。これらのデータの並びを変えるときは、この行で指定してください。この定義文の下に point=で指定した数だけデータを定義します。座標は、x,y,z 座標 [cm] で指定し、特異領域の半径 (r0) も [cm] で指定してください。ここで、特異領域の半径については、\phits\utility\tpoint フォルダにある readme をご参照ください。

次に、ring detector の場合は、次のような入力書式になります。

```
[ T-point ]
ring =    1     # number of ring detectors
non   axis    ar       rr       r0
  1    z      50.0     10.0     1.0
```

ring=n で ring detector の個数 n を指定します。次の行には、リングの軸を x,y,z の文字で指定する axis、原点からリングの位置までの距離 (ar)、リングの半径 (rr)、ポイント周辺の特異領域の半径 (r0) のデータ順を定義します。読み飛ばしのコラムの指標 (non) も使えます。これらのデータの並びを変えるときは、この行で指定してください。この定義文の下に ring=で指定した数だけデータを定義します。リング軸は (x,y,z) の文字で指定し、リングまでの距離 (ar) とリングの半径 (rr) は [cm] で指定します。特異領域の半径 (r0) も [cm] で指定してください。ここで、特異領域の半径については、\phits\utility\tpoint フォルダにある readme をご参照ください。

multiplier サブセクションを用いて、本タリーの結果にエネルギーに依存した係数を乗じることができます。任意の関数を用意する場合は [multiplier] セクションを使用します。係数のセットを表す ID 番号を k として (ただし $k < 0$)、基本的に $(C \quad k)$ の様な表式を用いて指定します。ここで、C は規格化定数です。

multiplier サブセクションの書式は以下の通りです。

```
multiplier =  number of material
      part = neutron
      emax = 1000
    mat     mset1          mset2
      1   ( 1    -201 )  ( 2 -202 )
      2   ( 1.2  -201 )  ( 3 -202 )
    ....   ....     ....   ....
    ....   ....     ....   ....
```

multiplier =で指定する物質の数を決定します。all も可能ですが、この時は、下の mat の項でも all を用います。part =で係数を乗じる粒子を指定します。6 個まで複数指定可能で、all も利用できます。省略すれば all を指定したことになります。ただし、指定した粒子以外の寄与はゼロになります。emax =で係数を乗じるエネルギーの上限を設定します。省略した場合は、[multiplier] セクションで指定したエネルギーの最大値、もしくはライブラリを使う場合はその上限値がセットされます。それ以上では、その最大値に対する値が入ります。mat が係数を乗じる物質番号です。mset1, mset2 は multiplier セットの指定です。このセットは全部で 6 つまで可能で、それぞれのセット毎に結果が出力されます。また、ひとつの [t-point] セクション内に複数の multiplier サブセクションを指定できます。ただし、その時の multiplier セットの数は等しくなければなりません。

[multiplier] セクションで定義せずに使用できる係数もあります。$k = -1$ の場合は 1/weight を、$k = -2$ の場合は 1/velocity を乗じます。$k = -120$ の場合は、物質の密度を乗じますので、icntl=5 と組み合わせれば領域内の質量が計算できます。他に、*PHITS* に内蔵されている線量換算係数のセットが利用できます。$k = -101, -102, -112, -114$ でそれぞれ陽子、中性子、電子、光子のセットが選択されます。これらの換算係数

は、AP(前方-後方) 照射条件で評価された実効線量換算係数です。[34] 線量換算係数の単位は (μSv/h)/(n/sec/cm^2) です。(*PHITS* ver.2.00 から、線量換算係数のデータの内挿の仕方を linear-linear から log-log に変えましたので、注意してください。)

また、以下に示す MCNP の FM カードと同様の書式も利用できます。

```
multiplier =  number of material
      part = proton
      emax = 150
    mat             mset1    mset2
      1   ( 0.1236 1 1 -4 )  ( 0.0 )
      2   ( 0.0060 2 1 -4 )  ( 0.0 )
      3   ( 0.0032 3 1 -4 )  ( 0.0 )
    ....   ....     ....   ....
    ....   ....     ....   ....
multiplier =  number of material
      part = neutron
      emax = 150
    mat                     mset1                mset2
      1   ( 0.1236 1 1 -4 : -6 -8 )  ( 1.0 -1 33 0.543 )
      2   ( 0.0060 2 1 -4 : -6 -8 )  ( 1.0 -1 34 0.321 )
      3   ( 0.0032 3 1 -4 : -6 -8 )  ( 1.0 -1 35 0.678 )
    ....   ....     ....   ....
    ....   ....     ....   ....
```

この例題の `mset1` は、heat をとるセットです。また、`mset2` は、陽子がゼロで、中性子は attenuator セットとなっています。

[34] Y. Sakamoto, O. Sato, S. Tsuda, N. Yoshizawa, S. Iwai, S. Tanaka, and Y. Yamaguchi, ”Dose conversion coefficients for high-energy photons, electrons, neutrons and protons”, JAERI-1345, (2003)

6.4 [T-Heat] セクション

生成熱を出力します。核データを用いる低エネルギーの中性子、光子による発熱も、Kerma factor を用いて同時にタリーすることができます。e-mode≥1 を用いると、核データを用いた中性子の Kerma factor は用いず、放出荷電粒子や、残留核の運動エネルギーから発熱を計算します。光子の発熱は、通常は核データの Kerma factor から計算しますが、electron=1、かつ、電子の輸送を含んでいる時は、電子のエネルギーロスから計算します。

表 88: [t-heat] パラメータ (1)

name	値	説明
mesh =	reg, r-z, xyz	形状メッシュ、形状メッシュサブセクションが必要
axis =	reg, x, y, z, r, xy, yz, xz, rz	出力データの x 軸 2 次元表示
file =	file name	axis の数だけ定義する
resfile =	(省略可, D=file)	再開始計算時の過去タリーファイル名。複数 axis の 場合でも 1 つのみ指定。
material = (次行)	(省略可) all, 数 2 5 8	スコアする material を限定する。複数定義可。 all : デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの material を次の行に 記述する。負の数にした場合は、それらの material を 対象から外すことを意味する。 material 番号
output =	heat simple	total: 全発熱量 total: 全発熱量 ncut, gcut, pcut: 計算打切エネルギー以下の中性子、光子、 陽子の発熱量。(ただしページ下の注意書きをご覧ください。) leakage: 外部ボイドに出ていった粒子の運動エネルギー。 更に、total: 全発熱量 recoil: 計算打切エネルギー (emin(15-19)) 設定時の残留 核の運動エネルギー。 ionization: 荷電粒子のエネルギーロスによる発熱量。 low neutron: 中性子カーマ係数を用いた際の発熱量。 photon: 光子カーマ係数を用いた際の発熱量。 ただし、electron=1 のときは、カットオフ電子による発熱量。 others: 残留核の残留エネルギー。igamma=1 の場合、励起 状態にある残留核から光子が放出されるのでゼロに近づく。

計算打切エネルギー emin 以下の中性子、光子、陽子の扱いに注意して下さい。計算打切エネルギー以下の中性子、光子、陽子であっても、[parameters] セクションにおいてこれらの情報を出力しない設定 (incut=0, igcut=0, ipcut=0:デフォルト値) の場合、上の ncut, gcut, pcut の成分には含まれず、stopped particle のところにタリーされます。incut>0, igcut>0, ipcut>0 の時のみ、ncut, gcut, pcut にタリーされます。

表 89: [t-heat] パラメータ (2)

name	値	説明
output = (続き)	all	simple の結果に加えて、以下の各粒子の寄与を書き出す。 recoil に対する $d, t, {}^3$He, α, residual nuclei の寄与。 ionization に対する p, π^+, π^-, others の寄与。 (ionization に関しては part で指定した粒子の寄与も出力するが、eps ファイルには出力されない。) stopped particle: 物質中で停止した粒子の運動エネルギーの proton, neutron, π^+, π^-, others の寄与。 (stopped particle に関しては part で指定した粒子の寄与も出力するが、eps ファイルには出力されない。) others: 残留核の励起エネルギー (remaining excitation energy) と fission の成分が出力される。 axis が 2 次元表示のときは、all は simple と同じで、total, recoil, ionization, low neutron, electron, others が出力される。
part =	粒子名 (省略可)	output=all の場合に、ここで指定した粒子の荷電粒子のエネルギーロスによる発熱 (ionization 成分) と物質中で停止した粒子のエネルギー量 (stopped particle 成分) を出力する。 ただし、eps ファイルには出力されない。
unit =	0, 1, 2	0: [Gy/source] 1: [MeV/cm^3/source] 2: [MeV/source]
factor =	(省略可、D=1.0)	normalization factor
title =	(省略可)	出力ファイルヘッダーのタイトル
angel =	(省略可)	angel パラメータ
2d-type =	1,2,3,4,5,6,7 (省略可)	2 次元表示のオプション
x-txt = y-txt = z-txt =	(省略可) (省略可) (省略可)	x-軸テキスト y-軸テキスト z-軸テキスト
gshow =	0(省略時), 1, 2, 3, 4	mesh=xyz, axis=xy,yz,xz の時、領域境界 (1)、物質番号 (2)、領域番号 (3)、LAT 番号 (4) を表示
rshow =	0(省略時), 1, 2, 3	mesh=reg, axis=xy,yz,xz の時、領域境界 (1)、物質番号 (2)、領域番号 (3) を表示。 この下に xyz 形状メッシュセクションが必要
ginfo =	0(省略時), 1 2	gshow, rshow オプションの時領域エラーチェックを行わない。 領域エラーチェックを行い 2 次元ジオメトリ図上にエラー領域を表示する。 エラー領域表示に加えて領域エラーファイル (.err) を出力。
resol = width =	1(省略時) 0.5(省略時)	gshow, rshow オプションの時、領域境界を求める分解能を各辺 resol 倍します。 gshow, rshow オプションの時、領域境界を表示する線の太さを定義します。

heat について全般的に言えることですが、最終的に熱に変換するエネルギーは、荷電粒子の ionization の過程を通してです。しかしながら、輸送コードの中では、cutoff エネルギーがあり、最後の過程まで追わずに輸送を終了します。従って、熱の成分として recoil、others などの成分が出力されます。これらの成分は、輸送のパラメータにより変化します。

表 90: [t-heat] パラメータ (3)

name	値	説明
volume reg vol	(省略可)	reg メッシュの時、 各領域の体積を定義します。 省略した場合、インプットエコーにデフォルト値が 表示されます。この行の下に volume 定義文が必要。 volume 定義文。書式は 5.1.2 を参照。
iechrl =	72(省略時)	volume のインプットエコーの時の最大コラム数
volmat =	(省略可、D=9)	xyz メッシュで material が指定されている時、 各メッシュの volume correction を行う。(0 で無し) volmat の値は、xyz メッシュ 1 辺のスキャン数
epsout =	0(省略時), 1	1 で出力ファイルを $A_N G_E L$ で処理した eps ファイル を作成。ファイル名は出力ファイルの拡張子を eps に 変えたファイル名
bmpout =	0(省略時), 1	二次元ビットマップ画像ファイルを出力する。 ファイル名は出力ファイルの拡張子を bmp に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkout =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルを出力する。 ファイル名は出力ファイルの拡張子を vtk に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkfmt =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルのフォーマット。 テキスト形式（0）またはバイナリ形式（1）を選択する。
electron =	0(省略時), 1	電子の寄与のオプション 0: 光子の KERMA factor を使う 1: 電子の ionization loss から計算する。 この場合、電子の輸送が必要
ctmin(i) = ctmax(i) =	(省略可、D=-9999) (省略可、D= 9999)	i-th カウンターの最小値 i-th カウンターの最大値
trcl =	(省略可)	r-z, xyz メッシュの座標変換番号もしくは座標変換定義
gslat =	1(省略時) 0	1 で gshow で lattice の境界を表示 0 で gshow で lattice の境界を非表示

unit = 0 とした場合、[Gy/source] を単位とする吸収線量が出力されます。mesh = reg の場合は、volume パラメーターを用いるか、[volume] セクションにおいて各領域の体積を設定する必要があります。ただし、吸収線量は不可算量ですので、各 x 軸に関する積分値 (sum over) は出力されません。また、領域内に複数の物質が混在する場合、その領域の吸収線量は全体の平均値とならないことにご注意ください。例えば、領域の 1 と 2 にある質量 M_1 と M_2 の物質にそれぞれ E_1 と E_2 の熱量が吸収された場合、その平均吸収線量は $\frac{E_1+E_2}{M_1+M_2}$ ですが、$P_{HI}T_S$ で計算する値は $\frac{E_1}{M_1}+\frac{E_2}{M_2}$ となります。

6.5 [T-Deposit] セクション

このタリーは、[t-heat] とほぼ同等のタリーで、dose や deposit energy 分布をタリーします。[t-heat] との違いは、まず、ここでの dose は、荷電粒子のエネルギーロスだけをタリーします。このタリーでは、任意の物質の LET(dE/dx) の関数のユーザー定義のファクターを乗じることができます。これは、usrdfn1.f, usrdfn2.f で定義されます。サンプルプログラムとして、usrdfn1.f には粒子の LET に ICRP60 で定義された $Q(L)$ 関係を乗じて線量当量を導出するプログラムが、usrdfn2.f には単にエネルギーロスを返すプログラムが入っています。また、時間メッシュと [Timer] を組み合わせることによって、TOF のカウンターを模擬でき、時間と deposit energy の相関を 2 次元プロットすることができます。

本タリーにおいて `output=deposit` とした場合は、検出器の応答を模擬することができます。ただし、`[weight window]` などの使用により、ウエイトが変わるような計算を行った場合は、適切な結果が得られませんのでご注意ください。Version 2.70 より、`output=deposit` としてイベント毎の付与エネルギー分布を計算する際、検出器のエネルギー分解能を考慮して計算できるようになりました。

表 91: `[t-deposit]` パラメータ (1)

name	値	説明
`mesh =`	`reg, r-z, xyz`	形状メッシュ、形状メッシュサブセクションが必要
`part =`	`all`(省略時), 粒子名	ひとつの `[t-deposit]` セクション最大 6 個まで ただし、`output=deposit,deposit=0` の場合、`all` が必須であるため、最大 5 個までしか指定できない。
`material =` (次行)	(省略可) `all`, 数 `2 5 8`	スコアする `material` を限定する。複数定義可。 `all :` デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの `material` を次の行に記述する。 負の数にした場合は、それらの `material` を対象から外すことを意味する。 `material` 番号
`letmat =`	(省略可)	`LET`(dE/dx) を評価する物質番号、省略時は、実際の物質。物質を指定する場合の密度は、`[Material]` セクションで定義された密度となる。したがって、水に対する `LET` 分布を計算したい場合、`[Material]` セクションで、水の密度が `1g/cm`^{3} となるように定義する必要がある。 また、負の場合は電子・陽電子の dE/dx として `1g/cm`^{3} の水に対する値が参照される。詳しくは、奨励設定の `particletherapy` を参照してください。
`dedxfnc =`	(省略可, `D=0`)	`0`:ファクター無し、`1:usrdfn1` を使用、`2:usrdfn2` を使用 サンプルプログラムとして、`usrdfn1.f` には粒子の `LET` に `ICRP60` で定義された $Q(L)$ 関係を乗じて線量当量を導出するプログラムが、`usrdfn2.f` には単にエネルギーロスを返すプログラムが入っています。
`e-type =`	`1, 2, 3, 4, 5`	エネルギーメッシュ、`output=deposit` に必要。 エネルギーメッシュサブセクションが必要
`t-type =`	`1, 2, 3, 4, 5` (省略可)	時間メッシュ、時間メッシュサブセクションが必要

表 92: [t-deposit] パラメータ (2)

name	値	説明
output =	dose deposit	荷電粒子のエネルギーロスをタリー イベント毎の deposit energy 分布をタリー e-type セクションが必要
unit =	0, 1, 2, 3, 4	0: Dose [Gy/source]; output=dose のみ 1: Dose [MeV/cm^3/source] 2: Dose [MeV/source] 3: Number [1/source]; output=deposit のみ 4: Number [1/nsec/source]; output=deposit のみ
deposit =	0(省略時), 1	output=deposit 選択時のエネルギー表示オプション 0: total deposit および各粒子の寄与の分布（通常はこちらをご使用ください） 1: 各粒子の deposit エネルギー分布（ver. 2.81 以前の計算方法です。）
axis =	eng, reg, x, y, z,r, t xy, yz, xz, rz, t-eng, eng-t	出力データの x 軸 2 次元表示
file =	file name	axis の数だけ定義する
resfile =	(省略可, D=file)	再開始計算時の過去タリーファイル名。複数 axis の場合でも 1 つのみ指定。
factor =	(省略可、D=1.0)	normalization factor
title =	(省略可)	出力ファイルヘッダーのタイトル

[t-deposit] では、荷電粒子のエネルギーロスだけをタリーするので、タリーする領域に入る粒子ごとのエネルギーロスを part = で指定することはできません。領域に入る粒子毎のエネルギーロスをタリーするには、[counter] で part = を用いて、特定の入射粒子から引き起こされるエネルギーロスをカウンターで指定する必要があります。

output = dose で unit = 0 とした場合、[Gy/source] を単位とする吸収線量が出力されます。mesh = reg の場合は、volume パラメーターを用いるか、[volume] セクションにおいて各領域の体積を設定する必要があります。ただし、吸収線量は不可算量ですので、各 x 軸に関する積分値 (sum over) は出力されません。また、領域内に複数の物質が混在する場合、その領域の吸収線量は全体の平均値とならないことにご注意ください。例えば、領域の 1 と 2 にある質量 M_1 と M_2 の物質にそれぞれ E_1 と E_2 の熱量が吸収された場合、その平均吸収線量は $\frac{E_1+E_2}{M_1+M_2}$ ですが、$P_{HI}T_S$ で計算する値は $\frac{E_1}{M_1}+\frac{E_2}{M_2}$ となります。

output = deposit で deposit = 0 とした場合、part パラメータで指定した粒子種の total deposit (part = all の結果) に対する寄与を出力します。その際に、part パラメータで all が指定されていない時は、自動的に all が追加されます。一方、output = deposit で deposit = 1 とした場合、粒子種毎の deposit エネルギー分布が個別に出力されるため、各粒子種について得られた結果を足し上げた値と all を指定した結果は一致しませんのでご注意下さい。

output = deposit で deposit = 0 とした場合、part = all 以外のタリー結果に対する統計誤差は、istdev の値に関わらず、ヒストリー分散モードにより計算した標準偏差となります。

表 93: [t-deposit] パラメータ (3)

name	値	説明
angel =	(省略可)	angel パラメータ
2d-type =	1,2,3,4,5,6,7 (省略可)	2 次元表示のオプション
x-txt = y-txt = z-txt =	(省略可) (省略可) (省略可)	x-軸テキスト y-軸テキスト z-軸テキスト
gshow =	0(省略時), 1, 2, 3, 4	mesh=xyz, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3)、LAT 番号 (4) を表示
rshow =	0(省略時), 1, 2, 3	mesh=reg, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3) を表示。 この下に xyz 形状メッシュセクションが必要
ginfo =	0(省略時), 1 2	gshow, rshow オプションの時領域エラーチェック を行わない。領域エラーチェックを行い 2 次元ジオメトリ図上にエラー領域を表示する。 エラー領域表示に加えて領域エラーファイル (.err) を出力。
resol = width =	1(省略時) 0.5(省略時)	gshow, rshow オプションの時、領域境界を求める 分解能を各辺 resol 倍します。 gshow, rshow オプションの時、領域境界を表示する 線の太さを定義します。
volume reg vol	(省略可)	reg メッシュの時、 各領域の体積を定義します。 省略した場合、インプットエコーにデフォルト値が 表示されます。この行の下に volume 定義文が必要。 volume 定義文。書式は 5.1.2 を参照。
iechrl =	72(省略時)	volume のインプットエコーの時の最大コラム数
volmat =	(省略可、D=9)	xyz メッシュで material が指定されている時、 各メッシュの volume correction を行う。(0 で無し) volmat の値は、xyz メッシュ 1 辺のスキャン数
epsout =	0(省略時), 1, 2	1 で出力ファイルを ANGEL で処理した eps ファイル を作成。ファイル名は出力ファイルの拡張子を eps に 変えたファイル名。 2 とすることで更に誤差棒を表示する。ただし、 2 次元表示 (axis= xy, yz, xz, rz) の場合は無効。
bmpout =	0(省略時), 1	二次元ビットマップ画像ファイルを出力する。 ファイル名は出力ファイルの拡張子を bmp に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkout =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルを出力する。 ファイル名は出力ファイルの拡張子を vtk に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkfmt =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルのフォーマット。 テキスト形式（0）またはバイナリ形式（1）を選択する。

表 94: [t-deposit] パラメータ (4)

name	値	説明
ctmin(i) = ctmax(i) =	(省略可、D=-9999) (省略可、D= 9999)	i-th カウンターの最小値 i-th カウンターの最大値
trcl =	(省略可)	r-z, xyz メッシュの座標変換番号もしくは座標変換定義
gslat =	1(省略時), 0	0 で gshow で lattice の境界を非表示、1 で表示
dresol	(省略可, D=0.0)	検出器の分解能を表すパラメータ 1。output = deposit の場合のみ有効。dresol=σ_r, dfano=F の場合、各イベントの付与エネルギー E は標準偏差が $\sqrt{\sigma_r^2 + FE}$ のガウス分布に従って分散される。
dfano	(省略可, D=0.0)	検出器の分解能を表すパラメータ 2。output = deposit の場合のみ有効。dresol=σ_r, dfano=F の場合、各イベントの付与エネルギー E は標準偏差が $\sqrt{\sigma_r^2 + FE}$ のガウス分布に従って分散される。

一般に fano factor は無次元量として定義されていますが、dfano パラメータはエネルギーの次元をもつ量として定義されていますのでご注意ください。

6.6 [T-Deposit2] セクション

このタリーは、[t-deposit] を 2 つの領域で同時に実行し、deposit energy をタリーし、その相関などを見るものです。dE, E カウンターなどの模擬に用いることができ、2 つの領域の deposit energy の相関を 2 次元プロットすることなどができます。`[t-deposit]` において `output=deposit` とした場合と同様に、検出器の応答を模擬することができます。ただし、`[weight window]` などの使用により、ウエイトが変わるような計算を行った場合は、適切な結果が得られませんのでご注意ください。このタリーでは、[t-deposit] タリー同様、任意の物質の LET(dE/dx) の関数のユーザー定義のファクターを乗じることができます。これは、usrdfn1.f, usrdfn2.f で定義されます。サンプルプログラムとして、usrdfn1.f には粒子の LET に ICRP60 で定義された $Q(L)$ 関係を乗じて線量当量を導出するプログラムが、usrdfn2.f には単にエネルギーロスを返すプログラムが入っています。また、時間メッシュと [Timer] を組み合わせることによって、TOF のカウンターを模擬でき、時間と deposit energy の相関を 2 次元プロットすることもできます。

表 95: `[t-deposit2]` パラメータ (1)

name	値	説明
`mesh =` `reg =`	`reg` `r1 r2`	形状メッシュ、`reg` のみ 必ず２つの領域番号を指定
`part =`	`all`(省略時), 粒子名	ひとつの `[t-deposit2]` セクション最大 6 個まで

`letmat1 =`
 	
`letmat2 =`
	(省略可)
	
	(省略可)
		領域 `r1` に対する LET(dE/dx) を評価する物質番号。
		省略時は、実際の物質。
		領域 `r2` に対する LET(dE/dx) を評価する物質番号。
		省略時は、実際の物質。
		物質を指定する場合の密度は、`[Material]` セクションで定義された密度となる。したがって、水に対する LET 分布を計算したい場合、`[Material]` セクションで、水の密度が 1g/cm^3 となるように定義する必要がある。
		また、負の場合は電子・陽電子の dE/dx として 1g/cm^3 の水に対する値が参照される。詳しくは、奨励設定の particletherapy を参照してください。

`dedxfnc1 =`
 	
`dedxfnc2 =`
	(省略可, D=0)
	
	(省略可, D=0)
		領域 `r1` に対して、0:ファクター無し、
		1: usrdfn1 を使用、2: usrdfn2 を使用。
		領域 `r2` に対して、0:ファクター無し、
		1: usrdfn1 を使用、2: usrdfn2 を使用。
		サンプルプログラムとして、usrdfn1.f には粒子の LET に ICRP60 で定義された $Q(L)$ 関係を乗じて線量当量を導出するプログラムが、usrdfn2.f には単にエネルギーロスを返すプログラムが入っています。

`e1-type =`
 	
`e2-type =`
	1, 2, 3, 4, 5
	
	1, 2, 3, 4, 5
		領域 `r1` に対するエネルギーメッシュ、
		エネルギーメッシュサブセクションが必要。
		領域 `r2` に対するエネルギーメッシュ、
		エネルギーメッシュサブセクションが必要。

`t-type =`	1, 2, 3, 4, 5 (省略可)	時間メッシュ、時間メッシュサブセクションが必要

表 96: [t-deposit2] パラメータ (2)

name	値	説明
unit =	1, 2	1: Number [1/source] 2: Number [1/nsec/source]
axis =	eng1, eng2, t, e12, e21, t-e1, t-e2, e1-t, e2-t	出力データの x 軸 2 次元表示
file =	file name	axis の数だけ定義する
resfile =	(省略可, D=file)	再開始計算時の過去タリーファイル名。複数 axis の場合でも 1 つのみ指定。
factor =	(省略可、D=1.0)	normalization factor
title =	(省略可)	出力ファイルヘッダーのタイトル
angel =	(省略可)	angel パラメータ
2d-type =	1,2,3,4,5,6,7 (省略可)	2 次元表示のオプション
x-txt = y-txt = z-txt =	(省略可) (省略可) (省略可)	x-軸テキスト y-軸テキスト z-軸テキスト
volume reg vol	(省略可)	reg メッシュの時、 各領域の体積を定義します。 省略した場合、インプットエコーにデフォルト値が表示されます。この行の下に volume 定義文が必要。 volume 定義文。書式は 5.1.2 を参照。
iechrl =	72(省略時)	volume のインプットエコーの時の最大コラム数
epsout =	0(省略時), 1	1 で出力ファイルを $A_N G_E L$ で処理した eps ファイルを作成。ファイル名は出力ファイルの拡張子を eps に変えたファイル名
ctmin(i) = ctmax(i) =	(省略可、D=-9999) (省略可、D= 9999)	i-th カウンターの最小値 i-th カウンターの最大値

[t-deposit2] では、荷電粒子のエネルギーロスだけをタリーするので、タリーする領域に入る粒子ごとのエネルギーロスを part = で指定することはできません。領域に入る粒子毎のエネルギーロスをタリーするには、[counter] で part = を用いて、特定の入射粒子から引き起こされるエネルギーロスをカウンターで指定する必要があります。

6.7 [T-Yield] セクション

生成核種を出力します。 dmax(2) 以下のエネルギーの中性子による生成核種はタリーされません。ただし、e-mode≥1 の時は、タリーされます。

表 97: [t-yield] パラメータ (1)

name	値	説明
mesh =	reg, r-z, xyz	形状メッシュ、形状メッシュサブセクションが必要
special =	D=0 (省略可)	> 0 の時、タリー領域での核反応を special 回繰り返し、また統計崩壊を反応あたり 10 回繰り返し、核種生成断面積の統計を上げる。
part =	all(省略時) 反応入射粒子名	ひとつの [t-yield] セクション最大 6 個まで 入射粒子を限定する
material =	(省略可) all, 数 (次行)	スコアする material を限定する。複数定義可。 all : デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの material を次の行に記述する。 負の数にした場合は、それらの material を対象から外すことを意味する。 material 番号
mother =	(省略可) all, 数 (次行)	核種生成をする母核を限定する。複数定義可。 all : デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの核種を次の行に記述する。 負の数にした場合は、それらの母核を対象から外すことを意味する。 質量数を指定すれば、その核。 質量数を指定しなければ、Pb の同位体全てを指定。 複数の母核群を指定したいときは、複数の [t-yield] セクションを定義する。
nucleus =	(省略可) all, 数 (次行)	出力する核種生成を限定する。複数定義可。 all : デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの核種を次の行に記述する。 質量数を指定すれば、その核。 質量数を指定しなければ、Pb の同位体全てを指定。
elastic =	1（省略時）,0,-1	弾性散乱による反跳原子核の寄与を含めるかどうかのオプション。 1: 含める 0: elastic collision の recoil された標的核を除く -1: 標的核と同じ原子核が生成された場合は除く
unit =	1, 2	1: [1/source] 2: [1/cm^3/source]

表 98: [t-yield] パラメータ (2)

name	値	説明
ndata =	0(省略時), 1	1 を指定すると、核子入射で標的核が α, ^{14}N, ^{16}O の場合の核種生成断面積に核データを用います。
axis =	reg, x, y, z, r, xy, yz, xz, rz mass charge chart dchain	出力データの x 軸 2 次元表示 質量分布を表示。nucleus を指定した場合 isotope 分布 荷電分布を表示。nucleus の指定不可 nucleus chart 形式 (x 軸:N, y 軸:Z)。 nucleus の指定不可 dchain-sp 用の出力。全ての isotope を出力。 mesh = reg のみ [t-yield] のみ、1 つのタリーにつき 1 つの axis しか指定できません。
file =	file name	出力ファイル名を定義する。
resfile =	(省略可, D=file)	再開始計算時の過去タリーファイル名。
output =	(省略可) product cutoff	タリーするタイミングを変更します。省略時は product 核反応で生成された核種をタリーする。デフォルト。 エネルギーカットオフで止まった核種をタリーする。 原子核を輸送させない場合は、product と同じになる。
info =	0, 1	chart のとき stable nuclei と magic number を出力
factor =	(省略可、D=1.0)	normalization factor
title =	(省略可)	出力ファイルヘッダーのタイトル
angel =	(省略可)	angel パラメータ
2d-type =	1,2,3,4,5,6,7 (省略可)	2 次元表示のオプション
x-txt = y-txt = z-txt =	(省略可) (省略可) (省略可)	x-軸テキスト y-軸テキスト z-軸テキスト

ndata = 1 の時、使用する核データの反応チャンネルは、以下のものです。

^{4}He$(n,x)^{3}$H	^{14}N$(n,x)^{3}$H	^{14}N$(n,x)^{7}$Be	^{14}N$(n,x)^{11}$Be	^{14}N$(n,x)^{10}$C	^{14}N$(n,x)^{11}$C
^{14}N$(n,x)^{14}$C	^{14}N$(n,x)^{13}$N	^{16}O$(n,x)^{3}$H	^{16}O$(n,x)^{7}$Be	^{16}O$(n,x)^{11}$Be	^{16}O$(n,x)^{10}$C
^{16}O$(n,x)^{11}$C	^{16}O$(n,x)^{14}$C	^{16}O$(n,x)^{15}$C	^{16}O$(n,x)^{13}$N	^{16}O$(n,x)^{16}$N	^{16}O$(n,x)^{14}$O
^{16}O$(n,x)^{15}$O	^{4}He$(p,x)^{3}$H	^{14}N$(p,x)^{7}$Be	^{14}N$(p,x)^{11}$Be	^{14}N$(p,x)^{10}$C	^{14}N$(p,x)^{11}$C
^{14}N$(p,x)^{13}$N	^{14}N$(p,x)^{14}$O	^{16}O$(p,x)^{3}$H	^{16}O$(p,x)^{7}$Be	^{16}O$(p,x)^{11}$Be	^{16}O$(p,x)^{10}$C
^{16}O$(p,x)^{11}$C	^{16}O$(p,x)^{14}$C	^{16}O$(p,x)^{13}$N	^{16}O$(p,x)^{14}$O	^{16}O$(p,x)^{15}$O	

output=cutoff を指定したときは、part, mother の指定は無視されます。

[parameters] セクションにおいて igamma=3 とすると、γ 崩壊とアイソマー生成が EBITEM モデルを用いて考慮されます。その際、axis=chart, dchain とした場合に、生成されたアイソマーの情報も出力されます。

表 99: [t-yield] パラメータ (3)

name	値	説明
gshow =	0(省略時) 1, 2, 3, 4	mesh=xyz, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3)、LAT 番号 (4) を表示
rshow =	0(省略時) 1, 2, 3	mesh=reg, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3) を表示。 この下に xyz 形状メッシュセクションが必要
ginfo =	0(省略時), 1 2	gshow, rshow オプションの時領域エラーチェックを行わない。 領域エラーチェックを行い 2 次元ジオメトリ図上にエラー領域 を表示する。 エラー領域表示に加えて領域エラーファイル (.err) を出力。
resol = width =	1(省略時) 0.5(省略時)	gshow, rshow オプションの時、領域境界を求める 分解能を各辺 resol 倍します。 gshow, rshow オプションの時、領域境界を表示する 線の太さを定義します。
volume reg vol	(省略可)	reg メッシュの時、 各領域の体積を定義します。 省略した場合、インプットエコーにデフォルト値が 表示されます。この行の下に volume 定義文が必要。 volume 定義文。書式は 5.1.2 を参照。
iechrl =	72(省略時)	volume のインプットエコーの時の最大コラム数
volmat =	(省略可、D=9)	xyz メッシュで material が指定されている時、 各メッシュの volume correction を行う。(0 で無し) volmat の値は、xyz メッシュ 1 辺のスキャン数
epsout =	0(省略時), 1	1 で出力ファイルを $A_N G_E L$ で処理した eps ファイルを作成。 ファイル名は出力ファイルの拡張子を eps に変えたファイル名
bmpout =	0(省略時), 1	二次元ビットマップ画像ファイルを出力する。 ファイル名は出力ファイルの拡張子を bmp に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkout =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルを出力する。 ファイル名は出力ファイルの拡張子を vtk に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkfmt =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルのフォーマット。 テキスト形式（0）またはバイナリ形式（1）を選択する。
ctmin(i) = ctmax(i) =	(省略可、D=-9999) (省略可、D= 9999)	i-th カウンターの最小値 i-th カウンターの最大値
trcl =	(省略可)	r-z, xyz メッシュの座標変換番号もしくは座標変換定義
gslat =	1(省略時) 0	1 で gshow で lattice の境界を表示 0 で gshow で lattice の境界を非表示

6.8 [T-Product] セクション

核反応による生成粒子、ソースの発生粒子をタリーします。[t-yield] との違いは、エネルギー分布や、時間分布が取れることです。低エネルギーの中性子、光子、電子の反応からの粒子は、タリーしていません。ただし、e-mode≥1 を用いると核データを用いた中性子反応からの生成粒子、残留核をタリーすることができます。

表 100: [t-product] パラメータ (1)

name	値	説明
mesh =	reg, r-z, xyz	形状メッシュ、形状メッシュサブセクションが必要
part =	all(省略時), 粒子名	ひとつの [t-product] セクション最大 6 個まで
material = (次行)	(省略可) all, 数 2 5 8	スコアする material を限定する。複数定義可。 all : デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの material を次の行に記述する。 負の数にした場合は、それらの material を対象から外すことを意味する。 material 番号
mother = (次行)	(省略可) all, 数 208Pb Pb	核反応をする母核を限定する。複数定義可。 all : デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの核種を次の行に記述する。負の数にした場合は、それらの母核を対象から外すことを意味する。 質量数を指定すれば、その核。 質量数を指定しなければ、Pb の同位体全てを指定。 複数の母核群を指定したいときは、複数の [t-product] セクションを定義する。
e-type =	1, 2, 3, 4, 5	エネルギーメッシュ エネルギーメッシュサブセクションが必要
t-type =	1, 2, 3, 4, 5 (省略可)	時間メッシュ、時間メッシュサブセクションが必要
a-type =	1, 2, -1, -2 (省略可)	角度メッシュ (1, 2 :cos, −1, −2 :degree) 角度メッシュサブセクションが必要

表 101: [t-product] パラメータ (2)

name	値	説明
unit =	1, 2, 3, 4, 5, 6	1: [1/source]
		2: [1/cm^3/source]
		3: [1/MeV/source]
		4: [1/cm^3/MeV/source]
		5: [1/Lethargy/source]
		6: [1/cm^3/Lethargy/source]
	11, 12, 13, 14, 15, 16	11:[1/nsec/source]
		12:[1/cm^3/nsec/source]
		13:[1/MeV/nsec/source]
		14:[1/cm^3/MeV/nsec/source]
		15:[1/Lethargy/nsec/source]
		16:[1/cm^3/Lethargy/nsec/source]
	21, 22, 23, 24, 25, 26	21: [1/sr/source]
		22: [1/cm^3/sr/source]
		23: [1/MeV/sr/source]
		24: [1/cm^3/MeV/sr/source]
		25: [1/Lethargy/sr/source]
		26: [1/cm^3/Lethargy/sr/source]
	31, 32, 33, 34, 35, 36	31:[1/nsec/sr/source]
		32:[1/cm^3/nsec/sr/source]
		33:[1/MeV/nsec/sr/source]
		34:[1/cm^3/MeV/nsec/sr/source]
		35:[1/Lethargy/nsec/sr/source]
		36:[1/cm^3/Lethargy/nsec/sr/source]

unit = 5, 6, 15, 16, 25, 26, 35, 36 の場合の Lethargy はエネルギーに関する自然対数目盛を表しています。各エネルギービンの上限と下限がそれぞれ $E_{\mathrm{high}}, E_{\mathrm{low}}$ のときに、各 Lethargy の幅を $\ln(E_{\mathrm{high}}/E_{\mathrm{low}})$ で与えます。

unit = 21 - 26, 31 - 36 の場合の sr は、立体角の単位ステラジアン (steradian) を表しています。

表 102: [t-product] パラメータ (3)

name	値	説明
axis =	eng, reg, x, y, z, r, xy, yz, xz, rz t, cos, the	出力データの x 軸 2 次元表示 時間 (t)、角度 (cos, the)
file =	file name	axis の数だけ定義する
resfile =	(省略可, D=file)	再開始計算時の過去タリーファイル名。複数 axis の 場合でも 1 つのみ指定。
output =	source nuclear (default) nonela elastic decay fission	ソースの発生粒子 核反応からの生成粒子、elastic を含む 非弾性散乱からの生成粒子 弾性散乱からの生成粒子 崩壊反応からの生成粒子 核分裂からの生成粒子
factor =	(省略可、D=1.0)	normalization factor
title =	(省略可)	出力ファイルヘッダーのタイトル
angel =	(省略可)	angel パラメータ
2d-type =	1,2,3,4,5,6,7 (省略可)	2 次元表示のオプション
x-txt = y-txt = z-txt =	(省略可) (省略可) (省略可)	x-軸テキスト y-軸テキスト z-軸テキスト
gshow =	0(省略時), 1, 2, 3, 4	mesh=xyz, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3)、LAT 番号 (4) を表示
rshow =	0(省略時), 1, 2, 3	mesh=reg, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3) を表示。 この下に xyz 形状メッシュセクションが必要
ginfo =	0(省略時), 1 2	gshow, rshow オプションの時領域エラーチェックを行わない。 領域エラーチェックを行い 2 次元ジオメトリ図上にエラー領域 を表示する。 エラー領域表示に加えて領域エラーファイル (.err) を出力。
resol = width =	1(省略時) 0.5(省略時)	gshow, rshow オプションの時、領域境界を求める 分解能を各辺 resol 倍します。 gshow, rshow オプションの時、領域境界を表示する 線の太さを定義します。
volume reg vol	(省略可)	reg メッシュの時、 各領域の体積を定義します。 省略した場合、インプットエコーにデフォルト値が 表示されます。この行の下に volume 定義文が必要。 volume 定義文。書式は 5.1.2 を参照。
iechrl =	72(省略時)	volume のインプットエコーの時の最大コラム数
volmat =	(省略可、D=9)	xyz メッシュで material が指定されている時、 各メッシュの volume correction を行う。(0 で無し) volmat の値は、xyz メッシュ 1 辺のスキャン数

表 103: [t-product] パラメータ (4)

name	値	説明
epsout =	0(省略時), 1, 2	1 で出力ファイルを ANGEL で処理した eps ファイル を作成。ファイル名は出力ファイルの拡張子を eps に 変えたファイル名。 2 とすることで更に誤差棒を表示する。ただし、 2 次元表示 (axis= xy, yz, xz, rz) の場合は無効。
bmpout =	0(省略時), 1	二次元ビットマップ画像ファイルを出力する。 ファイル名は出力ファイルの拡張子を bmp に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkout =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルを出力する。 ファイル名は出力ファイルの拡張子を vtk に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkfmt =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルのフォーマット。 テキスト形式（0）またはバイナリ形式（1）を選択する。
ctmin(i) = ctmax(i) =	(省略可、D=-9999) (省略可、D= 9999)	i-th カウンターの最小値 i-th カウンターの最大値
trcl =	(省略可)	r-z, xyz メッシュの座標変換番号もしくは座標変換定義
dump = (次行)	データ数 データ定義	ファイルに粒子の情報をダンプします。mesh=reg のみ。 正で与えた時はバイナリー、負の時はアスキーファイル。 データの順番を定義します。ただし、idmpmode=1 の使用の ためには、nocas および nobch の情報が必要です。
gslat =	1(省略時), 0	0 で gshow で lattice の境界を非表示、1 で表示

[t-product] で、dump を指定したときは、e-type, t-type の指定は、上限と下限の意味しかもちません。unit は 1 に固定です。dump 定義文を用いてダンプさせる時は、axis、file 数ともにひとつに制限されます。dump するデータを書き込むファイルは、file 定義文で指定した名前に “_dmp” を付けたファイルです。file 定義文で定義された file 名には、通常のタリーの結果が書き出されます。（バージョン 2.66 以前では、file 定義文で指定したファイルに dump するデータが書き込まれ、通常のタリー結果を出力するファイルとして .cfg をつけたファイルが作成されていました。）dump ファイルを使用した接続計算を idmpmode=1 で実行するには、nocas および nobch の情報を出力しておく必要があります。また、idmpmode=1 を使用するためには、ここで出力される “_dmp” が付いた dump ファイルと共に file=で指定した名前の通常のタリー出力ファイルが必要になります。メモリ共有型並列計算を実行する場合、本機能による dump ファイルの作成と dumpall によるイベント情報の書き出しは共存できませんのでご注意ください。

ソースに dump ファイルを用い、[t-product] で、dump を指定し、output = source、icntl = 6 を指定すると、dump ファイルの加工ができます。既存の dump ファイルから、特定の粒子や、特定のエネルギーの情報、また、情報の数を減らしたファイルなどを、新しい dump ファイルとして作成できます。

6.9 [T-DPA] セクション

“原子あたりのはじき出し数 (DPA; Displacement Per Atom)” の関数で評価される放射線損傷は、原子炉や加速器施設等で使用される材料の劣化を表す指標で、大強度ビーム、特に重イオンに関して重要な課題の一つです。原子のはじき出しは入射粒子と標的原子の衝突を起源として起こり、飛ばされた原子 (一次はじき出し原子, PKA; Primary Knock on Atom) が再度別の原子に衝突することによって連鎖的に発生します。このタリーではその頻度を DPA として出力します。輸送荷電粒子のクーロン散乱断面積からの寄与を含んでおり、重イオン入射の場合にも適切に評価されます。また、e-mode≥1 を用いて、低エネルギーの中性子入射反応から生成する荷電粒子による DPA の寄与も計算できます。

表 104: [t-dpa] パラメータ (1)

name	値	説明
mesh =	reg, r-z, xyz	形状メッシュ、形状メッシュサブセクションが必要
part =	all(省略時), 反応入射粒子名	ひとつの [t-dpa] セクション最大 6 個まで
material = (次行)	(省略可) all, 数 2 5 8	スコアする material を限定する。複数定義可。 all : デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの material を次の行に記述する。 負の数にした場合は、それらの material を対象から外すことを意味する。 material 番号
mother = (次行)	(省略可) all, 数 208Pb Pb	スコアする母核を限定する。複数定義可。 all : デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの核種を次の行に記述する。負の数にした場合は、それらの母核を対象から外すことを意味する。 質量数を指定すれば、その核。 質量数を指定しなければ、Pb の同位体全てを指定。 複数の母核群を指定したいときは、複数の [t-dpa] セクションを定義する。
unit =	1, 2	1: [DPA*1.e-24/source] 2: [DPA/source]
axis =	reg, x, y, z, r, xy, yz, xz, rz	出力データの x 軸 2 次元表示
file =	file name	axis の数だけ定義する
resfile =	(省略可, D=file)	再開始計算時の過去タリーファイル名。複数 axis の場合でも 1 つのみ指定。
angel =	(省略可)	angel パラメータ
2d-type =	1,2,3,4,5,6,7 (省略可)	2 次元表示のオプション
x-txt = y-txt = z-txt =	(省略可) (省略可) (省略可)	x-軸テキスト y-軸テキスト z-軸テキスト

表 105: [t-dpa] パラメータ (2)

name	値	説明
output =	dpa simple all	total DPA の出力 total: 全体の DPA cutoff1: 生成荷電粒子のエネルギーが、計算打切 エネルギー (emin) 以下の場合の DPA cutoff2: 輸送された荷電粒子のエネルギーが、計 算打切エネルギー (emin) 以下になった場合の DPA transpt: 輸送による全ての荷電粒子の DPA simple に加え、PKA として d, t, ^{3}He, α, Nucleus の寄与の出力。
factor =	(省略可、D=1.0)	normalization factor
title =	(省略可)	出力ファイルヘッダーのタイトル
gshow =	0(省略時), 1, 2, 3, 4	mesh=xyz, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3)、LAT 番号 (4) を表示
rshow =	0(省略時), 1, 2, 3	mesh=reg, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3) を表示。 この下に xyz 形状メッシュセクションが必要
ginfo =	0(省略時), 1 2	gshow, rshow オプションの時領域エラーチェックを行わない。 領域エラーチェックを行い 2 次元ジオメトリ図上にエラー領域 を表示する。 エラー領域表示に加えて領域エラーファイル (.err) を出力。
resol = width =	1(省略時) 0.5(省略時)	gshow, rshow オプションの時、領域境界を求める 分解能を各辺 resol 倍します。 gshow, rshow オプションの時、領域境界を表示する 線の太さを定義します。
volume reg vol	(省略可)	reg メッシュの時、 各領域の体積を定義します。 省略した場合、インプットエコーにデフォルト値が 表示されます。この行の下に volume 定義文が必要。 volume 定義文。書式は 5.1.2 を参照。
iechrl =	72(省略時)	volume のインプットエコーの時の最大コラム数
volmat =	(省略可、D=9)	xyz メッシュで material が指定されている時、 各メッシュの volume correction を行う。(0 で無し) volmat の値は、xyz メッシュ 1 辺のスキャン数

表 106: [t-dpa] パラメータ (3)

name	値	説明
epsout =	0(省略時) 1	1 で出力ファイルを $A_N G_E L$ で処理した eps ファイルを作成。ファイル名は出力ファイルの拡張子を eps に変えたファイル名
bmpout =	0(省略時), 1	二次元ビットマップ画像ファイルを出力する。 ファイル名は出力ファイルの拡張子を bmp に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkout =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルを出力する。 ファイル名は出力ファイルの拡張子を vtk に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkfmt =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルのフォーマット。 テキスト形式（0）またはバイナリ形式（1）を選択する。
ctmin(i) = ctmax(i) =	(省略可、D=-9999) (省略可、D= 9999)	i-th カウンターの最小値 i-th カウンターの最大値
trcl =	(省略可)	r-z, xyz メッシュの座標変換番号もしくは座標変換定義
gslat =	1(省略時), 0	0 で gshow で lattice の境界を非表示、1 で表示

6.10 [T-LET] セクション

任意の物質の LET(dE/dx) の関数として、track length や、dose をタリーします。[t-heat] と異なり、ここでの dose は、荷電粒子のエネルギーロスだけをタリーします。LET の単位は、keV/μm です。[T-LET] を使い、中性子の輸送計算を行う場合は、Event Generator mode (e-mode≥1) にする必要があります。また、電子に関しては、計算打切エネルギー (emin) 以下の粒子の寄与はタリーさせませんので、ご注意ください。

表 107: [t-let] パラメータ (1)

name	値	説明
mesh =	reg, r-z, xyz	形状メッシュ、形状メッシュサブセクションが必要
part =	all(省略時), 粒子名	ひとつの [t-let] セクション最大 6 個まで
material = (次行)	(省略可) all, 数 2 5 8	スコアする material を限定する。複数定義可。 all : デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの material を次の行に記述する。 負の数にした場合は、それらの material を対象から外すことを意味する。 material 番号
letmat =	(省略可)	LET(dE/dx) を評価する物質番号、省略時は、実際の物質。物質を指定する場合の密度は、[Material] セクションで定義された密度となる。したがって、水に対する LET 分布を計算したい場合、[Material] セクションで、水の密度が 1g/cm^3 となるように定義する必要がある。 また、負の場合は電子・陽電子の dE/dx として 1g/cm^3 の水に対する値が参照される。詳しくは、奨励設定の particletherapy を参照してください。
l-type =	1, 2, 3, 4, 5	LET メッシュ [keV/μm] LET メッシュサブセクションが必要。 ただし、メッシュを細かく設定しすぎる (1 桁あたり 20 分割以上) と、得られた LET スペクトルに物理的に意味のないピークが生じますのでご注意ください。

表 108: [t-let] パラメータ (2)

name	値	説明
unit =	1, 2, 3, 4, 5, 6 7, 8, 9, 10, 11, 12	1: Track [cm/(keV/μm)/source] 2: Dose [MeV/(keV/μm)/source] 3: Track [cm/ln(keV/μm)/source] 4: Dose [MeV/ln(keV/μm)/source] 5: Track [cm/source] 6: Dose [MeV/source] 7: Track [1/cm^2/(keV/μm)/source] 8: Dose [MeV/cm^3/(keV/μm)/source] 9: Track [1/cm^2/ln(keV/μm)/source] 10:Dose [MeV/cm^3/ln(keV/μm)/source] 11:Track [1/cm^2/source] 12:Dose [MeV/cm^3/source]
axis =	let, reg, x, y, z, r, xy, yz, xz, rz	出力データの x 軸 2 次元表示、
file =	file name	axis の数だけ定義する
resfile =	(省略可, D=file)	再開始計算時の過去タリーファイル名。複数 axis の 場合でも 1 つのみ指定。
factor =	(省略可、D=1.0)	normalization factor
title =	(省略可)	出力ファイルヘッダーのタイトル
angel =	(省略可)	angel パラメータ
2d-type =	1,2,3,4,5,6,7 (省略可)	2 次元表示のオプション
x-txt = y-txt = z-txt =	(省略可) (省略可) (省略可)	x-軸テキスト y-軸テキスト z-軸テキスト
gshow =	0(省略時), 1, 2, 3, 4	mesh=xyz, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3)、LAT 番号 (4) を表示
rshow =	0(省略時), 1, 2, 3	mesh=reg, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3) を表示 この下に xyz 形状メッシュセクションが必要
ginfo =	0(省略時), 1 2	gshow, rshow オプションの時領域エラーチェックを行わない。 領域エラーチェックを行い 2 次元ジオメトリ図上にエラー領域 を表示する。 エラー領域表示に加えて領域エラーファイル (.err) を出力。
resol = width =	1(省略時) 0.5(省略時)	gshow, rshow オプションの時、領域境界を求める 分解能を各辺 resol 倍します。 gshow, rshow オプションの時、領域境界を表示する 線の太さを定義します。

表 109: [t-let] パラメータ (3)

name	値	説明
volume reg vol	(省略可)	reg メッシュの時、 各領域の体積を定義します。 省略した場合、インプットエコーにデフォルト値が 表示されます。この行の下に volume 定義文が必要。 volume 定義文。書式は 5.1.2 を参照。
iechrl =	72(省略時)	volume のインプットエコーの時の最大コラム数
volmat =	(省略可、D=9)	xyz メッシュで material が指定されている時、 各メッシュの volume correction を行う。(0 で無し) volmat の値は、xyz メッシュ 1 辺のスキャン数
epsout =	0(省略時), 1	1 で出力ファイルを $A_N G_E L$ で処理した eps ファイル を作成。ファイル名は出力ファイルの拡張子を eps に 変えたファイル名
ctmin(i) = ctmax(i) =	(省略可、D=-9999) (省略可、D= 9999)	i-th カウンターの最小値 i-th カウンターの最大値
trcl =	(省略可)	r-z, xyz メッシュの座標変換番号もしくは座標変換定義
gslat =	1(省略時), 0	0 で gshow で lattice の境界を非表示、1 で表示

6.11 [T-SED] セクション

PHITS は、巨視的な体系内での放射線挙動を解析する計算コードですので、マイクロドジメトリなどで使われる DNA や細胞サイズの微視的な領域内での付与エネルギー (lineal energy (y) もしくは specific energy (z)) 分布を、[t-deposit] や [t-heat] などを用いて直接計算することはできません。そこで、飛跡構造解析の結果に基づいて構築した計算式を用いて、微小領域におけるエネルギー付与分布を計算するタリー [t-sed] を導入しました。このタリーを用いれば、δ 線やオージェ電子による寄与も考慮して、微小領域における y 分布や z 分布を計算することができます。なお、sed の名前の由来は、Specific Energy Distribution です。計算方法の詳細は、下記の文献[35] [36] を参照してください。

[t-sed] は任意の物質中の微小領域内付与エネルギー分布をタリーします。ただし、水以外の物質に対する精度検証は行っていませんので、ご注意ください。その際、[t-heat] と異なり、dose は、荷電粒子のエネルギーロスだけをタリーします。したがって、[T-SED] を使い中性子の輸送計算を行う場合は、event generator mode (e-mode≥1) にする必要があります。微小領域の付与エネルギーを表す単位として、deposit energy ε (MeV), lineal energy y (keV/μm), specific energy z (Gy) があります。各単位の詳細は、ICRU Report36 をご参照ください。

表 110: `[t-sed]` パラメータ (1)

name	値	説明
mesh =	reg, r-z, xyz	形状メッシュ、形状メッシュサブセクションが必要
part =	all(省略時), 粒子名	ひとつの [t-sed] セクション最大 6 個まで
material =	(省略可) all, 数 (次行)	スコアする material を限定する。複数定義可。 all : デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの material を次の行に記述する。 負の数にした場合は、それらの material を対象から外すことを意味する。 material 番号
letmat =	(省略可)	LET(dE/dx) を評価する物質番号、省略時は、実際の物質。物質を指定する場合の密度は、[Material] セクションで定義された密度となる。したがって、水に対する LET 分布を計算したい場合、[Material] セクションで、水の密度が 1g/cm^3 となるように定義する必要がある。水以外の物質に対する精度検証は行っていませんので、ご注意ください。また、負の場合は電子・陽電子の dE/dx として 1g/cm^3 の水に対する値が参照される。詳しくは、奨励設定の particletherapy を参照してください。
se-unit =	1,2,3	微小領域の付与エネルギー単位 1: Deposit energy ε [MeV] 2: Lineal energy y [keV/μm] 3: Specific energy z [Gy]
cdiam =	(省略可、D=1,0)	微小領域（球）の直径。単位は μm

[35] T. Sato, R. Watanabe and K. Niita, “Development of a calculation method for estimating the specific energy distribution in complex radiation fields”, Radiat. Prot. Dosim. 122, 41-45 (2006).

[36] T. Sato, Y. Kase, R. Watanabe, K. Niita and L. Sihver, “Biological dose estimation for charged-particle therapy using an improved PHITS code coupled with a microdosimetric kinetic model”, Radiat. Res. 171, 107-117 (2009).

表 111: [t-sed] パラメータ (2)

name	値	説明
se-type =	1, 2, 3, 4, 5	付与エネルギーメッシュ(単位は se-unit で指定)。 エネルギーメッシュサブセクションが必要 (ne, emin, emax などで指定)。 実行中に'Warning: Z bin is not enough!' と出た場合は、ここで指定する specific energy（もしくは lineal energy) の範囲が狭すぎて、マイクロドジメトリ関数で分布が正しく計算できないことを示しています。emin を下げる、emax を上げる、ne を大きくするなどの対応が必要です。微少領域の直径が 1μm の Lineal energy を計算する場合 (cdiam = 1.0, se-unit = 2)、se-type =3, emin = 0.01, emax = 10000.0, ne = 60 以上が適当です。
unit =	1, 2, 3, 4, 5, 6	1: Track [cm/(keV/μm)/source] 2: Dose [MeV/(keV/μm)/source] $y * f(y)$ に相当 3: Track [cm/ln(keV/μm)/source] 4: Dose [MeV/ln(keV/μm)/source] $y * d(y)$ に相当 5: Track [cm/source] 6: Dose [MeV/source] (se-unit=2 の場合。se-unit=1,3 の場合は、(keV/μm) が MeV もしくは Gy となる。)
axis =	sed, reg, x, y, z, r, xy, yz, xz, rz	出力データの x 軸 2 次元表示、
file =	file name	axis の数だけ定義する
resfile =	(省略可, D=file)	再開始計算時の過去タリーファイル名。複数 axis の場合でも 1 つのみ指定。
factor =	(省略可、D=1.0)	normalization factor
title =	(省略可)	出力ファイルヘッダーのタイトル
angel =	(省略可)	angel パラメータ
2d-type =	1,2,3,4,5,6,7 (省略可)	2 次元表示のオプション
x-txt = y-txt = z-txt =	(省略可) (省略可) (省略可)	x-軸テキスト y-軸テキスト z-軸テキスト

表 112: [t-sed] パラメータ (3)

name	値	説明
gshow =	0(省略時), 1, 2, 3, 4	mesh=xyz, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3)、LAT 番号 (4) を表示
rshow =	0(省略時), 1, 2, 3	mesh=reg, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3) を表示。 この下に xyz 形状メッシュセクションが必要
ginfo =	0(省略時), 1 2	gshow, rshow オプションの時領域エラーチェックを行わない。 領域エラーチェックを行い 2 次元ジオメトリ図上にエラー領域を表示する。 エラー領域表示に加えて領域エラーファイル (.err) を出力。
resol = width =	1(省略時) 0.5(省略時)	gshow, rshow オプションの時、領域境界を求める分解能を各辺 resol 倍します。 gshow, rshow オプションの時、領域境界を表示する線の太さを定義します。
volmat =	(省略可、D=9)	xyz メッシュで material が指定されている時、 各メッシュの volume correction を行う。(0 で無し) volmat の値は、xyz メッシュ1 辺のスキャン数
epsout =	0(省略時), 1	1 で出力ファイルを $A_N G_E L$ で処理した eps ファイルを作成。ファイル名は出力ファイルの拡張子を eps に変えたファイル名
ctmin(i) = ctmax(i) =	(省略可、D=-9999) (省略可、D= 9999)	i-th カウンターの最小値 i-th カウンターの最大値
trcl =	(省略可)	r-z, xyz メッシュの座標変換番号もしくは座標変換定義
gslat =	1(省略時), 0	0 で gshow で lattice の境界を非表示、1 で表示

6.12 [T-Time] セクション

Energy cut off により輸送を止められた粒子と escape 粒子、decay 粒子の個数を時間タリーで出力します。時間の単位は nsec です。

このタリーを用いると、energy cut off 粒子や escape 粒子、decay 粒子のエネルギースペクトルを取ることができます。特に、energy cut off 粒子だけに注目して発生時のスペクトルを取るタリーは、このタリー以外にはありません。

表 113: [t-time] パラメータ (1)

name	値	説明
mesh =	reg, r-z, xyz	形状メッシュ、形状メッシュサブセクションが必要
part =	all(省略時), 粒子名	ひとつの [t-time] セクション最大 6 個まで
material =	(省略可) all, 数 (次行)	スコアする material を限定する。複数定義可。 all : デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの material を次の行 に記述する。 負の数にした場合は、それらの material を対象から外すことを意味する。 material 番号
t-type =	1, 2, 3, 4, 5	時間メッシュ 時間メッシュサブセクションが必要
e-type =	1, 2, 3, 4, 5	エネルギーメッシュ、エネルギーメッシュサブセクションが必要
unit =	1, 2, 3, 4	1: [1/source] 2: [1/nsec/source] 3: [1/nsec/cm^3/source] 4: [1/nsec/cm^3/MeV/source]
axis =	t, eng, reg, x, y, z, r, xy, yz, xz, rz	出力データの x 軸 2 次元表示
file =	file name	axis の数だけ定義する
resfile =	(省略可, D=file)	再開始計算時の過去タリーファイル名。複数 axis の 場合でも 1 つのみ指定。
output =	all cutoff escape decay	energy cut off 粒子、escape 粒子および decay 粒子 energy cut off 粒子 escape 粒子 decay 粒子
factor =	(省略可、D=1.0)	normalization factor
title =	(省略可)	出力ファイルヘッダーのタイトル
x-txt = y-txt = z-txt =	(省略可) (省略可) (省略可)	x-軸テキスト y-軸テキスト z-軸テキスト
angel =	(省略可)	angel パラメータ

表 114: [t-time] パラメータ (2)

name	値	説明
2d-type =	1,2,3,4,5,6,7 (省略可)	2 次元表示のオプション
gshow =	0(省略時), 1, 2, 3, 4	mesh=xyz, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3)、LAT 番号 (4) を表示
rshow =	0(省略時), 1, 2, 3	mesh=reg, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3) を表示 この下に xyz 形状メッシュセクションが必要。
ginfo =	0(省略時), 1 2	gshow, rshow オプションの時領域エラーチェックを行わない。 領域エラーチェックを行い 2 次元ジオメトリ図上にエラー領域を表示する。 エラー領域表示に加えて領域エラーファイル (.err) を出力。
resol = width =	1(省略時) 0.5(省略時)	gshow, rshow オプションの時、領域境界を求める分解能を各辺 resol 倍します。 gshow, rshow オプションの時、領域境界を表示する線の太さを定義します。
volume reg vol	(省略可)	reg メッシュの時、 各領域の体積を定義します。 省略した場合、インプットエコーにデフォルト値が表示されます。この行の下に volume 定義文が必要。 volume 定義文。書式は 5.1.2 を参照。
iechrl =	72(省略時)	volume のインプットエコーの時の最大コラム数
volmat =	(省略可、D=9)	xyz メッシュで material が指定されている時、 各メッシュの volume correction を行う。(0 で無し) volmat の値は、xyz メッシュ 1 辺のスキャン数
epsout =	0(省略時) 1	1 で出力ファイルを $A_N G_E L$ で処理した eps ファイルを作成。ファイル名は出力ファイルの拡張子を eps に変えたファイル名
bmpout =	0(省略時), 1	二次元ビットマップ画像ファイルを出力する。 ファイル名は出力ファイルの拡張子を bmp に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkout =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルを出力する。 ファイル名は出力ファイルの拡張子を vtk に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkfmt =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルのフォーマット。 テキスト形式（0）またはバイナリ形式（1）を選択する。
ctmin(i) = ctmax(i) =	(省略可、D=-9999) (省略可、D= 9999)	i-th カウンターの最小値 i-th カウンターの最大値
trcl =	(省略可)	r-z, xyz メッシュの座標変換番号もしくは座標変換定義
dump = (次行)	データ数 データ定義	ファイルに粒子の情報をダンプします。mesh=reg のみ。 正で与えた時はバイナリー、負の時はアスキーファイル。 データの順番を定義します。ただし、idmpmode=1 の使用のためには、nocas および nobch の情報が必要です。
gslat =	1(省略時), 0	0 で gshow で lattice の境界を非表示、1 で表示

`[t-time]` で、dump を用いることができるのは、output = cutoff だけです。dump を指定したときは、e-type, t-type の指定は、上限と下限の意味しかもちません。unit は 1 に固定です。dump 定義文を用いてダンプさせる時は、axis、file 数ともにひとつに制限されます。dump するデータを書き込むファイルは、file 定義文で指定した名前に "_dmp" を付けたファイルです。file 定義文で定義された file 名には、通常のタリーの結果が書き出されます。（バージョン 2.66 以前では、file 定義文で指定したファイルに dump するデータが書き込まれ、通常のタリー結果を出力するファイルとして `.cfg` をつけたファイルが作成されていました。）dump ファイルを使用した接続計算を `idmpmode=1` で実行するには、nocas および nobch の情報を出力しておく必要があります。また、`idmpmode=1` を使用するためには、ここで出力される "_dmp" が付いた dump ファイルと共に `file=`で指定した名前の通常のタリー出力ファイルが必要になります。メモリ共有型並列計算を実行する場合、本機能による dump ファイルの作成と `dumpall` によるイベント情報の書き出しは共存できませんのでご注意ください。

`[t-time]` で dump を用いると、ncut、gcut、pcut で条件付けられた *PHITs* の繋ぎ計算用ファイルを作成できます。

6.13 [T-Star] セクション

核反応が起こった回数を出力します。ライブラリーによる反応は、中性子だけがタリーされます。

表 115: [t-star] パラメータ (1)

name	値	説明
mesh =	reg, r-z, xyz	形状メッシュ、形状メッシュサブセクションが必要
part =	all(省略時), 反応入射粒子名	ひとつの [t-star] セクション最大 6 個まで
material = (次行)	(省略可) all, 数 2 5 8	スコアする material を限定する。複数定義可。 all : デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの material を次の行に記述する。 負の数にした場合は、それらの material を対象から外すことを意味する。 material 番号
mother = (次行)	(省略可) all, 数 208Pb Pb	核反応をする母核を限定する。複数定義可。 all : デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの核種を次の行に記述する。負の数にした場合は、それらの母核を対象から外すことを意味する。 質量数を指定すれば、その核。 質量数を指定しなければ、Pb の同位体全てを指定。 複数の母核群を指定したいときは、複数の [t-star] セクションを定義する。
e-type =	1, 2, 3, 4, 5	エネルギーメッシュ エネルギーメッシュサブセクションが必要
t-type =	1, 2, 3, 4, 5 (省略可)	時間メッシュ、時間メッシュサブセクションが必要
unit =	1, 2 11, 12	1: [1/cm^3/source] 2: [1/cm^3/MeV/source] 11:[1/cm^3/nsec/source] 12:[1/cm^3/nsec/MeV/source]
axis =	eng, reg, x, y, z, r, xy, yz, xz, rz t	出力データの x 軸 2 次元表示 時間軸
file =	file name	axis の数だけ定義する
resfile =	(省略可, D=file)	再開始計算時の過去タリーファイル名。複数 axis の場合でも 1 つのみ指定。

表 116: [t-star] パラメータ (2)

name	値	説明
output =	all decay elastic nuclear fission absorption heavyion transmut	全ての反応の star density decay 反応の star density elastic 反応の star density non-elastic + Hydrogen + HI 反応の star density fission 反応の star density absorption 反応の star density Heavy Ion 反応の star density 標的核が核変換を起こした場合の star density
factor =	(省略可、D=1.0)	normalization factor
title =	(省略可)	出力ファイルヘッダーのタイトル
angel =	(省略可)	angel パラメータ
2d-type =	1,2,3,4,5,6,7 (省略可)	2 次元表示のオプション
x-txt = y-txt = z-txt =	(省略可) (省略可) (省略可)	x-軸テキスト y-軸テキスト z-軸テキスト
gshow =	0(省略時), 1, 2, 3, 4	mesh=xyz, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3)、LAT 番号 (4) を表示
rshow =	0(省略時), 1, 2, 3	mesh=reg, axis=xy,yz,xz の時、領域境界 (1)、 物質番号 (2)、領域番号 (3) を表示 この下に xyz 形状メッシュセクションが必要
ginfo =	0(省略時), 1 2	gshow, rshow オプションの時領域エラーチェックを行わない。 領域エラーチェックを行い 2 次元ジオメトリ図上にエラー領域を表示する。 エラー領域表示に加えて領域エラーファイル (.err) を出力。
resol = width =	1(省略時) 0.5(省略時)	gshow, rshow オプションの時、領域境界を求める分解能を各辺 resol 倍します。 gshow, rshow オプションの時、領域境界を表示する線の太さを定義します。
volume reg vol	(省略可)	reg メッシュの時、 各領域の体積を定義します。 省略した場合、インプットエコーにデフォルト値が表示されます。この行の下に volume 定義文が必要。 volume 定義文。書式は 5.1.2 を参照。
iechrl =	72(省略時)	volume のインプットエコーの時の最大コラム数
volmat =	(省略可、D=9)	xyz メッシュで material が指定されている時、 各メッシュの volume correction を行う。(0 で無し) volmat の値は、xyz メッシュ 1 辺のスキャン数

表 117: [t-star] パラメータ (3)

name	値	説明
epsout =	0(省略時), 1	1 で出力ファイルを $A_N G_E L$ で処理した eps ファイルを作成。ファイル名は出力ファイルの拡張子を eps に変えたファイル名
bmpout =	0(省略時), 1	二次元ビットマップ画像ファイルを出力する。 ファイル名は出力ファイルの拡張子を bmp に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkout =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルを出力する。 ファイル名は出力ファイルの拡張子を vtk に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkfmt =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルのフォーマット。 テキスト形式（0）またはバイナリ形式（1）を選択する。
ctmin(i) = ctmax(i) =	(省略可、D=-9999) (省略可、D= 9999)	i-th カウンターの最小値 i-th カウンターの最大値
trcl =	(省略可)	r-z, xyz メッシュの座標変換番号もしくは座標変換定義
gslat =	1(省略時) 0	1 で gshow で lattice の境界を表示 0 で gshow で lattice の境界を非表示

6.14 [T-Dchain] セクション

このセクションでは、DCHAIN-SP を実行するための入力ファイルを作成します。*PHITS* と接続計算用に整備した DCHAIN-SP の接続計算の概要を図 34 に示します。

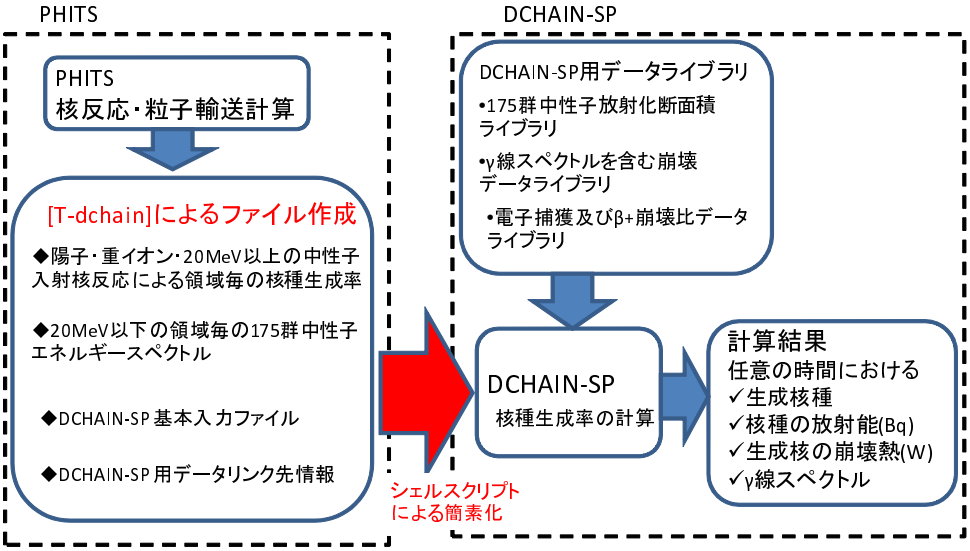


図 34: *PHITS* と DCHAIN-SP の接続計算の概要。

[t-dchain] は、1968 群の中性子スペクトル、核種生成率、DCHAIN-SP 用の基本入力ファイルを自動的に作成するためのタリーです。このタリーの出力を DCHAIN-SP に入力することにより、任意の時間における生成核種の放射能、崩壊熱、γ 線スペクトルを計算できます。

DCHAIN-SP は、20MeV 以下の中性子スペクトルに 1968 群の放射化断面積を乗じて 20MeV 以下の中性子による核種生成率を計算し、陽子、重イオン、中間子、20MeV 以上の高エネルギー中性子による核種生成率と足し合わせます。その後、崩壊データを利用して照射中及び照射終了後の任意の時刻における核種の蓄積量、誘導放射能、γ 線エネルギースペクトル及び α、β 及び γ 線の放出による崩壊熱等を評価します。DCHAIN-SP の詳細は、パッケージに含まれるマニュアル[37] をご参照ください。また、DCHAIN-SP の実行方法や例題については、\phits\recommendation\DCHAIN フォルダにある資料やサンプルインプットをご参照ください。

PHITS 計算における粒子輸送や [source] セクションで指定する線源の時間分布による経過時間は、DCHAIN-SP で考慮する経過時間とは無関係ですのでご注意ください。

なお、イベントジェネレータモードを使うと、低エネルギー中性子核反応による放射化も、DCHAIN-SP のデータライブラリを使わずに *PHITS* で直接計算することができます。ただし、その計算精度は、データライブラリを使った場合と比べてあまり検証されていませんので、[t-dchain] タリーを使う場合は、イベントジェネレータモードをオフ (e-mode=0) として計算することを奨励いたします。また、最初から放射化している核種を [material] セクションで設定しても、その放射能の寄与は DCHAIN-SP 計算に含まれないのでご注意下さい。

[37] 甲斐哲也 他, “DCHAIN-SP 2001: 高エネルギー粒子誘導放射能計算コード”, JAEA-Data/Code-2001-016 (2001).

表 118: [t-dchain] パラメータ (1)

name	値	説明
mesh =	reg	形状メッシュ（reg のみ対応）。 形状メッシュサブセクションが必要。 （reg=セル番号）
file =	file name	DCHAIN-SP のインプットファイル名。1 つだけ定義する。 拡張子は任意のものが使用できるが、 .dtrk, .dyld, .dout は使用不可。
title =	（省略可）	出力ファイルヘッダーのタイトル。
timeevo = (次行)	データ数 時間 相対強度	照射・冷却時間ステップの数。 照射時間または冷却時間、相対照射強度 (timeevo 個)。 ●照射・冷却時間ステップは、計算開始時からの通算時間ではなく、各照射・冷却ステップ毎の時間。時間の単位は、秒 (s)、分 (m)、時 (h)、日 (d)、年 (y) であり、数値と単位の間には 1 つ以上の空白が必要。 例：5 秒 → 5 s, 22 日 → 22.0 d, 8.3 年 → 8.3e+0 y ●相対照射強度 線源の基準強度 (amp) に対する相対強度で、照射時間ステップ毎に変化させることができる。 例：基準強度と同じ場合は 1.0、冷却時間ステップの場合は 0.0 とする。 ※ List 6.1 [t-dchain] タリーの入力例を参照
outtime = (次行)	データ数 時間	計算結果出力時間の数 計算結果出力時間（outtime 個）。 ●正の値を指定した場合は計算開始時からの通算時間で、負の値を指定した場合は最後の照射時間ステップ終了後からの時間になる。時間の単位は、秒 (s)、分 (m)、時 (h)、日 (d)、年 (y) であり、数値と単位の間には 1 つ以上の空白が必要。照射中と冷却中の任意の時間を指定でき、照射・冷却時間ステップの区切り時間と一致する必要はない。 ただし、timeevo で指定した値を超えた時間を指定することはできない。※ List 6.1 [t-dchain] タリーの入力例を参照 また、放射能が 0 のとき（例えば"0.0 m"など）の結果を出力するためには DCHAIN-SP パラメータ iprtb2 を 0 としてください（"iprtb2 = 0"と [t-dchain] セクション内に書けば設定できます）。
amp =	（省略可, D=1.0）	線源の基準強度 (source/秒)。

表 119: [t-dchain] パラメータ (2)

name	値	説明
target =	（省略可，D=0）	対象領域の構成物質（核種及び密度）の設定 なし：0，次行以降の記載不要 あり：1，次行以降の記載必要 ※構成物質に関するパラメータは、入力ファイルの [Material]，[Cell]，[Volume] セクションから自動的に設定されるので、通常は記述の省略が可能。 ※ [Material] で定義した核種以外を DCHAIN-SP で計算したい場合や [Volume] で体積を定義しない場合は、次行以降に target=1 として対象領域に対する通し番号、cell 番号、体積の順番で入力し、tg-list サブセクションが必要。（List 6.2 をご参照ください。）

これら以外にも、DCHAIN-SP のインプットに書き込むパラメータ

imode, jmode, idivs, iregon, inmtcf, ichain, itdecs, itdecn, isomtr, ifisyd, ifisye, iyild, iggrp, ibetap, acmin, istabl, igsdef, iprtb1, iprtb2, rprtb2, iprtb3, igsorg, ebeam, prodnp

を [t-dchain] セクションで直接指定することが可能です。各パラメータの意味は、DCHAIN-SP のマニュアルをご参照下さい。

List 6.1 ● [t-dchain] タリーの入力例

```
 1:        mesh = reg                      <-region メッシュの指定
 2:         reg = 100                      <-計算領域の指定
 3:        file = testDC.spd               <-DCHAIN-SP 入力ファイル名の指定
 4:       title = [t-dchain] test calc.
 5:         amp = 1.0E12                   <-線源の基準強度 (source/秒)
 6:
 7:     timeevo = 4                        <-照射・冷却時間ステップの数
 8:              3.0 h  1.0                <-3 時間運転
 9:              2.0 h  0.0                <-2 時間運転停止
10:              3.5 h  1.0                <-3.5 時間運転
11:             15.5 h  0.0                <-15.5 時間運転停止
12:
13:     outtime = 3                        <-計算結果出力時間の数
14:              3.0 h                     <-運転開始から 3.0 時間後
15:             -1.0 h                     <-最後の照射終了時間から 1 時間後
16:             -3.0 h                     <-最後の照射終了時間から 3 時間後
```

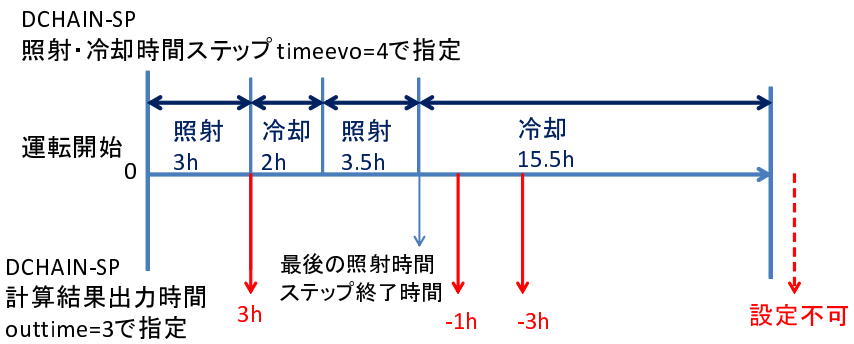


図 35: List 6.1 の例題における照射・冷却時間ステップと計算結果出力時間の関係。

List 6.2 ● [t-dchain] のパラメータ target=1 の場合の入力例

```
            .......
:      target = 1                          <-対象領域の構成物質（核種及び密度）の設定有
:      non    reg    vol                   <-省略可
:       1     1   8000.0                   <-通し番号、cell 番号、体積 (cm^3)
:     tg-list = 2                          <-対象領域構成物質に含まれる核種の数
:             H-1   6.689E-02              <-核種の元素記号と原子数密度 (10^24 個/cm^3)
:             O-16  3.345E-02
:       2     2   2000.0                   <-通し番号、cell 番号、体積 (cm^3)
:     tg-list = 1                          <-対象領域構成物質に含まれる核種の数
:            Fe-56  8.385E-02              <-核種の元素記号と原子数密度 (10^24 個/cm^3)
```

他に必要な PHITS 入力ファイルの設定及び注意事項は次のとおりです。

- [t-dchain] は、*PHITS* の入力ファイル 1 個につき、1 つしか設定できません。
- [Parameters] セクションにおいて次のパラメータの設定を行う必要があります。
  - jmout=1: 物質の原子数密度を表示させます。
  - file(21): DCHAIN-SP 用データフォルダの場所を指定します。
- [Volume] セクションにおいて各領域の体積を指定します。*PHITS* には reg メッシュの体積を自動で計算する機能がないため、この入力が必須となります。

このセクションが作成するファイルで、DCHAIN-SP 実行に必要なファイルを以下に示します。

- DCHAIN-SP の基本入力ファイル：file で指定したファイル。
- 20MeV 以下の中性子スペクトルファイル：対象領域の数に応じた “n.flux”（20MeV 以下の 175 群中性子スペクトル）。ファイル名に、mesh = reg による対象領域の指定順に 01 から始まる枝番号が振られます。例：n.flux_01, n.flux_02, $\cdots$, n.flux_XX
- 核種生成率ファイル：“nmtc_yield”
- DCHAIN-SP 用データ格納フォルダのリンク名が書かれたファイル：“dch_link.dat”

また、DCHAIN-SP 実行時には必要ないのですが、*PHITS* の再開始計算等に必要なファイルが 4 つできます。これらは拡張子により区別され、***.dtrk, ***.dyld, ***_err.dyld, ***.dout となります。ここで、***は file=で指定した、拡張子を除く名前を示します。

6.15 [T-WWG] セクション

[t-wwg] は、[Weight Window] セクションのパラメータを出力するタリーです。想定している 3 次元体系に対して、Weight Window 機能が効果的に動作するパラメータを自動で得ることができる Weight Window Generator (WWG) です。基本的には、まず [t-wwg] を設定したインプットで仮計算を行い、次に、得られた [Weight Window] のパラメータを加えたインプットで本計算を行う、という流れで本タリーを利用します。出力する [Weight Window] のパラメータは、仮計算で得られた各セルに飛んできた粒子の数[38] に基づいて決定しています。仮計算において、[t-wwg] で指定したセルに飛んできた粒子の数をタリーしており、その中で最も大きな値が 1 となるよう全ての値を規格化し、[Weight Window] のパラメータへと変換します。その際、一つも粒子が飛んで来なかったセルについては、最も数の少なかったセルと同じ Weight Window を設定します。パラメータは、part で指定した粒子毎に 1 セット出力します。もし、エネルギーメッシュや時間メッシュが定義されている場合は、それらのビン毎にパラメータのセットを書き出します。

[t-wwg] の入力書式は、以下の通りです。

表 120: [t-wwg] パラメータ (1)

name	値	説明
mesh =	reg	形状メッシュ。reg だけ指定可。
part =	all(省略時), 粒子名	ひとつの [t-wwg] セクション最大 6 個まで
material =	(省略可) all, 数 (次行)	スコアする material を限定する。複数定義可 all: デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの material を 次の行に記述する。 material 番号。
e-type =	1, 2, 3, 4, 5	エネルギーメッシュ。 エネルギーメッシュサブセクションが必要。
t-type =	1, 2, 3, 4, 5 (省略可)	時間メッシュ。エネルギーメッシュと併用不可。 時間メッシュサブセクションが必要
unit =	1(省略可)	1: [1/cm^2/source] 1 のみ指定可。
axis =	eng, reg, xy, yz, xz, t, wwg	出力データの x 軸。wwg 以外は [t-track] と同じ。 再開始計算が必要な場合は、2 つ以上の axis を定義し、 最初の 1 つを reg, eng, t のどれかで指定する。
file =	file name	axis の数だけ定義する。
resfile =	(省略可, D=file)	再開始計算時の過去タリーファイル名。複数 axis の 場合でも 1 つのみ指定。
multiplier =	物質数 (省略可)	multiplier を物質毎に指定します。
factor =	(省略可、D=1.0)	normalization factor
title =	(省略可)	出力ファイルヘッダーのタイトル。
angel =	(省略可)	angel パラメータ
x-txt = y-txt = z-txt =	(省略可) (省略可) (省略可)	x-軸テキスト y-軸テキスト z-軸テキスト

[38] 正確には，モンテカルロ粒子フルエンス，すなわち輸送粒子のウェイトを常に 1 にした粒子フルエンスに基づいて決定します

表 121: [t-wwg] パラメータ (2)

name	値	説明
rshow =	0(省略時), 1, 2, 3	axis=xy,yz,xz の時、領域境界 (1)、物質番号 (2)、 領域番号 (3) を表示。 この下に xyz 形状メッシュセクションが必要。
ginfo =	0(省略時), 1, 2	rshow 時の領域チェックのオプション。
resol = width =	1(省略時) 0.5(省略時)	rshow 時の表示オプション。 rshow 時の表示オプション。
volume	(省略可)	各領域の体積を定義します。この行の下に volume 定義文が 必要です。省略された場合、[volume] セクションで定義された volume を使用します。
epsout =	0(省略時), 1	1 で出力ファイルを ANGEL で処理した eps ファイル を作成。ファイル名は出力ファイルの拡張子を eps に 変えたファイル名。
ctmin(i) = ctmax(i) =	(省略可、D=-9999) (省略可、D= 9999)	i-th カウンターの最小値。 i-th カウンターの最大値。
trcl =	(省略可)	rshow での座標変換。
gslat =	1(省略時), 0	rshow で lattice の境界を表示するオプション。

このタリーでは、mesh は reg だけが指定できます。これは、[Weight Window] セクションのパラメータが reg 毎に指定する形式であるためです。出力の axis には、基本的に、[Weight Window] セクションのパラメータを出力する wwg を指定してください。この他に、eng, reg, xy, yz, xz, t も指定できますが、これらは [T-Track] で指定した場合と同じ結果しか得られず、[Weight Window] に関係する値は得られません。ここで、xy, yz, xz の 2 次元表示は rshow=1 を指定したときのみ有効です。

wwg の出力例を以下に示します。

```
[ Weight Window ]
set:  c71[0.0]  c72[c71+2.13496E-08]  c73[1.25478E-04]
set:  c74[0.0]  c75[c74+9.29569E-09]  c76[2.08987E-04]

    part = neutron
     eng =  2
          1.00000E-03   1.00000E+03

    reg          ww1                    ww2
    1   (8.94092E-05+c71)/c73  (2.08987E-04+c74)/c76
    2   (1.25478E-04+c71)/c73  (1.26817E-04+c74)/c76
    3   (8.53835E-05+c71)/c73  (5.78131E-05+c74)/c76
```

ここでは、中性子に対して、2 群のエネルギービンの Weight Window のパラメータが定義されています。パラメータの定義には c**で表される定数が使われています。この場合，c71-c73 が第 1 エネルギー群（E <1keV）に対するパラメータ c74-76 が第 2 エネルギー群（1keV< E <1GeV）に対するパラメータです。

これらのパラメータの中でユーザー自身が変更する可能性があるパラメータは，各群に対して左端で定義される初期設定値が 0 のパラメータ c71 や c74 などです。これらのパラメータは，各群の Weight Window にある一定値を加える際，利用します。

6.16 [T-Userdefined] セクション

ユーザー定義タリーは、通常のタリーでは得ることができない様々な物理量をタリーする機能です。各ユーザーがファイル usrtally.f を Fortran 言語を用いて変更し、求める物理量を書き出します。このため、ソースファイルのコンパイルが必要となります。

インプットファイルで [t-userdefined] セクションを定義することにより、*PHITS* 実行時に subroutine usrtally を通過するようになります。通過のタイミングは dumpall の書き出しと同じで、あらゆるイベントが発生した時です。本タリーの基本的な考え方として、subroutine usrtally で示されている変数を if 文等の条件で選別してカウントすることで、求める物理量を引き出します。この subroutine で扱うことができる変数を以下で説明しますが、中には値を変更すると *PHITS* の計算がおかしくなる可能性があるので、十分に注意してください。本タリーの具体的な利用方法については、\phits\utility\usrtally フォルダにある readme やサンプルインプットをご参照ください。

[t-userdefined] セクションで定義できる変数は、書き出し用ファイル名を与える file と subroutine usrtally で使用できるパラメータ udtpara$i$ ($i = 0 - 9$) の 2 種類です。これらは再コンパイルせずに変更できる変数となります。

表 122: [t-userdefined] パラメータ

name	値	説明
file =	file name	subroutine usrtally 内で使用できる書き出し用ファイル。 最大 50 まで定義できる (装置番号=151-200)。
udtpara$i$ = ($i = 0 - 9$)	(省略可)	subroutine usrtally 内で使用できるパラメータ。 udtpara0 から udtpara9 まで最大 10 定義できる。

List 6.3 ● [T-Userdefined] パラメータの入力例。

```
1:        file = output1.dat       <-装置番号は 151 で、subroutine usrtally 内では iudtf(1)
2:        file = output2.dat       <-装置番号は 152 で、subroutine usrtally 内では iudtf(2)
3:    udtpara0 = 10.0              <-subroutine usrtally 内では udtpara(0)
4:    udtpara3 = 20.0              <-subroutine usrtally 内では udtpara(3)
```

以下が `subroutine usrtally` 内で示されている変数となります。

(1) NCOL:
これはプログラムの中の変数で、粒子輸送のタイミングの認識変数です。

NCOL

1	:	start of calculation
2	:	end of calculation
3	:	end of a batch
4	:	source
5	:	detection of geometry error
6	:	recovery of geometry error
7	:	termination by geometry error
8	:	termination by weight cut-off
9	:	termination by time cut-off
10	:	geometry boundary crossing
11	:	termination by energy cut-off
12	:	termination by escape or leakage
13	:	(n,x) reaction
14	:	(n,n'x) reaction
15	:	sequential transport only for tally

(2) npe, me:
それぞれメモリ分散型並列計算時の並列 PE(Processor Element) 数と各 Element の識別番号。

(3) ipomp,npomp:
それぞれメモリ共有型並列計算時のスレッド識別番号と使用している全スレッド数。

(4) iusrtally:
`subroutine usrtally` への通過をコントロールするパラメータ。インプットファイルに `[T-Userdefined]` セクションがあれば 1 となっています。

(5) iudtf(50):
`file` で定義したファイルの装置番号 (151-200)。

(6) udtpara(0:9):
`udtpara`i $(i = 0 - 9)$ で定義した値。

(7) NOCAS, NOBCH, RCASC, RSOUIN:

NOCAS	:	current event number in this batch
NOBCH	:	current batch number
RCASC	:	real number of NOCAS+maxcas*(NOBCH-1)
RSOUIN	:	sum of the weight of source particle

(8) NO, IDMN, ITYP, KTYP, JTYP, MTYP, RTYP, OLDWT:

NO	:	cascade id in this event
IDMN	:	material id
ITYP	:	particle type
KTYP	:	particle kf-code
JTYP	:	charge number of the particle
MTYP	:	baryon number of the particle
RTYP	:	rest mass of the particle (MeV)
OLDWT	:	wight of the particle at (x,y,z)

(9) QS:
電子の阻止能 (dE/dx)。

(10) IBLZ1, IBLZ2, ILEV1, ILEV2:

IBLZ1	:	cell id at (x,y,z)
IBLZ2	:	cell id after crossing
ILEV1	:	level structure id of the cell at (x,y,z)
ILEV2	:	level structure id of the cell after crossing

(a) ILAT1:
階層構造をもつ CELL の変数です。

(b) ILAT2:
階層構造をもつ CELL の変数です。

(11) COSTH, UANG(1), UANG(2), UANG(3), NSURF:

COSTH	:	cosine of an angle of incidence in a surface crossing
UANG(1,2,3)	:	x,y,z component of a normal vector of its surface, respectively
NSURF	:	internal number of the surface Note that this is different from the surface number defined in the [surface] section

(12) NAME, NCNT(1), NCNT(2), NCNT(3):

NAME	:	collision number of the particle
NCNT(1,2,3)	:	values of counter 1, 2, and 3

(13) WT, U, V, W:

WT	:	wight of the particle at (xc,yc,zc)
U, V, W	:	unit vector of momentum of the particle

(14) E, T, X, Y, Z:

E	:	energy of the particle at (x,y,z) (MeV)
T	:	time of the particle at (x,y,z) (nsec)
X, Y, Z	:	position coordinate of the preceding event point (cm)

(15) EC, TC, XC, YC, ZC:

EC	:	energy of the particle at (xc,yc,zc) (MeV)
TC	:	time of the particle at (xc,yc,zc) (nsec)
XC, YC, ZC	:	position coordinate of the particle (cm)

(16) SPX, SPY, SPZ:

SPX, SPY, SPZ	:	unit vector of spin direction of the particle

(17) NZST:
粒子がもつ電荷 (charge state)。

(18) NCLSTS:
意味は、衝突もしくは崩壊による生成粒子の個数です。原子核も含みます。

(a) MATHZ, MATHN, JCOLL, KCOLL:

MATHZ	:	Z number of the mother nucleus
MATHN	:	N number of the mother nucleus
JCOLL	:	reaction type id1
KCOLL	:	reaction type id2

JCOLL, KCOLL はそれぞれ次のような意味を持ちます。

JCOLL

0	:	nothing happen
1	:	Hydrogen collisions
2	:	Particle Decays
3	:	Elastic collisions
4	:	High Energy Nuclear collisions
5	:	Heavy Ion reactions
6	:	Neutron reactions by data
7	:	Photon reactions by data
8	:	Electron reactions by data
9	:	Proton reactions by data
10	:	Neutron event mode
11	:	delta ray production
13	:	Photon reactions by EGS5
14	:	Electron reactions by EGS5

KCOLL

0	:	normal
1	:	high energy fission
2	:	high energy absorption
3	:	low energy n elastic
4	:	low energy n non-elastic
5	:	low energy n fission
6	:	low energy n absorption

(b) ICLUSTS, JCLUSTS, QCLUSTS, JCOUNT:
これらの変数は、各生成粒子の情報を記述するもので、NCLSTS だけ繰り返し、また、それぞれの変数も以下のように配列を持ちます。

ICLUSTS kind of particle

0	:	nucleus
1	:	proton
2	:	neutron
3	:	pion
4	:	photon
5	:	kaon
6	:	muon
7	:	others

JCLUSTS(i)

i = 0	:	angular momentum
= 1	:	proton number
= 2	:	neutron number
= 3	:	ityp
= 4	:	status of the particle 0: real, <0 : dead
= 5	:	charge number
= 6	:	baryon number
= 7	:	kf code

QCLUSTS(i)

i = 0	:	impact parameter
= 1	:	px (GeV/c)
= 2	:	py (GeV/c)
= 3	:	pz (GeV/c)
= 4	:	$e_{tot} = \sqrt{p^2 + m^2}$ (GeV)
= 5	:	rest mass (GeV)
= 6	:	excitation energy (MeV)
= 7	:	kinetic energy (MeV)
= 8	:	weight
= 9	:	time (nsec)
= 10	:	x coordinate (cm)
= 11	:	y coordinate (cm)
= 12	:	z coordinate (cm)

6.17 [T-Gshow] セクション

xyz メッシュで指定する面の領域境界を表示します。この機能は、パラメーターセクションで icntl = 7 を指定すれば、輸送計算を行うことなく、出力させることができます。

表 123: [t-gshow] パラメータ

name	値	説明
mesh =	xyz	形状メッシュ xyz のみ、形状メッシュサブセクションが必要
axis =	xy, yz, xz	2 次元表示
file =	file name	axis の数だけ定義する
output =	1, 2 3, 4 5, 6 7, 8	1: 境界線、2: 境界線+物質色 3: 境界線+物質番号、4: 境界線+物質色+物質番号 5: 境界線+領域番号、6: 境界線+物質色+領域番号 7: 境界線+LAT 番号、8: 境界線+物質色+LAT 番号
resol = width =	1(省略時) 0.5(省略時)	領域境界を求める分解能を各辺 resol 倍します。 領域境界を表示する線の太さを定義します。
title =	(省略可)	出力ファイルヘッダーのタイトル
angel =	(省略可)	angel パラメータ
x-txt = y-txt =	(省略可) (省略可)	x-軸テキスト y-軸テキスト
epsout =	0(省略時), 1	1 で出力ファイルを $A_N G_E L$ で処理した eps ファイルを作成。ファイル名は出力ファイルの拡張子を eps に変えたファイル名
bmpout =	0(省略時), 1	二次元ビットマップ画像ファイルを出力する。 ファイル名は出力ファイルの拡張子を bmp に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkout =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルを出力する。 ファイル名は出力ファイルの拡張子を vtk に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
vtkfmt =	0(省略時), 1	タリー出力結果の三次元可視化用ファイルのフォーマット。 テキスト形式（0）またはバイナリ形式（1）を選択する。
trcl =	(省略可)	r-z, xyz メッシュの座標変換番号もしくは座標変換定義
gslat =	1(省略時), 0	0 で gshow で lattice の境界を非表示、1 で表示
ginfo =	2(省略時), 0 1 2	領域エラーチェックのオプション 領領域エラーチェックを行わない。 領域エラーチェックを行い 2 次元ジオメトリ図上にエラー領域を表示する。 エラー領域表示に加えて領域エラーファイル (.err) を出力。

output=7, 8 は、最下層のセルが Lattice それ自身の時のみ、 Lattice の成分番号を (4,1,2) の形式で出力します。例えば、4.6.4 節で示した図 24 は、次の様な [t-gshow] のインプットで出力しています。

List 6.4 ● [t-gshow] の例題

```
 1:   [ T - gshow ]
 2:        mesh = xyz
 3:      x-type = 2
 4:          nx = 100
 5:        xmin = -10.
 6:        xmax =  10
 7:      y-type = 1
 8:          ny = 1
 9:              -5.0 5.0
10:      z-type = 2
11:       nz    = 100
12:       zmin  = -10.
13:       zmax  =  10.
14:        axis = xz
15:      output = 8
16:        file = cell-example6.dat
17:      epsout = 1
```

6.18 [T-Rshow] セクション

xyz メッシュで指定する面の領域境界と、与えられた領域毎の物理量に比例した色でその領域を塗ります。通常は、reg メッシュで得られた物理量を value 定義文に書きこみ xyz メッシュで指定する面に領域の色として出力することを目的とします。従って、この機能は、パラメーターセクションで icntl = 9 を指定した時だけ 出力させることができます。

物理量に対して、色の変化を線形でするか、対数でするかは、A_NG_EL パラメーターで、zlog, zlin で指定して下さい。デフォルトは、zlin です。

表 124: [t-rshow] パラメータ

name	値	説明
mesh =	xyz	形状メッシュ xyz のみ、形状メッシュサブセクションが必要
axis =	xy, yz, xz	2 次元表示
file =	file name	axis の数だけ定義する
output =	1, 2, 3	1: 境界線、2: 境界線+物質番号、 3: 境界線+領域番号
resol = width =	1(省略時) 0.5(省略時)	領域境界を求める分解能を各辺 resol 倍します。 領域境界を表示する線の太さを定義します。
title =	(省略可)	出力ファイルヘッダーのタイトル
angel =	(省略可)	angel パラメータ
x-txt = y-txt = z-txt =	(省略可) (省略可) (省略可)	x-軸テキスト y-軸テキスト z-軸テキスト
reg = value reg val		領域定義 各領域の物理量を定義します。 この行の下に value 定義文が必要。 value 定義文。書式は volume 定義文と同じ。5.1.2 を参照。
iechrl =	72(省略時)	value のインプットエコーの時の最大コラム数
epsout =	0(省略時), 1	1 で出力ファイルを A_NG_EL で処理した eps ファイル を作成。ファイル名は出力ファイルの拡張子を eps に 変えたファイル名
bmpout =	0(省略時), 1	二次元ビットマップ画像ファイルを出力する。 ファイル名は出力ファイルの拡張子を bmp に変えたファイル名。 mesh=xyz, axis=(xy, yz, xz のいずれか) のときのみ有効。
trcl =	(省略可)	r-z, xyz メッシュの座標変換番号もしくは座標変換定義
gslat =	1(省略時), 0	0 で gshow で lattice の境界を非表示、1 で表示
ginfo =	2(省略時), 0 1 2	領域エラーチェックのオプション 領領域エラーチェックを行わない。 領域エラーチェックを行い 2 次元ジオメトリ図上にエラー領域 を表示する。 エラー領域表示に加えて領域エラーファイル (.err) を出力。

例えば、4.6.4 節の図 24 に色を付ける場合、次の例題で示すような [t-rshow] タリーを用いると、図 36 の結果を得ることができます。

List 6.5 ● [t-rshow] の例題

```
[ T - rshow ]
      mesh = xyz
    x-type = 2
        nx = 100
      xmin = -10.
      xmax =  10.
    y-type = 1
        ny = 1
            -5.0  5.0
    z-type = 2
        nz = 100
      zmin = -10.
      zmax =  10.
      axis = xz
    output = 1
      file = cell-example6-rshow.dat
    epsout = 1
       reg = (201<101[-1 1 0]<1) (201<101[0 1 0]<1) (201<101[1 1 0]<1)
             (201<101[-1 0 0]<1) (201<101[0 0 0]<1) (201<101[1 0 0]<1)
             (201<101[-1 -1 0]<1) (201<101[0 -1 0]<1) (201<101[1 -1 0]<1)
     value
    non     reg      val     # reg definition
     1  1000001   1.0000E+00 # ( 201 < 101[ -1 1 0 ] < 1 )
     2  1000002   2.0000E+00 # ( 201 < 101[ 0 1 0 ] < 1 )
     3  1000003   3.0000E+00 # ( 201 < 101[ 1 1 0 ] < 1 )
     4  1000004   4.0000E+00 # ( 201 < 101[ -1 0 0 ] < 1 )
     5  1000005   5.0000E+00 # ( 201 < 101[ 0 0 0 ] < 1 )
     6  1000006   6.0000E+00 # ( 201 < 101[ 1 0 0 ] < 1 )
     7  1000007   7.0000E+00 # ( 201 < 101[ -1 -1 0 ] < 1 )
     8  1000008   8.0000E+00 # ( 201 < 101[ 0 -1 0 ] < 1 )
     9  1000009   9.0000E+00 # ( 201 < 101[ 1 -1 0 ] < 1 )
```

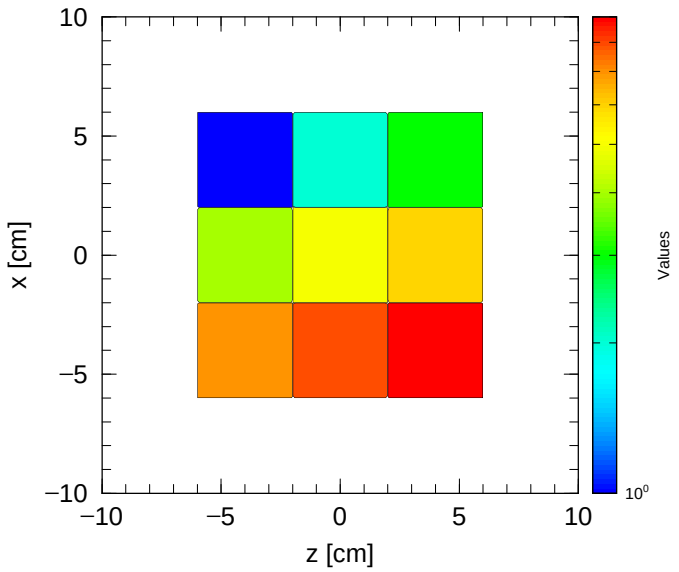


図 36: [t-rshow] の例題を用いて図 24 の各領域に色をつけた結果。

6.19 [T-3Dshow] セクション

幾何形状の 3 次元パース図を出力します。このタリーは、パラメーターセクションで icntl = 11 を指定した時だけ出力させることができます。

表 125: [t-3dshow] パラメータ (1)

name	値	説明
output =	0, 1, 2, 3 (省略時 3)	0: ドラフト、 1: 境界線のみ、 2: 境界線無し、 3: 境界線+カラー
material =	(省略可) all, 数	表示する material を限定する。複数定義可。 all : デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの material を次の行に記述する。 負の数にした場合は、それらの material を表示しないことを意味する。
(次行)	2 5 8	material 番号
x0 = y0 = z0 =	(D=0.0) (D=0.0) (D=0.0)	視線、光線の原点の座標。 この点と視点を結んだ線が画面の中央になる。 視点、光源の座標は、この点を原点とする。
e-the = e-phi = e-dst =	(D=80) (D=140) (D=w-dst*10)	視点の z 方向からの角度 θ(degree) 視点の x 方向からの方位角 ϕ(degree) 視点の視点原点からの距離 (cm)
l-the = l-phi = l-dst =	(D=e-the) (D=e-phi) (D=e-dst)	光源の z 方向からの角度 θ(degree) 光源の x 方向からの方位角 ϕ(degree) 光源の光線原点からの距離 (cm)
w-wdt = w-hgt = w-dst = w-mnw = w-mnh = w-ang =	(D=100) (D=100) (D=200) (D=100) (D=100) (D=0.0)	画面フレームの幅 (cm) 画面フレームの高さ (cm) 画面フレームの視点原点からの距離 (cm)。画面フレームの中央と視点原点を結ぶ直線は、画面フレーム平面に垂直で、かつ視点を通る。 画面横方向のメッシュ数 画面高さ方向のメッシュ数 画面の回転 (degree)
heaven =	(D=y)	画面上方向の向き x, -x, y, -y, z, -z で指定
mirror =	(D=0)	=-1 で画面の左右鏡面変換

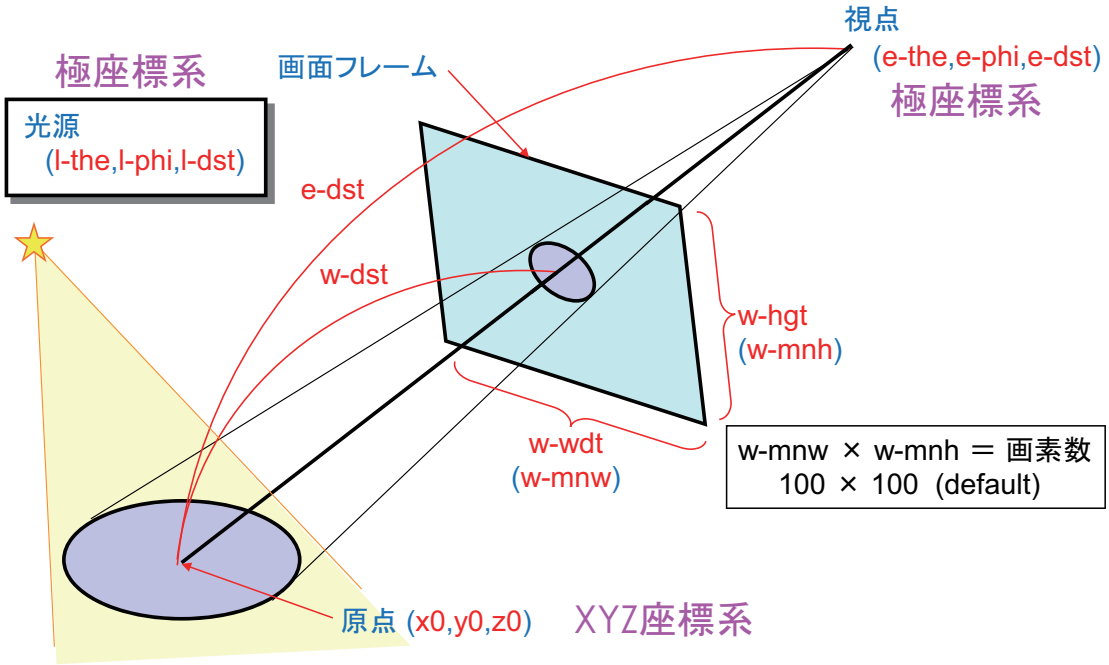


図 37: 3dshow タリー: 原点 (`x0,y0,z0`)、視点 (`e-the,e-phi,e-dst`)、光源 (`l-the,l-phi,l-dst`) 及び画面フレーム (`w-wdt,w-hgt,w-dst`) の与え方。

表 126: [t-3dshow] パラメータ (2)

name	値	説明
line =	(D=0) 0, 1	output = 1, 3 の時、0: 物質境界 + 面境界、 1: 物質境界 + 面境界 + 領域境界 を描く。
r-out =	(D=50000)	視点、光源を含む外部ボイドの半径 (cm)
shadow =	(D=0)	影の強さ、0 で影無し。2 ぐらいが適当。
bright = dark =	(D=0.8) (D=0.2)	面の明るさの制限、1 で最大、0 で明るさ無し 面の暗さの制限、1 で暗さ無し、0 で最大
box = box	(D=0) 10 個の数字	透過 box の数、最大 5 個まで。 透過 box の数だけ繰り返す
matinbox = (次行)	(省略可) all, 数 2 5 8	box 内で表示する material を指定する。複数定義可。 all : デフォルト、この場合省略した場合と同じ。 数を指定した場合、その数だけの material を次の行に記述する。 負の数は不可。 material 番号
reginbox =	(省略可) all, 領域番号	box 内で表示する領域を指定する。複数定義可。 all : デフォルト、この場合省略した場合と同じ。 matinbox の指定のある物質では、非表示にする
resol = width =	1(省略時) 0.5(省略時)	境界を求める分解能を各辺 resol 倍します。 域境界を表示する線の太さを定義します。
file =	file name	axis の数だけ定義する
title =	(省略可)	出力ファイルヘッダーのタイトル
angel =	(省略可)	angel パラメータ
x-txt = y-txt = z-txt =	(省略可) (省略可) (省略可)	x-軸テキスト y-軸テキスト z-軸テキスト
epsout =	0(省略時), 1	1 で出力ファイルを $A_N G_E L$ で処理した eps ファイルを作成。ファイル名は出力ファイルの拡張子を eps に変えたファイル名
axishow =	(D=1) 0, 1, 2	0:xyz 軸を表示しない。 1:xyz 軸を左下に表示する。 2:xyz 軸を中心に表示する。

reg = また、reginbox = の表式は、サブセクション 5.1.1 の領域メッシュの指定の仕方と同じです。

計算時間短縮のため、光源、視点が外部ボイドの内側に納まるように、半径 r-out の球を新たに設定してその外側を外部ボイド、その内側を内部ボイドに設定しています。r-out のデフォルトの値が小さい場合は、つまり、幾何形状がこの値より大きい場合、もしくは、光源、視点を非常に遠くに置きたい時など、r-out の値を大きく設定して下さい。この新しく加えられた外部ボイドを定義する球の設定は、インプットエコーで見ることができます。従って、この icntl = 11 の計算のインプットエコーの出力を再び計算のインプットに用いますとエラーします。視点と光源を一致させれば、影はできません。

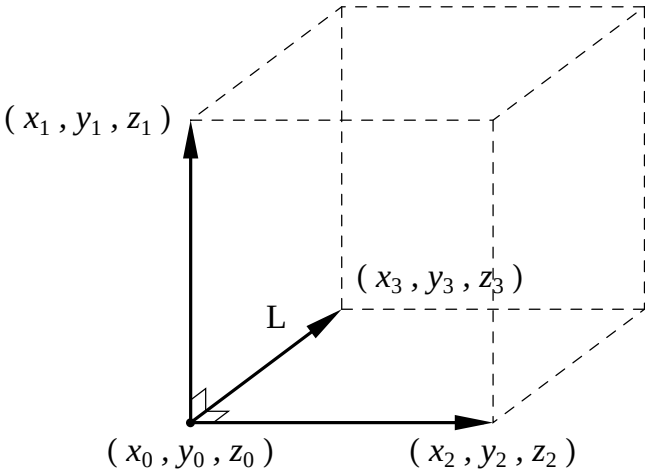
6.19.1 box の定義

透過 box は、最大 5 個まで定義できます。これらの box で定義された内部は、光は透過します。box の定義は、空間内の 3 点、$\mathbf{b}_0(x_0,y_0,z_0)$, $\mathbf{b}_1(x_1,y_1,z_1)$, $\mathbf{b}_2(x_2,y_2,z_2)$ を与えて、この 3 点で定義される平面に垂直な方向、つまり、$(\mathbf{b}_1 - \mathbf{b}_0) \times (\mathbf{b}_2 - \mathbf{b}_0)$ の方向に点 $\mathbf{b}_3$ を点 $\mathbf{b}_0$ から長さ L のところに取ります。この 4 点で下図のように box を定義します。この box の定義には、座標変換が使えます。box の各点を定義する前に trcl=で座標変換番号を定義するか、trcl=(•••••••••) のように座標変換を直接定義します。

ただし、透過 box で定義した空間に真空 (void) があると適切に機能しない場合があるようです。そのときは、密度を薄くした空気等の物質でその領域を満たしてください。

box の定義の書式は、

```
box = 2
box   x0  y0  z0
      x1  y1  z1
      x2  y2  z2  L
box   trcl = 2
      x0  y0  z0
      x1  y1  z1
      x2  y2  z2  L
box   *trcl = (0 0 0  0 90 90  90 60 150  90 30 60  -1)
      0.0  0.0  0.0
     -5.0  0.0  0.0
      0.0  0.0  5.0   5.0
```



6.19.2 3dshow の例題

3dshow の例題を見てみましょう。以下の例題の幾何形状は、以下のようなものです。

List 6.6 ● [t-3dshow] の例題

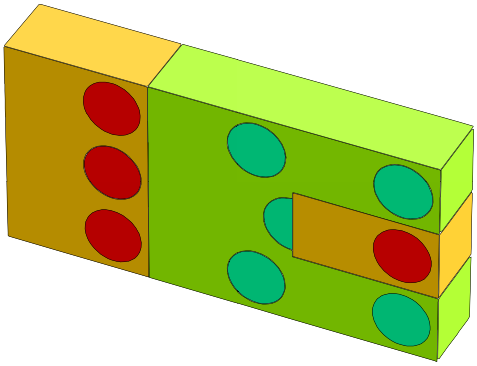
```
[cell]
    1  0 -1 fill=1
    2  0 -41 42 -43 44 -45 46  u=1 fill=5
   22  0 -41 42 -43 44 -45 46  u=1 trcl=(0 0 20) fill=6
   23  like 22 but trcl=(0 0 40) fill = 7
    5  0 -21 22 -23 24 -25 26  u=5 lat=1 fill=3
    6  0 -21 22 -23 24 -25 26  u=6 lat=1 fill= -1:1 0:0 0:0  2 2(0 0 5) 2
    7  0 -21 22 -23 24 -25 26  u=7  fill= -1:1 0:0 0:0  2 3 2 lat=1
    3  1  3.97300E-02  3 u=2
    4  6  4.18280E-02 -3 u=2
   13  5  8.47130E-04 -3 u=3
   14  3  1.23620E-01  3 u=3
   8  -1 +1
[surface]
    1   rpp -15 15 -5 5  -5 55
    21   px    5
    22   px   -5
    23   py    5
    24   py   -5
    25   pz   15
    26   pz   -5
    41   px   15
    42   px  -15
    43   py    5
    44   py   -5
    45   pz   15
    46   pz   -5
    5   rpp -20 20 -5 5 -5 35
    6   rpp -20 20 -5 5 -5 15
    7   rpp -20 20 -5 5 35 55
    3   c/y   0 10  4
```

全体は直方体、中にやはり直方体の lattice とその中に円柱が入っています。これを表示する 3dshow のインプットは次のようになります。

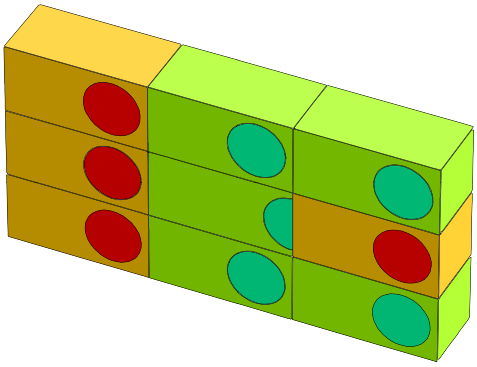
List 6.7 ● [t-3dshow] の例題

```
[t-3dshow]
   output = 3
   heaven = x
   resol = 2
   width = 0.1
   x0 =  0
   y0 =  0
   z0 = 25
   e-the = 70
   e-phi = 50
   e-dst = 1000
   l-the = 50
   l-phi = 25
   l-dst = 2000
   w-wdt = 60
   w-hgt = 40
   w-dst = 150
   file = dshow.dat
```

この結果は、



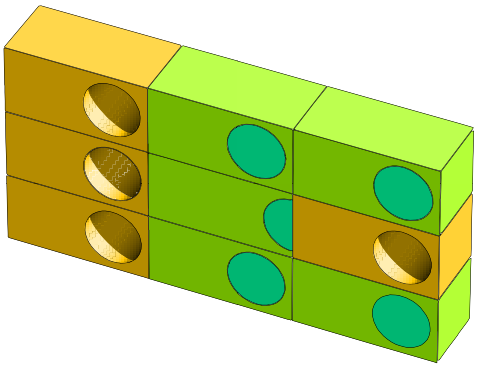
となります。これに、`line = 1` を加えて、領域境界の線も書き込んでみます。



lattice がどう組まれているかが分かると思います。次に、赤の領域、物質 5 番を透明にするために、また、影を付けるために、

```
material = -1
           5
  shadow =  2
```

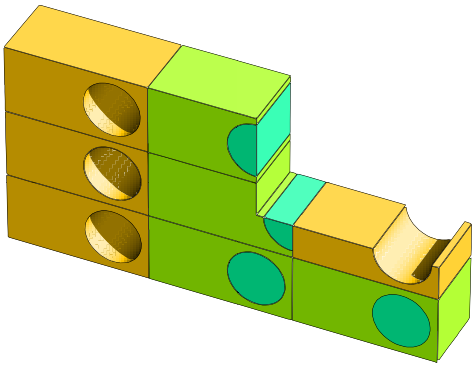
を加えます。



次に、box を設定します。

```
box = 1
box    0  10  30
     100  10  30
       0  10  100 100
```

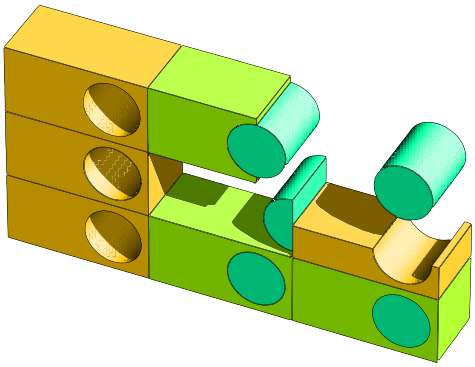
box により、一部が透明になり内部が見られます。



次に、

```
reg = ( 3 < 6[0 0 0] )
matinbox = 1
           6
```

を加え、box 外では、`reg = ( 3 < 6[0 0 0] )` を透明にし、box 内では、物質 6 番を見える様にします。



これらの組み合わせで、複雑な構造の内部を希望どおり表示することが可能になります。

7 タリーを用いた体積、面積計算

reg メッシュを用いたタリーにおいて、各領域の体積やある横断面の面積の値が必要になります。本説ではそれらの値をモンテカルロ積分により求める方法についてご紹介します。

モンテカルロ計算で体積や面積を求めるには、空間的に均一な粒子軌跡を用意して、領域を通過する、もしくは横断面を通過する flux を計算することが必要です。そのためには、`s-type=9` の球殻線源において `r1=r2` とし、`dir=-all` とします。 その際、目的の領域や横断面が半径 `r1` の球に入るようにしてください。`dir=-all` は球殻から内向きに線源が発生する設定で、$\cos^2\theta$ のバイアスを掛けてあるため、より中心部分で統計が良くなるようになっています。また、`totfact` として πr_1^2 の値を入れてください。線源の発生する面積を入れることで適切に規格化が行われます。

体積を求める場合は `[t-track]` タリーを用います。`mesh=reg` とし、`unit=4` としてください。 対して、面積を求める場合は `[t-cross]` タリーを使います。`mesh=reg`、`unit=1` とし、`output=flux` としてください。

上記の線源とタリーを設定した上で、`icntl=5` とし輸送計算を実行します。詳細は、/phits/lecture/lec03 フォルダにある基礎実習 3 の講習会資料をご参照ください。

以下に `[source]` セクションや `[t-track]`,`[t-cross]` タリーの例題を示します。

List 7.1 ● 体積、面積計算線源例

```
[ S o u r c e ]
 s-type =   9
   proj =   proton
     e0 =   500.0
     x0 =   0.0
     y0 =   0.0
     z0 =   30.0
     r1 =   18
     r2 =   18
    dir =   -all
totfact = 18**2*pi
```

この例題では、中心が (0,0,30)、半径が 18cm の球面が設定されています。粒子種、エネルギーは任意です。目的の領域や横断面がこの球内に入る様に半径、中心位置を決めます。次に、体積、面積を求めるタリーの例を示します。

List 7.2 ● 体積を求めるタリー例

```
[ T - T r a c k ]
    mesh = reg
    reg = 1 2 3 4 5
  e-type = 2
    emin = 0.
    emax = 1000.0
      ne = 1
    axis = reg
    unit = 4
    file = volume.dat
```

求めたい領域を指定し、エネルギー範囲は線源のエネルギーを含む 1 群とします。unit=4 とします。unit=4 より、体積は自動的に 1 にセット されますから、volume セクションは特に必要ありません。この計算の結果得られた体積の値を、実際に必要なタリーの体積定義のところにコピーして使います。

また、`multiplier` を使えば、領域の質量（体積 × 密度）を直接計算することもできます。具体的には、上記 `[t-track]` セクションに

```
multiplier = all
      part = all
      emax = 1.0e10
mat mset1
all ( 1.0 -120 )
```

を加えます。ここで、multiplier 番号 −120 は、計算した飛跡長に密度を乗じるための関数です。詳しくは、6.1[t-track] セクションをご参照ください。

次は面積の例題です。

List 7.3 ● 面積を求めるタリー例

```
 1:   [ T - C r o s s ]
 2:        mesh = reg
 3:        reg = 3
 4:        r-in    r-out    area
 5:        ( 1 2 ) ( 1 2 )  1.0000E+00
 6:        ( 2 3 ) ( 2 3 )  1.0000E+00
 7:        ( 3 4 ) ( 3 4 )  1.0000E+00
 8:      e-type = 2
 9:        emin = 0.
10:        emax = 1000.0
11:          ne = 1
12:        axis = reg
13:        unit = 1
14:        file = area.dat
```

求めたい面を指定し、エネルギー範囲は線源のエネルギーを含む 1 群とします。unit=1 とします。この計算の結果得られた面積の値を、上の area の値にコピーして使います。

8 dump ファイルの処理

[t-cross], [t-time], [t-product] タリーで、粒子の情報をファイルにダンプすることができます。このダンプファイルをソースとして新たな継続計算ができます。また、ダンプファイルの情報を処理することにより、既存のタリーにはない物理量の集計を取ることができます。そのためには、ユーザーがダンプファイルを読み込んで処理するプログラムを書かなければなりません。そのプログラムの雛形として、バイナリーダンプファイルをアスキーへ、アスキーダンプファイルをバイナリーへ変換するプログラムを添付しました。プログラムは、dump-a.f 、Windows の実行ファイルが dump_a.exe です。

このプログラムを書き換えることで、ダンプファイルを処理していろいろな物理量を求めるユーザープログラムを作ることができます。その助けのために、ソースリストと解説を以下に示します。

List 8.1 ● dump-a.f のソース

```
 1:  ************************************************************************
 2:  *                                                                      *
 3:  *     This program exchanges the binary data and the ascii data        *
 4:  *     of dump file.                                                    *
 5:  *                                                                      *
 6:  *       modified by K.Niita on 2005/08/15                              *
 7:  *                                                                      *
 8:  *                                                                      *
 9:  *                                                                      *
10:  *                                                                      *
11:  ************************************************************************
12:        implicit real*8 (a-h,o-z)
13:  *----------------------------------------------------------------------
14:        dimension isdmp(0:30)
15:        dimension jsdmp(0:30)
16:        data isdmp / 31*0 /
17:        data jsdmp / 31*0 /
18:        character chin*80
19:        character chot*80
20:        logical exex
21:        character dmpc(30)*4
22:        data dmpc / '  kf','   x','   y','   z','   u','   v','   w',
23:       &            '   e','  wt','  tm','  c1','  c2','  c3',
24:       &            '  sx','  sy','  sz','  n0','  nc','  nb','  no',
25:       &            '    ','    ','    ','    ','    ','    ',
26:       &            '    ','    ','    ','    ','    '/
27:        dimension dmpd(30)
28:        dimension dmpp(30)
29:        data dmpp /  2112.,   0.0,   0.0,   0.0,   0.0,   0.0,   1.0,
30:       &              100.,   1.0,   0.0,   0.0,   0.0,   0.0,
31:       &              0.0,    0.0,   0.0,   0.0,   1.0,   1.0,   1.0,
32:       &              0.0,    0.0,   0.0,   0.0,   0.0,   0.0,   0.0,
33:       &              0.0,    0.0,   0.0/
34:  *----------------------------------------------------------------------
35:           in = 5
36:           io = 6
37:           id = 20
38:           ia = 21
39:           iserr = 0
40:  *----------------------------------------------------------------------
41:  *        user program frag : 0 => no,  1 => with user program
42:  *----------------------------------------------------------------------
43:           iuser = 0
44:  *----------------------------------------------------------------------
45:  *        read ascii or binary frag
46:  *----------------------------------------------------------------------
47:           write(io,*) ' ** 0 => read binary to ascii'
48:           write(io,*) ' ** 1 => read ascii  to binary'
49:           read(in,*,end=993) iasb
50:  *----------------------------------------------------------------------
51:  *        read the name of input dump file
52:  *----------------------------------------------------------------------
53:           write(io,*)
```

```
 54:          write(io,*) ' ** put the file name of input dump file'
 55:          read(in,'(a80)',end=998) chin
 56:             inquire( file = chin, exist = exex )
 57:             if( exex .eqv. .false. ) then
 58:               write(io,*) ' ** Error : the file does not exist'
 59:               goto 999
 60:             end if
 61:             if( iasb .eq. 0 ) then
 62:               open(id, file = chin,
 63:      &             form='unformatted',status = 'old' )
 64:             else
 65:               open(id, file = chin,
 66:      &             form='formatted',status = 'old' )
 67:             end if
 68:  *-----------------------------------------------------------------
 69:  *        read the number of data and data sequence
 70:  *-----------------------------------------------------------------
 71:          write(io,*)
 72:          write(io,*) ' ** put the number of data in a record'
 73:          read(in,*,end=997) isdmp(0)
 74:          write(io,*)
 75:          write(io,*) ' ** put the ID numbers of data in a record'
 76:          read(in,*,end=996) ( isdmp(i), i = 1, isdmp(0) )
 77:             do k = 1, isdmp(0)
 78:                if( isdmp(k) .gt. 20 .or.
 79:      &             isdmp(k) .le.  0 ) goto 992
 80:                  jsdmp(isdmp(k)) = k
 81:             end do
 82:             write(io,*)
 83:             write(io,'('' # dump data : '',30(a4))')
 84:      &         ( dmpc(isdmp(j)), j = 1, isdmp(0) )
 85:  *-----------------------------------------------------------------
 86:  *        read the name of output dump file
 87:  *-----------------------------------------------------------------
 88:          write(io,*)
 89:          write(io,*) ' ** put the file name of output'
 90:          read(in,'(a80)',end=998) chot
 91:             inquire( file = chot, exist = exex )
 92:             if( exex .eqv. .true. ) then
 93:               write(io,*)
 94:               write(io,*) ' ** Warning : the file already exists'
 95:               write(io,*) ' ** Do you want to overwrite ?'
 96:               write(io,*) ' ** Yes <= 0, No <= 1'
 97:               read(in,*,end=995) iyes
 98:               if( iyes .ne. 0 ) goto 999
 99:             end if
100:             if( iasb .eq. 0 .or. iuser .ne. 0 ) then
101:               open(ia, file = chot,
102:      &             form='formatted',status = 'unknown' )
103:             else
104:               open(ia, file = chot,
105:      &             form='unformatted',status = 'unknown' )
106:             end if
107:  *-----------------------------------------------------------------
108:  *        read the number of records to read
109:  *-----------------------------------------------------------------
110:          write(io,*)
111:          write(io,*) ' ** put the number of records to read'
112:          write(io,*) ' ** all <= 0, or positive integer'
113:          read(in,*,end=994) irec
114:  *-----------------------------------------------------------------
115:  *        start reading the data
116:  *-----------------------------------------------------------------
117:          write(io,*)
118:          write(io,*) ' ** start read and write the data'
119:  *-----------------------------------------------------------------
120:          jrec = 0
121:    100   jrec = jrec + 1
122:          if( irec .gt. 0 .and. jrec .gt. irec ) goto 500
123:    687      continue
124:             if( iasb .eq. 0 ) then
125:                read(id,end=688,err=690)
```

```
126:       &         ( dmpd(isdmp(k)), k = 1, isdmp(0) )
127:              else
128:                 read(id,'(30(1p1d24.15))',end=688,err=690)
129:       &         ( dmpd(isdmp(k)), k = 1, isdmp(0) )
130:              end if
131:              goto 689
132:    688       if( irec .gt. 0 ) then
133:                 rewind id
134:                 goto 687
135:              else
136:                 goto 500
137:              end if
138:    690       continue
139:              iserr = iserr + 1
140:              write(io,'(''  ** Error in dump file no ='',i5)') iserr
141:              goto 687
142:    689       continue
143:  *-----------------------------------------------------------------------
144:  *     user program here
145:  *-----------------------------------------------------------------------
146:        if( iuser .ne. 0 ) then
147:              do k = 1, 20
148:                 if( jsdmp(k)  .gt. 0 ) dmpp(k) = dmpd(k)
149:              end do
150:                 kf = nint( dmpp(1) )
151:                 x  = dmpp(2)
152:                 y  = dmpp(3)
153:                 z  = dmpp(4)
154:                 u  = dmpp(5)
155:                 v  = dmpp(6)
156:                 w  = dmpp(7)
157:                 e  = dmpp(8)
158:                 wt = dmpp(9)
159:                 t  = dmpp(10)
160:                 n1 = nint( dmpp(11) )
161:                 n2 = nint( dmpp(12) )
162:                 n3 = nint( dmpp(13) )
163:                 sx = dmpp(14)
164:                 sy = dmpp(15)
165:                 sz = dmpp(16)
166:                 n0 = nint( dmpp(17) )
167:                 nc = nint( dmpp(18) )
168:                 nb = nint( dmpp(19) )
169:                 no = nint( dmpp(20) )
170:        end if
171:  *-----------------------------------------------------------------------
172:  *     write data on the file
173:  *-----------------------------------------------------------------------
174:        if( iuser .eq. 0 ) then
175:              if( iasb .eq. 0 ) then
176:                 write(ia,'(30(1p1d24.15))')
177:       &         ( dmpd(isdmp(k)), k = 1, isdmp(0) )
178:              else
179:                 write(ia)
180:       &         ( dmpd(isdmp(k)), k = 1, isdmp(0) )
181:              end if
182:        end if
183:  *-----------------------------------------------------------------------
184:           goto 100
185:  *-----------------------------------------------------------------------
186:  *        end of process
187:  *-----------------------------------------------------------------------
188:    500    continue
189:           write(io,*)  ' ** end of read and write the data'
190:           write(io,'(''  ** number of processed records is '',
191:       &         i8)') jrec-1
192:           write(io,*)
193:           close( id )
194:           close( ia )
195:           goto 999
196:  *-----------------------------------------------------------------------
197:    992 continue
```

```
198:        write(io,*) ' ** Error : ID should be 1 - 20'
199:        goto 999
200:    993 continue
201:        write(io,*) ' ** Error : the ascii or binary frag is wrong'
202:        goto 999
203:    994 continue
204:        write(io,*) ' ** Error : the number of records is wrong'
205:        goto 999
206:    995 continue
207:        write(io,*) ' ** Error : the answer should be 0 or 1'
208:        goto 999
209:    996 continue
210:        write(io,*) ' ** Error : the ID numbers is wrong'
211:        goto 999
212:    997 continue
213:        write(io,*) ' ** Error : the number of data is wrong'
214:        goto 999
215:    998 continue
216:        write(io,*) ' ** Error : file name is wrong'
217:        goto 999
218:    999 continue
219:        stop
220:        end
```

このプログラムは、アスキーもしくはバイナリーのダンプファイルを読み込んで、バイナリーもしくはアスキーのダンプファイルに書き換えるプログラムになっています。

入力パラメーターは、標準入力から読み込みます。通常は、入力を促すメッセージの後にパラメーターを入力する会話形式で入力します。まず最初が入力ダンプファイルがバイナリーかアスキーかのフラグです。

```
** 0 => read binary to ascii
** 1 => read ascii  to binary
```

バイナリーの時は 0、アスキーのときは 1 を入力します。次に、入力ダンプファイル名を聞いてきます。

```
** put the file name of input dump file
```

入力ダンプファイル名をを打ち込みます。次に、

```
** put the number of data in a record
```

のように、ひとつのレコードの dump データの個数を聞いてきます。バイナリー、アスキー両者ともここでは正の数で答えます。

```
** put the ID numbers of data in a record
```

ここで、dump データの個数分のデータの ID を打ち込みます。dump データの種類と ID 番号は、表 77, 78 にあります。

```
** put the file name of output
```

ここに、出力のファイル名を打ち込みます。もし、そのファイルが存在する場合には、上書きするかどうか聞いてきます。次に、

```
** put the number of records to read
** all <= 0, or positive integer
```

のように、読み込むデータのレコード数を聞いてきます。0 を打ち込むと全てのデータ、正の数を入れるとその個数だけ処理します。もしこの数がデータファイルのレコード数より多い場合は、データの最初に戻って処理を続けます。この後、データを読み込み、出力ファイルに書き込んで終了します。最後に、実際に何個のレコードを処理したかが打ち出されます。

このプログラムをもとにユーザープログラムを書くには、35 行目の iuser を 1 にします。そうすることによって、出力ファイルへの変換データの書き出しはしません。また、出力ファイルはアスキーとなります。次に、150 行目から 169 行までを見てください。ここに、読み込んだデータを、kf, x, y, z, u, v, w, e, wt, t, n1, n2, n3, sx, sy, sz, n0, nc, nb, no という変数で定義しています。ここで、kf は、粒子を識別する kf-code、表 4 を参照してください。x, y, z は、座標 (cm)、u, v, w は、運動量の単位ベクトル、e は、エネルギー (MeV)、原子核の場合は核子当たりのエネルギー、wt は粒子ウエイト、time は時間 (nsec)、n1, n2, n3 は、カウンターの値です。入力データにない変数は、29-33 行目で定義されているデフォルト値が入ります。必要ならここを書き換えてください。この部分に統計処理をするプログラム、また、最後に出力のための write 文を書きます。

9　出力中性子、光子データフォーマット

中性子、光子、電子、陽電子、陽子に関しては、計算打切エネルギー以下の情報を、MCNP, EGS4 などの他のコードで接続計算をするために、ファイル (file(12), file(13), file(10)) に書き出すことができます。ファイルにバイナリーで書き出す書式を以下に示します。これらは、中性子、光子、電子、陽電子、陽子とも同じです。

データ書式は、基本的に次の様な形になっています。

```
rd, rn, ( data(i), i = 1, nint(abs(rd)) )
rd, rn, ( data(i), i = 1, nint(abs(rd)) )
................
................
```

これらは、それぞれの場合に具体的に次の様になっています。
まず、`incut = 1` でインポータンスを含まない場合 (rd<0)、

```
rd, rn, x, y, z, ( e(i), u(i), v(i), w(i), i = 1, n )
rd, rn, x, y, z, ( e(i), u(i), v(i), w(i), i = 1, n )
................
................
```

次に、`incut = 1` でインポータンスを含む場合 (rd>0)、

```
rd, rn, x, y, z, ( e(i), u(i), v(i), w(i), wt(i), i = 1, n )
rd, rn, x, y, z, ( e(i), u(i), v(i), w(i), wt(i), i = 1, n )
................
................
```

次に、`incut = 2` でインポータンスを含まない場合 (rd<0)、

```
rd, rn, x, y, z, ( e(i), u(i), v(i), w(i), t(i), i = 1, n )
rd, rn, x, y, z, ( e(i), u(i), v(i), w(i), t(i), i = 1, n )
................
................
```

次に、incut = 2 でインポータンスを含む場合 (rd>0)、

```
rd, rn, x, y, z, ( e(i), u(i), v(i), w(i), wt(i), t(i), i = 1, n )
rd, rn, x, y, z, ( e(i), u(i), v(i), w(i), wt(i), t(i), i = 1, n )
................
................
```

上記の記述で、n = nint(rn)、x, y, z は、座標 (cm)、e(i) は、エネルギー (MeV)、u(i), v(i), w(i) は、運動量の単位ベクトル、wt(i) は、粒子のウエイト、t(i) は、時間 (ns) です。

igcut = 3 の場合は、上の incut = 2 の場合の t(i) の変わりに、粒子の判別子 p(i) が入ります。

p(i) = 3.0 が光子、p(i) = 4.0 が電子、p(i) = 5.0 が陽電子です。

10　領域エラーチェック

3 次元体系を構築する際、ある場所を 2 つ以上の領域で定義してしまう多重定義や、1 つも領域を設定しない未定義領域が生じる可能性があります。これらの領域エラーが発生した状態では、輸送している粒子の現在位置がわからなくなってしまう場合があり、適切な輸送計算を実行することができません。領域エラーチェックは、構築した体系内に 2 重定義や未定義領域がないか確認するための機能です。この機能を利用して領域エラーのない 3 次元体系を作成してください。

領域エラーチェック機能は、ジオメトリを 2 次元表示するタリーで自動的に実行されます。具体的には `[t-gshow], [t-rshow]` タリー、及び `icntl=8 or 10` の場合に `axis = xy, yz, or xz` として gshow or rshow オプションを指定したタリーが該当します。領域エラーチェック機能をコントロールするパラメータは各タリーの `ginfo` です。`ginfo=0, 1, 2` が指定でき、それぞれ、チェックしない、2 次元ジオメトリ図上にエラー領域を表示する（図 38 参照）、エラー領域表示に加えて後述の領域エラーファイル出力する、となります。ジオメトリを表示するタリーに対しては、自動的に `ginfo=2` となります。また、飛跡長などを 2 次元表示する通常のタリー（`icntl = 0` かつ `axis = xy, yz, or xz`）でも、`ginfo=1 or 2` と指定すれば領域エラーチェックを行います。

領域エラーがある場合、図 38 のような出力結果が得られます。多重定義領域がある場合はその領域を黒色で、未定義領域がある場合はその領域を紫色で表示します。ただし、未定義領域がある場合に、その周りの定義された領域が消えてしまう場合があります。

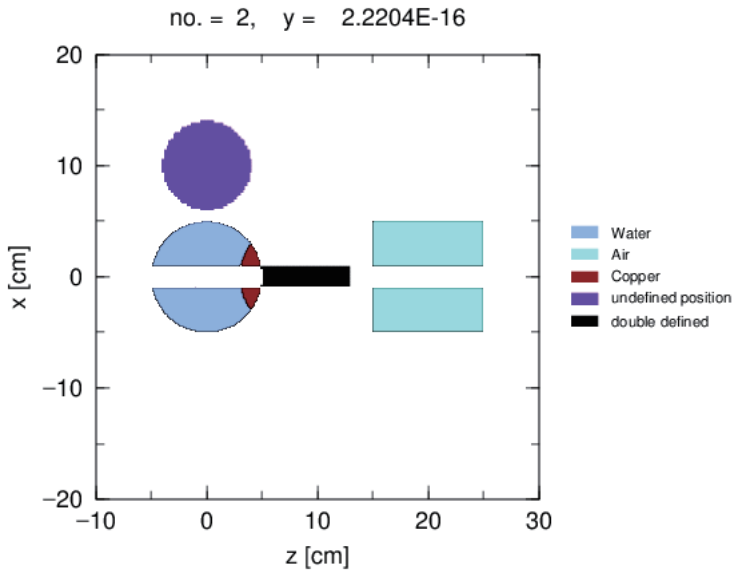


図 38: 領域エラーがある場合の出力結果

`ginfo=2` の場合、`file=`で指定したファイル名の拡張子を “.err” に変えた領域エラーファイルを作成します。このファイルには、エラーが起こっている領域の (x,y,z) 座標が次のような形式で出力されます。

```
Errors of cell definition in EPS Page No. =   1
Overlapped Cell IDs  x, y, z  coodinates
(Cells 0       0  indicate undefined region)
     100     102    -4.847761E+00  1.234568E-11 -1.211940E+00
       0       0    -4.241791E+00 -2.500000E+00 -8.079602E-01
```

1 行目のエラーは、セル 100 と 102 が x = -4.847761E+00, y = 1.234568E-11, z = -1.211940E+00 で重なって

いることを示しています。2 行目のエラーは、 x = -4.241791E+00, y = -2.500000E+00, z = -8.079602E-01 の場所が未定義領域であることを示しています。

この領域エラーチェック機能を用いれば、複雑な 3 次元体系を作成する場合でも、効率的にエラー領域を見つけることができます。ただしこの機能は、タリー領域で指定している範囲しかチェックしませんのでご注意ください。また、範囲内であっても、xyz メッシュで区切った格子点しかチェックしませんので、小さな領域エラーがある場合は見逃してしまう可能性がありますので、ご注意ください。

11　並列版のための指定方法

並列計算には、メモリ分散型並列とメモリ共有型並列計算があり、PHITS では、その両方に対応しています。また、両者を組み合わせたハイブリッド並列も可能です。メモリ分散型並列のためには、お使いのコンピュータに MPI をインストールする必要があります。メモリ共有型並列は、特にプロトコルなどをインストールする必要はありません。シングル・並列計算の切り替えはコンパイラオプションで行い、それぞれ別の実行ファイルを作る必要があります。詳しくは 2.4 節をご参照ください。

メモリ分散型並列では、各コアにそれぞれバッチ単位で Job を渡して、全てのコアがそのバッチ計算を終了した時点でメインのコアが結果を集計します。全てのコアがジオメトリやタリーの情報をそれぞれ読み込むため、メモリは、シングルで走らせるときのコア数倍必要となります。したがって、ボクセルファントムなど、巨大なメモリを必要とする計算には不向きです。また、全てのコアが計算終了するまで集計を待つため、バッチ当たりのヒストリー数 (`maxcas`) が少ないなど、各コア間での計算時間にばらつきが大きい場合は、計算時間が不必要に長くなってしまいます。

メモリ共有型並列では、各コアにヒストリー単位で Job を渡して、各コアは、ジオメトリやタリーなどの情報を共有しながら計算します。したがって、メモリの使用量は、シングルで走らせる場合と大差はありません。しかし、メモリへの書き込みアクセスが競合するため、タリー結果などを頻繁に書き込む計算 (`[t-sed]` を使った計算など) では、計算時間が不必要に長くなってしまいます。

11.1　メモリ分散型並列

11.1.1　実行方法

メモリ分散型並列には MPI プロトコルを用いるため、その実行コマンドは

```
mpirun -np 8 phits_LinIfort_MPI.exe
```

のようになります。ここで`-np` の後の数字は並列の実行 PE 数で、phits_LinIfort_MPI.exe は PHITS の実行ファイル名です。このコマンドを qsub など各コンピュータに合わせた方法で投入してください。また、メモリ分散型並列版では、自動的に `phits.in` ファイルから入力ファイル名を読み込みます。この `phits.in` は固定です。このファイルの 1 行目に

```
file = input_file_name
```

のように入力ファイル名を記述します。従って、起動シェルでインプットファイルのリダイレクトはできませんので注意して下さい。これは、メモリ分散型並列だけの制約です。シングル版はこれまでの通りです。

なお、`phits.in` の 1 行目に `file=phits.in` と書けば、それ以下の行に通常のインプットファイルを書いて実行することもできます (2.4 節参照)。

11.1.2　maxcas, maxbch の指定方法

PHITS のメモリ分散型並列計算では、バッチ単位で並列化しています。従って、入力のバッチ数 `maxbch` は、並列の実行 PE 数 (コントロールに 1PE 使うので、全 PE 数 −1) の整数倍になるように指定して下さい。もし整数倍でないときは、整数倍になるように、またトータルヒストリー数が入力データとほぼ同じになるように、プログラムが自動で変換します。変換した場合は、出力のインプットエコーの最後にコメントが出力されます。後述しますが、バッチ毎の情報はメモリ分散型並列の場合は、バッチ数 × (PE −1) 毎に出力されます。途中停止もこの単位で行うことができます。

なお、再開始計算の場合 (`istdev < 0`) は、`maxcas` を過去の計算結果と一致するよう自動調整するため、全ヒストリー数を合わせる上記調整は行いません。`maxbch` を (PE−1) の整数倍にする調整のみ行います (端数は切り捨てます)。

11.1.3 異常終了の処理

プログラムが異常終了した場合、その異常終了した PE を切り離して実行します。最終的に残った PE の計算結果の合計を全体の結果として出力します。バッチ毎の情報と計算のサマリーの中に PE の状態が出力されます。MCNP などの他の計算コードとの接続計算の時は、異常終了の PE がある場合、ncut ファイルがその PE で不完全ですので、注意が必要です。

11.1.4 PHITS での ncut, gcut, pcut, dumpall ファイルの指定

ncut, gcut, pcut ファイル名は、通常のように入力ファイルで、

```
file(12) = temp/ncut.dat
```

のように指定します。1PE で走らせる場合は、カレントディレクトリの (パスを含む場合はそのディレクトリの)ncut.dat に書きこまれます。2PE 以上の計算では、デフォルトで各ノードの

```
/wk/j9999/temp/ncut.dat
```

に分散して書きこまれます。ここで `j9999` は、ユーザー名です。これは環境変数 `LOGNAME` から自動で読み込んでいます。この環境変数 `LOGNAME` は、デフォルトでユーザー名が入っています。計算の前に各ノードの `/wk` ディレクトリに `j9999` なるディレクトリを作ってください (作り方は並列コマンド `brsh` を使います)。もし、ユーザー名以外のディレクトリに ncut ファイルを作りたいときは、実行前に環境変数 `LOGNAME` を変更して運用して下さい。その時、その名前のディレクトリが、各ノードの`/wk` ディレクトリにあることを確認して下さい。

書きこみファイルのオプションとして、inpara, igpara, ippara があります (デフォルトで 0 です)。これらを 1 にすると、ファイル名の後に IP の番号を付けます。例えば、

```
/wk/j9999/temp/ncut.dat.005
```

これは、IP=5 のファイル名です。

inpara, igpara, ippara を 3 にするとデフォルトでついている ヘッダー

```
/wk/j9999/
```

が付きません。3 は 1 と同じくファイル名の最後に IP の番号を付けます。

11.1.5 PHITS での読み込みファイルの指定

PHITS で読み込みが必要となるファイルは、Decay-Turtle からの出力をソースとして使うときのソースファイルです。前者は、2.6MB のファイルで、各 PE が同じファイルを同時に読み込みを初めても、読み込みが最初の一回だけということもあり、それほどネットワークに負担を掛けないと思われますので、1 箇所に置いていても問題無いと思います。しかし、後者はファイルが 100MB 近く大きいもので、ソースの読み込みのたびにファイルをアクセスしに行きますので、1 箇所からの読み込みには問題が生じる可能性があります。そこで、Decay-Turtle のデータファイルを利用する場合は、あらかじめこのファイルを 各 PE のワークディレクトリ `/wk` の下に、例えば、`/wk/j9999/turtle/sours.dat` のようにコピーして、PHITS のインプットで、`file = /wk/j9999/turtle/sours.dat` のように指定して下さい。

11.2 メモリ共有型並列

11.2.1 実行方法

メモリ共有型並列版 PHITS の実行方法は、シングル版 PHITS の実行方法と同じく

```
phits_LinIfort_OMP.exe < phits.inp
```

のようになります (phits_LinIfort_OMP.exe は実行ファイル名)。メモリ分散型並列と異なり、リダイレクトが使えるため、インプットファイル名 (`phits.inp`) は可変です。ただし、並列数を指定するため、環境変数 OMP_NUM_THREADS をコア数に合わせて設定する必要があります。通常、この環境変数はデフォルトでコア数に設定されていますが、最新のコンピュータは、1 つのコアで複数のスレッドを処理することが可能なため、コア数よりも多く設定されている場合があります。しかし、PHITS の並列処理では、OMP_NUM_THREADS にコア数以上の値を設定しても、ファイルの書き込み競合による計算遅延が起きてしまい、計算時間が長くなってしまいます。そのような場合は、手動で環境変数を変更してください。環境変数の変更は

```
set OMP_NUM_THREADS=4 (Windows の場合)
export OMP_NUM_THREADS=8 (Mac, Linux の場合)
```

のように行います。もし、作成した実行ファイルを「送る」機能で使用する場合は、`\phits\bin` フォルダにある `phits.bat` をテキストエディタで開き、その 3 行目でコア数を指定してください。例えば 4 コアで計算する場合は、

```
set PHITS_PARALLEL=4
```

と設定してください。なお、PHITS のメモリ共有型並列は、Hyper Threading 技術には対応しておりませんので、`PHITS_PARALLEL=0` としてデフォルトの `OMP_NUM_THREADS` を使っても計算時間が長くなる場合があります。その場合は、OS が認識しているコア数の半分の値を `PHITS_PARALLEL` に直接入力してください。また、ハイブリッド並列の場合は、各実行ノードの環境変数をそれぞれ変更する必要があることにご注意ください。

バージョン 2.73 より、Windows OS については、メモリ共有型並列版 PHITS の実行ファイルとして 64bit 版をインストールするようにしました。32bit 版のメモリ共有型並列の実行ファイルの場合、使用するコア数が増えるとヒープメモリが不足してエラーが発生していましたが、64bit 版を使用することでこれが回避される可能性があります。

11.2.2 メモリ共有型並列計算の注意点

コア数が 1 つの場合、メモリ共有型並列版 PHITS の計算時間は、シングル版 PHITS の計算時間よりも 2 倍くらい長くなります。したがって、コア数が 2 以下のコンピュータでは、メモリ共有型並列計算による計算時間短縮のメリットはほとんどありません。

また、現在、下記の計算機能がメモリ共有型並列計算に対応できていません。

(1) *AnGeL* の起動 (タリーで `epsout=1` と指定した場合、Windows のみ)

(1) 番目は Windows 固有の問題で、Intel Fortran Compiler を用いてコンパイルした場合に起こります。そこでバージョン 2.52 では、実行時にバッチファイル `phits.bat` を用いることで、`epsout=1` と指定しても動作するようにしました。このバッチファイルを使うと、PHITS 計算の最初に内部で `epsout=0` に設定し直し、計算終了後に *AnGeL* を用いて eps ファイルを作成します。ただしこの場合、`itall=1` を設定してバッチ終了毎にタリー結果を更新する、という機能が使えませんのでご注意ください。また、ユーザー自身でコンパイルする際は、`a-angel.f` を `a-angel-winopenmp.f` に変更する必要があります（2.5 節参照）。

また、Linux で実行時にセグメンテーションエラーが生じる場合は、実行前にスタックサイズを広げることにより解決する場合があります。スタックサイズの広げ方は、

```
export OMP_STACKSIZE=1G
```

のようになります (スタックメモリを 1GB にする場合)。

メモリ共有型並列計算結果は、シングルでの計算結果と一致するように設計しています。両者で結果が異なる場合は、バグの可能性がありますので PHITS 事務局までご連絡ください。

12 FAQ

12.1 パラメータ設定関連

Q1.1 低エネルギー中性子の輸送計算精度が良くないのですが。

A1.1 一般に 20MeV 以下の中性子の挙動は複雑で、核内カスケードモデル（INC) や量子分子動力学モデル（QMD）では再現できません。したがって、核データライブラリーを使用した計算が必要となります。中性子に対して核データを利用するには、`emin(2)`=1.0e-10, `dmax(2)`=20.0 のように、中性子の計算打切エネルギーを下げ、核データを利用する上限エネルギーを大きくしてください。

Q1.2 光子の輸送計算ができません。

A1.2 デフォルトの設定では、光子 · 電子の輸送計算は行われません。光子・電子の輸送計算が必要な場合は、`[parameters]` セクションで `negs=1` として，EGS5 の輸送アルゴリズムを起動するようにしてください。また，その上限，下限エネルギーは `dmax(12-14)` と `emin(12-14)` で指定できます。`negs` を指定せずに `dmax(12-14)` や `emin(12-14)` を指定すると，古い PHITS のアルゴリズムを使って光子・電子の輸送計算を行います。なお，入射エネルギーと `dmax` の値が完全に一致すると結果が変わってしまう場合がありますので，`dmax` は入射エネルギーよりも少し高い値を設定して下さい。

Q1.3 核反応モデルに関して、デフォルトの設定は最も精度がよい設定ですか？

A1.3 計算する問題に依存しますが、バージョン 2.50 以降では、基本的にそうなるように設定しています。ただし、20MeV 以下の中性子を扱う場合、 核データを用いた計算が必須になります (Q1.1 参照)。

Q1.4 イベントジェネレータモードは、どのような場合に使えばよいのでしょうか？

A1.4 イベントジェネレータモード（以下、e-mode）を使った方がよい計算は、検出器の応答関数計算や、半導体ソフトエラー発生率の計算など、イベント毎の情報が必要となる計算です。具体的には、`[t-deposit]`, `[t-let]`, `[t-yield]` などを使う計算では、e-mode を使った方がよい場合が多いです。逆に、使わない方がよい計算は、中性子束やガンマ線束だけを求める遮蔽計算などです。`[t-track]`, `[t-cross]` などしか利用しない場合は、一般的に e-mode を使う必要はありません。詳しくは「4.2.22 Event Generator Mode」をご参照ください。なお，version2.76 からイベントジェネレータモードを使う場合の奨励値は `e-mode=2` となっていますので，ご注意ください。

Q1.5 統計誤差導出方法 (`istdev` の設定) はどういう条件で切り替えれば良いですか？

A1.5 バッチ分散モード (`istdev=-1` or 1) しか選択できないメモリ共有型並列計算の場合を除き、基本的に全ヒストリー数 (`maxcas*maxbch`) で統計誤差を評価するヒストリー分散モード (`istdev=-2` or 2) の方を推奨します。しかしながら、ヒストリー分散モードでは、ヒストリーの計算が終了する毎に誤差情報を記録するため、タリーのメッシュ数が大きい場合 (xyz メッシュで `nx` や `ny` を数 1000 ずつ設定するなど) には、計算時間が異常に長くなる可能性があります。そのような条件で計算する際は、バッチ分散モードに切り替えて、`maxbch` を十分に大きく (10 以上) 設定するようにしてください。

Q1.6 JENDL 以外の核データを PHITS で使うことは可能でしょうか？

A1.6 ACE フォーマット（MCNP が読み込める形式）であれば可能です。以下，その手順です。

(1) 新しく入手した核データに付録されている xsdir ファイルをテキストエディタで開き，核データのアドレス情報に関する部分（例えば 1001.80c 0.999167 xdata/endf71x/H/1001.710nc 0 1 4 17969 0 0 2.5301E-08）を全て PHITS の data フォルダ内にある xsdir.jnd の末尾にコピー＆ペーストする。

(2) PHITS の XS フォルダ内に，上記アドレス情報が一致するように新しく入手した核データファイルをコピーする。例えば，上記のアドレス情報であれば，XS フォルダ内に xdata/endf71x/H フォルダを作って 1001.710nc ファイルをその中にコピーする。

(3) [material] セクションで，使う核データを明示的に指定する。例えば，上記のアドレス情報であれば 1H.80c もしくは 1001.80c と指定する。核データを明示的に指定しない場合は，xsdir.jnd の中でその核種に対して最初に書かれている核データが自動的に使われます。

Q1.7 光核反応により生成する粒子（中性子・放射性核種）の統計を効果的にためる方法はありますか？

A1.7 pnimul パラメータを使うことにより，他の反応（コンプトン散乱など）と比べて光核反応を起こしやすくすることができます。例えば，pnimul=2.0 とすると，光核反応を引き起こす確率を 2 倍にする代わりに，光核反応で生成した粒子のウェイトを 1/2 にします。ウェイトを変化させるので，イベント毎の情報が必要な場合は使えません。また，pnimul パラメータを大きくしすぎると（100 以上），光子・電子の挙動がずれてきてしまいますので，本パラメータを使う場合は，光子・電子の挙動（フラックス・線量など）が大きく変動しないことを確認しながら使ってください。

Q1.8 再開始計算を行おうとすると，タリーを変更していないにも関わらず “Error: inconsistent tally parameters” というエラーが出て止まります。対処方法はありますか?

A1.8 この問題は，桁落ちの関係で新旧タリーのパラメータが異なると PHITS が認識してしまうことにより発生します。タリーで使うパラメータを複雑な数式などで定義した場合によく発生します。この問題が生じた場合は [parameters] セクションで ireschk = 1 と設定し，タリーの一致性をチェックしないモードで再開始計算を実行して下さい。

12.2 エラー、コンパイル関連

Q2.1 コンパイルがうまくできません。

A2.1 PHITS のコンパイルに関しては，基本的には「2.4Makefile を利用したコンパイル」をご参照ください。PHITS 事務局で推奨しているコンパイラは Intel Fortran Compiler（11.1 以降）と gfortran（4.7 及び 4.8）です。gfortran の最新版（4.9 以降）には対応していませんのでご注意ください。また、makefile に書かれていないオプションや環境でコンパイル&実行に成功した方は、PHITS 事務局までご連絡ください。

Q2.2 巨大な体系について計算しようとすると、Segmentation エラーが起きてしまいます。

A2.2 *PHITS* では使用するメモリの最大値をあらかじめ定義しており、実際に必要となるメモリがその最大値を超えてしまうと、Segmentation エラーを引き起こします。使用するメモリの最大値 は、「src」フォルダにある「param.inc」中の mdas パラメータで定義されていますので、必要に応じてこの値を大きくし再コンパイルしてください。詳しくは「2.9 配列の大きさ」をご参照ください。また、ボクセルファントムなど巨大な Lattice 構造を扱う場合は、同時に latmax パラメータも大きくしてください。なお、Windows および Mac では 1 つの FORTRAN プログラムが 2GB 以上のメモリを使用することは許されていませんので、mdas の上限値は 268435450 です。それ以上のメモリ領域が必要となる計算には、64bit の Windows もしくは Linux をお使い下さい。

Q2.3 外部ファイルを取り込むためのコマンド「infl:」が使えません。

A2.3 infl:コマンドを使うためには、インプットファイルの 1 行目に「file=インプットファイル名」と書いておく必要があります。もしくは、通常の *PHITS* 用インプットファイルの他に、別のファイルを準備し、そのファイルの先頭行に「file=通常 *PHITS* のインプットファイル名」と書いて実行することも可能です。詳しくは、「2.7 実行シェル」をご参照ください。

Q2.4 WINDOWS では実行できるインプットファイルが LINUX や UNIX では実行できません。

A2.4 いろいろな原因が考えられますが、WINDOWS と LINUX 系 (UNIX 含む) では、リターンコードが違いファイルの転送に失敗している可能性があります。通常、WINDOWS から LINUX 系にファイルを転送する際、FTP などのプロトコルを使うと思いますが、その転送の際、アスキーモードとバイナリーモードというのがあり、アスキーファイル (*PHITS* のインプットファイルなど) は、アスキーモードで

転送しないと、ファイルが破損してしまう可能性があります。お使いの FTP ソフトの設定をご確認ください。なお、Windows で作られたバイナリファイル (*PHITS* の実行ファイルなど) を LINUX 系に転送しても、通常、正しく動作しません。

Q2.5 Cygwin 上で *PHITS* は動作しますか？
A2.5 はい。*PHITS* の makefile にある Cygwin の項目をご参照ください。

Q2.6 Intel fortran と gfortran，どちらが PHITS をコンパイルするのに適していますか？
A2.6 基本的には，Intel fortran を使うことをお勧めいたします。理由は，計算速度が速いためです。gfortran では，最適化オプションを付けると正しく動作しない場合があるため，奨励の最適化オプションは-O0（すなわち最適化しない）です。したがって，平均で３-５倍くらいの計算速度の差があると考えています。また，最新の gfortran では PHITS を正しくコンパイルできない問題もあります（Q2.1 参照）。ただし，Intel fortran は Linux 版を含めて有償ですので，その点，ご留意ください。

12.3 タリー関連

Q3.1 [t-heat] と [t-deposit] は何が違うのでしょうか？
A3.1 ユーザー定義関数 (usrdfn) を指定しない限り、基本的には同じになります。大きな違いは、[t-heat] では非荷電粒子（中性子や光子）のカーマ近似を使った計算が可能である点です。カーマ近似を使う場合は、中性子に関しては [parameters] セクションにおいて e-mode=0 とし、光子に関しては [t-heat] セクションにおいて electron=0 としてください。また、[t-deposit] は、被ばくによる生物学的効果など同じ発熱量でも影響の大きさの違う量を計算する際、極めて有用となります。

Q3.2 [t-track] で重イオンの track length を計算すると、結果が不自然なのですが。
A3.2 重イオンのエネルギーは、入射エネルギーや切断エネルギーに対しては核子あたりのエネルギー (MeV/u) で指定しますが、タリーの中でエネルギーを指定する場合は、核子あたりではなく、全エネルギー (MeV) で指定します。ややこしくて申し訳ないのですが、従来からこの方針でプログラミングしておりますので、ご了承ください。

Q3.3 [t-let]、[t-sed] の結果が不自然なのですが。
A3.3 letmat で LET を計算する物質を指定する際、その物質が *PHITS* の体系内で使われている場合、密度はその体系内で使用している密度となります。例えば、水蒸気内の発熱量に対する LET 分布を、液体の水 (1g/cm^3) に対する LET の関数として計算したい場合、[Material] で２つの水を定義し、１つを *PHITS* の体系内で使用し、もう１つを letmat で使用する必要があります。また、letmat で使用する物質の密度は、原子数密度の絶対値 (1H 6.893407e-2 16O 3.446704e-2) を指定してください。詳しくは，奨励設定ファイルの ParticleTherapy をご参照下さい。

Q3.4 タリー結果の誤差はどのように計算されているのでしょうか？
A3.4 バージョン 2.50 より、正しく計算できるように修正しました。詳しくは「4.2.2 ヒストリー数、バンク配列の大きさ」をご参照ください。

Q3.5 並列計算で dump モードは使えるでしょうか？
A3.5 基本的には使えます。ただし，メモリ共有型（OpenMP）並列の場合は，1 つのファイルに全てのノードからの情報を書き込むため，書き込む順番がヒストリー番号毎に並ばなくなってしまいます。一方，メモリ分散型（MPI）並列の場合は，dump ファイル名にコア番号付けたファイルを出力しますので，そのような問題は起きません。

Q3.6 mesh = reg と mesh = xyz を用いて同じ直方体内のフラックスや発熱量を計算した場合に、それぞれの結果が異なるのですが。

A3.6 タリー結果を単位体積あたりに規格化して出力する場合、xyz や r-z メッシュではタリー領域の体積を自動的に計算して出力しますが、reg メッシュの場合は、[volume] セクションで体積を指定しない限り体積を 1cm^3 として出力します。例えば、mesh = reg で [volume] セクションがない場合、[t-track] の unit=1[1/cm^2/source] と unit=4[cm/source] は同じ値を与えます。

Q3.7 [t-deposit] で output=deposit とした場合に入射エネルギーよりも高いイベントが発生するのはなぜでしょうか？

A3.7 発熱する核反応（Q 値が正）が発生すると，入射粒子よりも 2 次粒子の合計エネルギーが高くなるため，入射エネルギーよりも付与エネルギーが大きくなる場合があります。また，xyz メッシュもしくは r-z メッシュかつエネルギー分散を考慮した場合（nedist≠0），エネルギー分散の考慮方法の仕様により，イベント毎の情報が正しく計算できなくなってしまいます。そのような場合は，mesh = reg として，[cell] セクションで領域を細かく分割して計算してください。

12.4 線源設定関連

Q4.1 等方線源ソース (s-type=9,10, dir = −all) のときの規格化はどのようにすればよいですか？

A4.1 *PHITS* で作る等方線源は、指定した球の中でフラックスが一定値: $1/\pi/r_1^2$ (r_1 は球の内径) となるように作られています。したがって、*PHITS* の結果 (/source) を単位フルエンス当たりに変換するには、結果に πr_1^2 を乗じてください。また、等方線源の作成にはウェイトの概念を利用していますので、検出器の応答関数などイベント毎の分布計算には利用できませんので、ご注意ください。そのような計算が必要な方は、`dir = iso` としてください。

Q4.2 *PHITS* で認識する線源領域が、実際の領域とずれるのですが。

A4.2 面線源や点線源を定義する際、その面や点が、Cell で使う surface と完璧に一致している場合、*PHITS* の中で認識する線源領域が実際の領域とずれる場合があります。このような状態を避けるため、面線源や点線源は、cell で使う面とは少しずらして作成するようにしてください。また、特定の面に完璧に平行なビームを発生させた場合も同様のエラーが起きる可能性がありますのでご注意下さい。

索 引

<source>, 62
¡source, 84
[Weight Window], 221
[brems bias], 21, 143
[cell], 8, 21–23, 44, 63, 99, 103, 105–110, 112, 113, 115–119, 122, 126, 139, 141, 148, 219
[counter], 3, 21, 151, 166
[data max], 2, 21, 135
[delta ray], 21, 132
[elastic option], 21, 134
[electro magnetic field], 21, 63, 131
[end], 22, 23
[forced collisions], 21, 142
[frag data], 1, 5, 21, 136
[importance], 21, 45, 139, 140
[magnetic field], 21, 63, 128, 131
[mat name color], 21, 44, 148
[mat time change], 21, 127
[material], 8, 21, 44, 98, 100, 101, 106, 110, 112, 113, 115–119, 122, 148, 219
[multiplier], 11, 21, 146, 147, 174, 180, 184
[parameters], 2, 4, 5, 19, 21, 28, 84, 99, 120, 131, 135, 142, 148, 169, 186, 196, 220, 256
[photon weight], 21, 144
[reg name], 21, 150
[source], 3, 6–9, 21, 25, 36, 61, 129, 131, 217
[super mirror], 21, 133
[surface], 21–23, 55, 102, 105–110, 112, 113, 115–119, 122, 226
[t-3dshow], 1, 22, 28, 150, 153, 233
[t-cross], 8, 11, 22, 153, 165, 176, 182, 240
[t-dchain], 1, 4, 9, 10, 19, 22, 153, 217
[t-deposit2], 22, 30, 153, 193, 194
[t-deposit], 3, 4, 7, 8, 22, 30, 68, 153, 189, 190, 192
[t-dpa], 4, 22, 153, 202
[t-gshow], 22, 28, 114, 150, 153, 162, 164, 165, 229–231
[t-heat], 4, 8, 22, 153, 186
[t-let], 22, 153, 205
[t-point], 3, 22, 153, 182, 184
[t-product], 8, 11, 22, 28, 153, 198
[t-rshow], 22, 28, 150, 153, 164, 165, 231, 232
[t-sed], 22, 153, 208
[t-star], 3, 22, 153, 214
[t-time], 8, 11, 22, 153, 211
[t-track], 4, 11, 22, 131, 146, 147, 153, 171, 174, 180, 182, 240, 241
[t-userdefined], 11, 22, 153, 224
[t-wwg], 2, 22, 153, 221
[t-yield], 3, 5, 22, 35, 153, 161, 195
[temperature], 21, 126
[timer], 21, 152
[title], 21, 27
[transform], 21, 102, 103, 107, 116, 123–125
[volume], 21, 45, 145, 155, 219, 220
[weight window], 2, 21
[wieght window], 141
2d-type, 164, 172, 178, 188, 192, 194, 196, 201, 202, 207, 210, 213, 216
3dshow, 166

a-curr, 178, 179
a-type, 7, 8, 63, 65–68, 72, 75, 86, 87, 158, 171, 177, 179, 198
absorption, 216
actlow, 85
ag1, 87
ag2, 87
all, 135, 147, 171, 174, 177, 180, 182, 184, 186, 189, 193, 195, 198, 202, 204, 205, 208, 211, 214, 216, 221
amp, 219
andit, 37
angel, 172, 178, 183, 188, 192, 194, 196, 201, 202, 207, 210, 211, 216, 221, 229, 231, 232, 235
ANGEL , 18, 30, 163, 164, 166, 231
angle straggling, 39
area, 179, 241
ascat1, 39
ascat2, 39
ATIMA, 2, 3, 39
axis, 5, 30, 35, 156, 161, 164, 165, 172, 177–179, 183, 186, 189, 190, 194, 196, 201, 202, 204, 207, 210, 211, 213, 214, 216, 221, 229–232, 240, 241

axishow, 235

b-curr, 178
batch.out, 2, 19, 48
becquerel, 84
Bertini, 32, 35, 37
bias, 143
Bitmap, 2
bmpout, 174, 188, 192, 198, 201, 204, 213, 216, 229, 231
bnum, 50
BOX, 104, 113, 115, 117–119
bremsstrahlung, 50, 143
brsh, 252

C/X, 103
C/Y, 103
C/Z, 103
cdiam, 208
CGVIEW, 28, 48
chard, 5, 54
charge, 161, 196
chart, 35, 161, 196
cmin(i), 31
color, 127, 148
COND, 99
cos, 161, 177, 198, 201
CPU 時間, 45
ctmax, 166, 174, 179, 183, 188, 192, 194, 198, 201, 204, 207, 210, 213, 216, 222
ctmin, 166, 174, 179, 183, 188, 192, 194, 198, 201, 204, 207, 210, 213, 216, 222
Cugnon, 35
current, 95–97, 176, 178, 179
cutoff, 211
CX, 103
CY, 103, 112, 116–118
CZ, 103

dbcutoff, 39
dchain, 6, 35, 48, 161, 196
DCHAIN-SP, 217
decay, 201, 216
decay 粒子, 211
decay-turtle, 61, 69, 252
DECDC, 2, 48, 84
dedxfnc, 189
dedxfnc1, 193
dedxfnc2, 193
deg, 172
delayed neutron, 49
delt0, 47
Δ 角分布, 37
δ 線, 4, 9, 11, 21, 132
deltb, 47
deltc, 5, 47
deltg, 47
deltm, 5, 47
deltt, 47
density, 8, 98, 99
deposit, 4, 30, 68, 186, 188, 190
Deposit energy, 208
dfano, 192
dir, 63, 65–72, 75, 86, 240
dircha, 48
dl0, 95
dl1, 95
dl2, 95
dmax, 135
dmax(i), 31, 195
dmpmulti, 73
dnb, 49
dom, 63, 65–67, 70–72, 95
dose, 190
dpa, 11, 202, 204
dpf, 95
drd, 95
DRES, 35
dresol, 192
dtime, 84
dump, 8, 11, 61, 64, 72, 74, 75, 168, 179, 201, 242
dumpall, 8, 11, 28, 46, 48, 54
DWBA, 8, 38
dxw, 95
dyw, 95

e-dst, 233
e-mode, 6, 7, 39, 59, 60, 186, 195, 198, 202
e-phi, 233
e-the, 233
e-type, 8, 72, 75, 79, 81, 83, 157–160, 171, 177, 179, 182, 189, 198, 211, 214, 221, 240, 241
e0, 65–72, 75, 240

e1-t, 161, 193
e1-type, 193
e12, 161, 193
e2-t, 161, 193
e2-type, 193
e21, 161, 193
EBITEM, 9, 35, 196
eg0, 83
eg1, 83
eg2, 83
eg3, 83
EGS5, 3–7, 36, 48, 51
eielfmax, 32
eielfmin, 32
einclmax, 32
einclmin, 32
eisobar, 32
ejamnu, 32
ejampi, 32
ejamqmd, 32
elastic, 3, 216
electron, 50, 60, 186, 188
Element file, 121
elf, 131
ELIB, 99
ELL, 10, 104
emax, 147, 174, 175, 180, 181, 184, 185
emcnf, 49
emcpf, 50
emin(i), 31, 186, 205
emumax, 37
emumin, 37
energy cut off 粒子, 211
energy straggling, 39
eng, 5, 141, 161, 172, 177, 183, 201, 211, 214, 221
eng-t, 161, 189
eng1, 193
eng2, 193
ENSDF, 9
eps, 166, 174, 178, 183, 188, 192, 194, 198, 201, 204, 207, 210, 213, 216, 222, 229, 231
epsout, 166, 174, 178, 183, 188, 192, 194, 198, 201, 204, 207, 210, 213, 216, 222, 229–231, 235
epstfl, 54
eqmdnu, 32
escape, 211
escape 粒子, 211
esmax, 3, 31
esmin, 3, 31
ESTEP, 99
et0, 83
et1, 83
et2, 83
event generator mode, 39
exa, 71
Excel, 164

f-curr, 178
factor, 63, 73, 162, 172, 178, 183, 188, 190, 194, 196, 201, 204, 207, 210, 211, 216, 221
FAQ, 255
fcl, 142
file, 30, 72, 162, 172, 178, 179, 183, 186, 190, 194, 196, 201, 202, 207, 210, 211, 213, 214, 219, 221, 224, 229–232, 235, 240, 241
file(20), 7
file(22), 2, 19
file(24), 84
file(6), 44, 135, 148
file(i), 48
FILL, 106, 107, 112–115, 117–119, 122
fiss, 3
fission, 201, 216
flight mesh, 47
fluence, 171, 176
flux, 68, 176, 178, 180, 240
Fortran, 13, 24, 164

gap, 128
GAS, 99
gcut, 43, 186, 252
GEM, 35
GG, 1, 21, 44, 45, 126, 128, 139, 141, 142, 144, 145, 153, 156
GG(General Geometry), 106, 107
ghostview, 166
ginfo, 174, 178, 188, 192, 198, 201, 204, 207, 210, 213, 216, 222, 229, 231, 249
GQ, 103
grav, 38
gshow, 28, 150, 164–166, 172, 178, 188, 192, 198, 201, 204, 207, 210, 213, 216

gslat, 174, 179, 188, 192, 198, 201, 204, 207, 210, 213, 216, 222, 229

heat, 186
heaven, 233
heavyion, 216
HEX, 104
HLIB, 99
HSB, 148

iaprim, 54
iauger, 54
ibad, 50
ibod, 23
ibound, 54
ibrdst, 54
icntl, 28, 48, 49, 165, 169, 229, 231, 240
icput, 45
icrdm, 2, 37
icrhi, 37, 137
icxsni, 37, 137
icxspi, 2, 38
idam(i), 49
idbg, 23
idelt, 47
ides, 50
idmpmode, 73
idpara, 46
idwba, 38
iechrl, 174, 188, 192, 194, 201, 204, 207, 213, 216, 231
iedgfl, 54
iegsout, 51
iegsrand, 54
ieispl, 54
ielas, 35
ielctf, 37, 131
ieleh, 36
ielms, 35
igamma, 6, 9, 35, 48, 60, 196
igchk, 47
igcut, 43, 186
igerr, 47
iggcm, 45
Ignatyuk, 36
igpara, 43, 252
ih2o, 39
iidfs, 38
imagnf, 37, 47, 128
imout, 44
impacr, 54
importance, 45, 139, 140
imsegs, 5, 51
imubrm, 36
imucap, 36
imuint, 36
imuppd, 36
INC-ELF, 10, 32
incelf, 32
INCL, 5, 9, 10, 32
inclg, 32
incohr, 54
incut, 43, 186, 247, 248
infl, 3, 19, 23
info, 163, 196
infout, 7, 40
inmed, 35
inmedium, 35
inpara, 43, 252
input echo, 28, 44, 45
Intel Fortran, 6
interpolation, 146
inucr, 49
ionization potential, 39
ipara, 45
ipcut, 43, 186
ipegs, 5, 51
iphot, 50
iphter, 54
ipngdr, 36
ipnint, 5, 36, 51
ipout, 45
ippara, 43, 252
iprdst, 54
ipreeq, 36
iprofr, 54
iprtb2, 219
iraylr, 54
ireschk, 29, 256
irqmd, 7, 32
iscorr, 62
ismm, 35

iso, 68
isobar, 35
ispfs, 64
istdev, 9, 29, 190, 251
istrg, 50
isumtally, 169
itall, 19, 44, 166, 253
itetra, 46
itetvol, 46
itstep, 44, 131
iunr, 49
iunrst, 54
ivout, 45
ivoxel, 46, 48, 120
izst, 63, 129, 131

JAM, 6, 9, 32
JAMQMD, 2, 6, 32
jmout, 44, 220
JQMD, 7, 25, 32
JQMD-2.0, 4, 7, 32

K/X, 103
K/Y, 103
K/Z, 103
Kerma, 32
kf コード, 25, 160
kf-code, 25
kmout, 44, 99
KUROTAMA, 6, 10, 37
KX, 103
KY, 103
KZ, 103

l-dst, 233
l-phi, 233
l-the, 233
l-type, 157, 158, 205
Landau, 39
LAT, 46, 106, 107, 113, 115, 117–119, 122, 229
lattice, 9, 63, 106, 107, 113, 116, 117, 119, 126, 128, 132, 133, 139, 141, 142, 144, 145, 151, 152, 154, 155, 180, 230
lattice 座標, 114, 117, 119, 154, 155
LAT 番号, 164, 165
let, 189, 193, 205, 207, 208
letmat, 189, 205, 208
letmat1, 193
letmat2, 193
LET メッシュ, 157
level, 36
LIKE *n* BUT, 106, 116
line, 235
Lineal energy, 208
Linux, 13
lost particle, 47
lpolar, 54
Lynch の式, 39

Mac, 13
magnetic field, 47
makefile, 4, 17
MARS-PF, 28, 48
mass, 161, 196
MAT, 106, 107, 116
mat, 44, 98, 100, 127, 128, 135, 147, 148, 174, 175, 180, 181, 184, 185
matadd, 44, 106, 148
material, 143, 166, 171, 174, 186, 188, 189, 192, 193, 195, 198, 201, 202, 204, 205, 207, 208, 210, 211, 213, 214, 216, 221, 233
maxbch, 73
maxbnk, 29
maxcas, 29, 73, 251
MCNP, 44, 175, 181, 185, 247, 252
mdbatima, 39
mesh, 3, 4, 9, 114, 117, 120, 153–157, 160, 171, 172, 177–180, 186, 189, 190, 193–196, 198, 201, 202, 204, 205, 207, 208, 210, 211, 213, 214, 216, 219–221, 229–232, 240, 241
mgf, 128, 131
mirror, 233
Moliere 理論, 39
mother, 195, 198, 202, 214
mset, 147, 174, 175, 180, 181, 184, 185
multiplier, 9, 146, 147, 172, 174, 175, 178, 180, 181, 183–185, 221, 241

na, 7, 86
name, 127, 148, 150
NASA, 37
natural, 44
naz, 23

ncut, 43, 186, 252
ndata, 196
ndedx, 2
ne, 79, 81, 146
nedisp, 39, 47
negs, 6, 7, 36, 48, 51
neispl, 54
nevap, 35
nfile, 169
ni, 84
NLIB, 99
nlost, 47
nm, 8, 83
NMTC/JAM, 1
nn, 7, 8, 87
no ionization, 28
no reaction, 28
nocoh, 50
Node file, 121
non, 128, 150
nonu, 4, 49
norm, 85
NRF, 4
nspred, 6, 39, 47
ntetelem, 46
ntetsurf, 46
ntmax, 73, 75
ntt, 89
nuclear, 201, 216
nucleus, 135, 195, 196
nuclide, 84
num, 143
numb, 50
nwsors, 45

o-curr, 178
o-type, 89
oa-curr, 178, 179
ob-curr, 178
of-curr, 178
output, 4, 30, 68, 114, 163, 178, 179, 186, 190, 193, 201, 204, 211, 216, 229–233, 240
outtime, 219

P, 103, 115
p-type, 79, 81, 83
param.inc, 8, 20
ParaView, 2
part, 3, 4, 135, 139, 141, 142, 147, 151, 152, 160, 171, 174, 175, 177, 180–182, 184, 185, 188, 189, 193, 198, 202, 205, 208, 211, 214, 221
pcut, 43, 186, 252
PE, 19, 29, 43, 44, 46, 251, 252
PEGS5, 48
phi, 63, 65–67, 70–72
phits.in, 19, 251
photon, 25, 50, 144, 186
PLIB, 99
pnimul, 5, 36
point, 182
pwt, 144
PX, 103, 108–110, 112, 113, 115–119
px, 122
PY, 103, 108–110, 112, 113, 115–119
py, 122
PZ, 103, 108–110, 112, 113, 115–119
pz, 122
pz0, 69

q-type, 86, 87
q:, 23
qp:, 23

r, 161, 172, 177, 186, 190, 195, 201, 202, 207, 210, 211, 214
r-in, 179, 241
r-out, 179, 241
r-type, 156, 158
r-z, 4, 153, 156, 177, 180
r0, 65, 70
r1, 65, 67, 68, 70, 104, 105, 240
r2, 68, 70, 104, 105, 240
rad, 172
RCC, 104
rdam(i), 49
REC, 10, 104
reg, 3, 28, 63, 73, 75, 114, 117, 120, 126, 128, 131, 134, 139–142, 144, 145, 150–156, 161, 165, 171, 174, 177, 179, 180, 186, 188, 189, 192–195, 198, 201, 202, 204, 205, 207, 208, 210, 211, 213, 214, 216, 219–221, 231–233, 240, 241

region error, 47
resfile, 30, 162, 172, 178, 183, 186, 190, 194, 196, 201, 202, 207, 210, 211, 214, 221
resol, 166, 174, 178, 188, 192, 198, 201, 204, 207, 210, 213, 216, 222, 229, 231, 235
RHO, 106, 107
RHP, 104
rijk, 19
ring, 182
RIsource.dat, 84
rn, 66, 67
rnok, 50
RPP, 104
rpp, 122
rseed, 29
rshow, 28, 150, 164–166, 172, 188, 192, 198, 201, 204, 207, 210, 213, 216, 222
rx, 68
ry, 68
rz, 30, 161, 172, 186, 190, 196, 201, 202, 207, 210, 211, 214

S, 103
s-type, 61, 63, 65–72, 75, 78, 240
$S(\alpha,\beta)$, 100
SDM, 35
se-type, 210
se-unit, 208
sed, 208, 210
sfile, 169
Shen, 37
simple, 186, 204
size, 150
SO, 103
source, 201
SPAR, 3
special, 195
Specific energy, 208
SPH, 104
SQ, 103
star density, 214
straggling, 50
sumfactor, 169
sumtally, 1–6, 28, 169
surface, 102, 179
swtm(i), 34
SX, 103
sx, 36, 63, 72, 74, 75
SY, 103
sy, 36, 63, 72, 74, 75
symbol, 25
SZ, 103, 107, 109, 110
sz, 36, 63, 72, 74, 75

t, 161, 172, 177, 183, 201, 211, 221
t-e1, 161, 193
t-e2, 161, 193
t-eng, 161, 189
t-type, 73, 75, 88, 158, 171, 177, 179, 182, 189, 193, 198, 211, 221
t0, 88
target, 219
tc, 88
td, 88
Tetra.bin, 46
TFILE, 122
tg-list, 219
the, 161, 177, 201
tim, 141
timeevo, 219
title, 163, 172, 178, 183, 188, 190, 194, 196, 201, 204, 207, 210, 211, 216, 221, 229, 231, 235
tmax(i), 3, 34
TMP, 106, 107
tmp, 126
tn, 88
totfact, 62, 63, 73, 84, 240
transmut, 3, 216
TRC, 10, 104
TRCL, 63, 106, 107, 116, 123, 128, 167, 174, 179, 188, 192, 198, 201, 204, 207, 210, 213, 216, 222, 229, 231, 236, 237
trcl, 73, 131
trcle, 131
trclm, 131
trxcrd.dat, 6, 35, 39, 48
TSFAC, 122
tw, 88
TX, 103
TY, 103
typ, 128

TZ, 103

U, 63, 106, 107, 110, 112, 113, 115, 117–119, 122, 126, 128, 132, 133, 139, 141, 142, 144, 145, 151, 152, 154, 180
udtpara, 224
unit, 162, 172, 177, 179, 182, 186, 188, 190, 194, 195, 199, 202, 207, 210, 211, 214, 221, 240, 241
universe, 63, 107, 110, 113, 117, 119, 126, 128, 132, 133, 139, 141, 142, 144, 145, 151, 152, 154, 180
usrelst, 38
usrmgt, 38
usrsors.f, 61
usrtally.f, 224

Vavilov, 39
VOL, 106, 107
vol, 145, 155, 156, 174, 188, 192, 194, 198, 201, 204, 207, 213, 216
volmat, 166, 174, 188, 192, 198, 201, 204, 207, 210, 213, 216
volume, 45, 145, 155, 174, 188, 192, 194, 198, 201, 204, 207, 210, 213, 216, 222, 231, 240
voxel, 46, 119
vtkfmt, 174, 188, 192, 198, 201, 204, 213, 216, 229
vtkout, 174, 188, 192, 198, 201, 204, 213, 216, 229

w-ang, 233
w-dst, 233
w-hgt, 233
w-mnh, 233
w-mnw, 233
w-wdt, 233
wc1(i), 34, 142
wc2(i), 34
WED, 10, 104
weight cutoff, 142
weight window, 141
wem, 68
wgt, 72
width, 166, 174, 178, 188, 192, 198, 201, 204, 207, 210, 213, 216, 222, 229, 231, 235
Windows, 6, 13, 19, 24, 253
Wobbler magnet, 38, 128
wt0, 69
wwg, 221
wwi, 141

x, 161, 172, 177, 186, 190, 196, 201, 202, 207, 210, 211, 214
x-ray, 50
x-txt, 166, 172, 178, 188, 192, 194, 196, 201, 202, 207, 210, 211, 216, 221, 229, 231, 235
x-type, 157, 158, 165, 230, 232
x0, 65–72, 75, 156, 233, 240
x1, 65, 66, 68, 70–72, 75
x2, 71
x3, 71
xnum, 50
xp, 69
xq, 69
xsdir, 48, 99
XY, 103
xy, 30, 161, 164, 165, 172, 177, 186, 190, 196, 201, 202, 204, 207, 210, 211, 213, 214, 216, 221, 229, 231, 232
xyz, 9, 28, 153, 157, 164–166, 177, 180, 229, 231
xz, 165, 177, 202, 204, 207, 210, 221, 230

y, 161, 172, 177, 186, 190, 196, 201, 202, 207, 210, 211, 214
y-txt, 166, 172, 178, 188, 192, 194, 196, 201, 202, 207, 210, 211, 216, 221, 229, 231, 235
y-type, 157, 158, 165, 230, 232
y0, 65–72, 75, 156, 233, 240
y1, 65, 66, 68, 70–72, 75
y2, 71
y3, 71
yp, 69
yq, 69
yz, 161, 164, 165, 172, 177, 186, 190, 196, 201, 202, 204, 207, 210, 211, 213, 214, 216, 221, 229, 231

z, 161, 172, 177, 186, 190, 195, 196, 201, 202, 207, 210, 211, 214
z-txt, 166, 172, 178, 188, 192, 194, 196, 201, 202, 207, 210, 211, 216, 221, 231, 235
z-type, 156–158, 165, 230, 232
z0, 65–72, 75, 233, 240
z1, 65–68, 70–72, 75
z2, 71

z3, 71
zlin, 231
zlog, 231
ZP, 103
zx, 161, 165

異常終了, 19, 252
階層構造, 117, 139, 154, 156
核種, 25, 99, 100, 195, 198, 202, 214
核データ, 20, 44, 99, 195, 196
角度メッシュ, 158
吸収反応, 36
境界線, 162, 164, 165, 229, 231
切り替えエネルギー, 31
空白, 21–24, 27, 106, 154, 160
繰り返し幾何形状, 110, 115
群数, 79, 158, 159
計算打切エネルギー, 10, 31, 37, 43, 48, 186, 204, 205, 247
計算打切時間, 34
継続行, 159
最小値, 47, 79, 104, 158–160, 165
最大値, 79, 104, 158–160, 165
座標変換, 21, 102, 116, 123, 128, 167, 174, 179, 188, 192, 198, 201, 204, 207, 210, 213, 216, 222, 229, 231, 236
座標変換番号, 116
残留核, 35, 252
残留核の γ 崩壊, 35, 48
四重極電磁石, 128
質量密度, 99, 106
集合代数, 106, 107
詳細モデル, 50
時間カット, 34
時間メッシュ, 158, 198, 211
磁場, 37, 128
重イオン, 1
重力, 38
状態密度, 37
蒸発モデル, 35
生成核種, 195
セル, 106, 107
セル定義文, 105, 106, 114
セル番号, 106, 107, 111, 113, 116, 117, 119, 153, 154
体積, 145, 154–156, 166, 240
体積補正, 166
体積、面積計算, 28, 68, 240
タブ, 22
弾性散乱, 35, 49
中性子光学, 79, 129
中性子捕獲, 49
定数定義, 102, 106, 123
等高線, 164
二重極電磁石, 128
馬場, 36
標準出力, 18, 27, 48
標準入力, 18, 19
物質番号, 44, 98, 100, 106, 107, 110, 112, 116, 143, 147, 164, 165, 172, 174, 178, 180, 184, 188, 192, 198, 204, 213, 216, 222, 229, 231
分解能, 165
偏極率, 64, 130
密度, 8, 44, 99, 100, 106
メッシュ幅, 158–160
メモリ共有型並列, 17, 30, 251
面記号, 102, 103, 108, 113, 119
面積, 179, 240, 241
面定義数値, 102
面番号, 102, 105–107, 112, 113, 119
余弦, 65–69, 123
乱数, 29
粒子定義, 160
粒子密度, 44, 99, 106, 110
領域番号, 126, 139, 141, 142, 144, 145, 151–156, 164, 165, 179
領域メッシュ, 153

ソースのチェック, 28
ソース粒子のタリー, 28
ソースフアイル, 252
ダクトソース, 95
インクルード, 23, 24, 102, 106, 123
インクルードフアイル, 20
インストール, 13
ウエイトウインドウ, 34, 141
ウエイトカット, 34
データフアイル, 48, 252
エネルギーメッシュ, 157, 171, 177, 182, 198, 211, 214, 221
エネルギー分散, 39

カウンター, 74, 151, 166, 167, 174, 179, 183, 188, 192, 194, 198, 201, 204, 207, 210, 213, 216, 222
カラープロット, 164
クーロン散乱, 39
クラスタープロット, 164
バッチ, 19, 29, 44, 166, 251, 252
バンク, 20, 29
パラメーターセクション, 229, 231
パラメータセクション, 20, 165, 166
ヒストリー数, 29, 251
コメント文字, 23, 102, 106, 123
コンパイル, 13, 17, 61
スーパーミラー, 133
スコアリングメッシュ, 153, 156, 157
スピン, 64, 74, 130, 167
セルパラメータ, 106, 107, 110, 116, 126
セル番号, 150
ミューオン核反応, 7, 36
ミューオン捕獲反応, 3, 5, 6, 36
メッシュ定義文, 154, 156–158, 165
メッシュタイプ, 158
メモリ分散型並列, 17, 19, 29, 43, 45, 46, 168, 251–253
能, 136
ライブラリー, 28, 31, 99, 100, 214

位相, 128

核共鳴蛍光散乱, 5, 36
角度メッシュ, 171

光核反応, 7–10, 36

再開始計算, 9, 10, 29, 162, 172, 178, 183, 186, 190, 194, 196, 201, 202, 207, 210, 211, 214, 221, 251

時間メッシュ, 182, 221

統計誤差, 29, 161

連続四面体形状, 46, 121

本ソフトウェアおよび本マニュアルに関するお問い合わせは、PHITS 事務局 (phits-office@jaea.go.jp) までお願いいたします。

PHITS Ver.2.88 User's Manual

2013 年 11 月 19 日　Ver.2.64　発行
2014 年 01 月 09 日　Ver.2.64a　発行
2014 年 01 月 30 日　Ver.2.65　発行
2014 年 02 月 21 日　Ver.2.66　発行
2014 年 05 月 22 日　Ver.2.67　発行
2014 年 08 月 30 日　Ver.2.70　発行
2014 年 09 月 26 日　Ver.2.71　発行
2014 年 10 月 21 日　Ver.2.72　発行
2014 年 11 月 05 日　Ver.2.73　発行
2015 年 01 月 30 日　Ver.2.74　発行
2015 年 02 月 09 日　Ver.2.75　発行
2015 年 03 月 23 日　Ver.2.76　発行
2015 年 05 月 19 日　Ver.2.77　発行
2015 年 09 月 02 日　Ver.2.80　発行
2015 年 09 月 16 日　Ver.2.80a　発行
2015 年 10 月 15 日　Ver.2.81　発行
2015 年 12 月 16 日　Ver.2.82　発行
2016 年 03 月 03 日　Ver.2.83　発行
2016 年 03 月 16 日　Ver.2.84　発行
2016 年 05 月 16 日　Ver.2.85　発行
2016 年 08 月 23 日　Ver.2.86　発行
2016 年 09 月 16 日　Ver.2.87　発行
2016 年 09 月 29 日　Ver.2.88　発行

PHITS 開発チーム：
仁井田浩二 [1]，佐藤達彦 [2]，岩元洋介 [2]，橋本慎太郎 [2,†]，小川達彦 [2]，古田琢哉 [2]，
安部晋一郎 [2]，甲斐健師 [2]，松田規宏 [2]，岩瀬広 [3]，中島宏 [2]，深堀智生 [2]，奥村啓介 [2]，
甲斐哲也 [2], 千葉敏 [4]，Lembit Sihver[5]
[1](財) 高度情報科学技術研究機構，　[2](国研) 日本原子力研究開発機構
[3](共) 高エネルギー加速器研究機構，　[4](国) 東京工業大学，
[5] ウィーン工科大学, オーストリア
[†] Ver.2.30 より本マニュアルの編集を担当