

KEK Internal 99-5  
July 1999  
R/D

# **Lecture Notes of EGS4 Course at KEK (Japanese Part)**

(Last modified in 13 FEB 2003)

**H. Hirayama and Y. Namito**

**High Energy Accelerator Reserach Organization**

# Lecture Notes of EGS4 Course at KEK (Japanese Part)

H. Hirayama and Y. Namito

*High Energy Accelerator Reserach Organization  
1-1 Oho, Tsukuba-shi, Ibaraki, 305-0801 Japan*

# 目次

EGS4 コードシステムの概要	1
1 EGS4 コードシステム	1
2 EGS4 の構成とカスケードの追跡方法	3
3 EGS4 で使用されている主な変数	4
4 得たい情報の収集	4
4.1 エネルギー吸収量の計算	5
4.2 粒子束の計算	5
4.3 Track length の計算	5
4.4 ウェイトを使用する場合	5
5 EGS4 で使用されている配列の大きさ	5
Mortran3 の使い方	10
1 はじめに	10
2 基礎編	10
2.1 Statement	10
2.2 コメント	11
2.3 DO-loop	11
2.4 IF Statement	11
2.5 入出力文	12
2.6 Multiple Assignment	12
2.7 Label	13
2.8 Iteration	13
2.9 Operator	14
2.10 Mortran プログラムの一部を除く方法	14
2.10.1 NONGENERATE command	14
2.11 Mortran, FORTRAN リスト出力のコントロールコマンド	15
2.12 Mortran 使用時に発生しやすいエラーとその処理	16
3 応用編	17
3.1 マクロ	17
3.1.1 マクロの定義	17
3.2 EGS4MAC で使用されているマクロ	18
3.2.1 PARAMETER マクロ	18
3.2.2 COMIN マクロ	19
3.2.3 \$LGN (List-Generator) マクロ	20
3.2.4 \$RANDOMSET マクロ	20
3.2.5 デバッグのためのマクロ	21
3.2.6 The SPECIFY and \$SETINTERVAL#,#;	21
3.2.7 \$EVALUATE Macro	21

3.2.8	\$TRANSFERPROPERTIESTO#FROM#; . . . . .	22
3.2.9	ユーザが様々な機能を組み込む為に用意されているマクロ . . . . .	22
<b>4</b>	<b>EGS4MAC 等のユーザコードでの応用例</b>	<b>23</b>
4.1	EGS4 で使用されている mfp 等のパラメータの使用 . . . . .	23
4.2	NEGATIVE-USTEP . . . . .	23
4.3	LATCH の使用例 . . . . .	24
4.4	新しい属性の追加 . . . . .	24
4.5	Mortran 及び EGS4 実行時のエラーとその処理 . . . . .	25
4.5.1	Mortran エラーで原因が判らない時 . . . . .	25
4.5.2	Large negative“USTEP”が発生した時 . . . . .	25
4.5.3	Subroutine HATCH に関連したエラー . . . . .	25
4.5.4	配列オーバのエラーが発生した時 . . . . .	26
4.5.5	LOOP が発生してヒストリーが終了しない場合 . . . . .	26
	<b>PEGS4 の使い方</b>	<b>31</b>
1	元素用入力ファイル	31
2	化合物用入力ファイル	33
3	混合物用入力ファイル	34
4	CALL オプション	35
5	国内で改良した PEGS4 の使用方法	35
	<b>English Part</b>	<b>37</b>
1	Input file for element	37
2	Input file for compound	39
3	Input for a mixture	40
4	CALL Option	41
5	How to use a PEGS4 improvement developed in Japan	42
	<b>ユーザコードの書き方</b>	<b>45</b>
1	はじめに	45
2	STEP 1	45
3	STEP 2	47
3.1	オプションの設定 . . . . .	48
4	STEP 3	49
5	STEP 4	50

6	STEP 5	50
7	STEP 6	50
8	STEP 7	51
8.1	統計誤差	51
8.1.1	MCNPで使用している方法	51
8.1.2	MORSE-CGで使用している方法	52
8.1.3	統計誤差評価のために行っている ucna3p.mor での処理	52
9	STEP 8	53
10	SUBROUTINE AUSGAB	54
11	SUBROUTINE HOWFAR	55
	EGS4におけるジオメトリーの扱い	56
1	SUBROUTINE HOWFARの機能	56
2	\$PLANE1 マクロ	56
3	\$PLAN2P マクロ	58
3.1	Multislab 形状	59
3.1.1	半無限平板形状	59
3.1.2	有限平板形状	60
3.1.3	3-Dimensional Cartesian Geometry	61
4	\$CYLINDR 及び \$CYL2 マクロ	65
4.1	円筒平板形状	66
4.2	DELCYL マクロ	67
5	\$SPHERE 及び \$SPH2 マクロ	68
6	円錐	69
7	\$FINVAL マクロ	69
8	ジオメトリーマクロの修正、作成	70
9	標準のジオメトリーの一部変更	71
	線源ルーチンの作り方	73
1	サンプリング法	73
1.1	Direct method	73
1.2	Combination of of “composition” and “rejection” techniques	73
1.3	von Neumann method	74

<b>2</b>	<b>線源ルーチン</b>	<b>74</b>
2.1	線源位置の決定	75
2.1.1	線線源	75
2.1.2	半径 $R_0$ と $R_1$ の領域に一様に分布した面線源	75
2.1.3	直円筒の側面に平行ビームが当たる場合	77
2.1.4	円筒体積線源	77
2.2	方向余弦の決定	78
2.2.1	点等方線源	78
2.2.2	コサイン分布	81
2.3	エネルギーの決定	82
2.3.1	内挿法による決定	82
2.3.2	Rejection 法による決定	82
2.3.3	ウエイトを用いる方法	83

# EGS4 コードシステムの概要

## 1 EGS4 コードシステム

EGS4 コードシステムと其中で扱っている物理現象は以下に示す。

- EGS4 コードシステムは、任意の元素、化合物あるいは混合物中の電子、陽電子あるいは光子の輸送を扱う事ができる。EGS4 で使用する各物質の諸データは、原子番号 1 から 100 までの元素の断面積データを用いて PEGS4(A Preprocessor for EGS4) により計算される。
- 光子、荷電粒子共に、離散的なステップではなく、ランダムな輸送として扱われる。
- 荷電粒子の適用エネルギー範囲は、運動エネルギーで数 keV から数千 GeV である。上限値については、更に高いエネルギーまで拡張できる可能性があるが、そのためには物理的なモデルの妥当性を調べる必要がある。
- 光子の適用エネルギー範囲は、1keV から数千 GeV である。(上記の注参照)
- EGS4 では、以下の物理現象が扱われている。
  - Bremsstrahlung production (excluding the Elwert correction at low energies).
  - Positron annihilation in flight and (the annihilation quanta are followed to completion).
  - Molière multiple scattering (*i.e.*, Coulomb scattering from nuclei). The reduced angle is sampled from a continuous (rather than discrete) distribution. This is done for arbitrary step sizes, selected randomly, provided that they are not so large or small as to invalidate the theory.
  - Møller ( $e^-e^-$ ) and Bhabha ( $e^+e^-$ ) scattering. Exact rather than asymptotic formula are used.
    - Continuous energy loss applied to charged-particle tracks between discrete interactions.
      - \* Total stopping power consists of soft bremsstrahlung and collision loss terms.
      - \* Collision loss determined by the (restricted) Bethe-Bloch stopping power with Sternheimer treatment of the density effect.
  - Pair production.
  - Compton scattering.
  - Coherent (Rayleigh) scattering can be included by means of an option.
  - Photoelectric effect.
    - \* Neither fluorescent photons nor Auger electrons are produced or transported in the default version of subroutine PHOTO.
    - \* Other *user-written* version of PHOTO can be created, however, that allow for the production and transport of K- and L-edge photons.
- PEGS4 は、12 のサブルーチンと 85 の関数からなるスタンドアロンのプログラムである。その出力は直接 EGS4 で読み込む形になっている。

- PEGS4 は、広いエネルギー範囲の断面積、分岐比等のデータを piecewise-linear fit したパラメータの形で出力する。
- 条件が同じであれば、PEGS4 の計算は一度で良い。
- PEGS4 の入力データには、FORTRAN の NAMELIST を使用している。
- EGS4 で使用するデータを作り出す以外に、PEGS4 は、EGS4 で使用される物理量のプロットや USTESTSR というユーザコードを使用してサンプリングした結果と理論値を比較する目的に使用する事もできる。
- EGS4 は、融通性のあるユーザインターフェイスを持ったサブルーティンとブロックデータで構成されるパッケージである。ユーザインターフェイスと EGS4 の関係を第 1 図に示す。
  - Mortran3 のマクロ機能と併せて、このような構造がユーザがコードにバグを持ち込むのを防ぐ事ができる。
  - EGS4 は、PEGS4 によって作られた断面積や分岐比等のデータを使用する。
- それぞれの問題のジオメトリーは、ユーザが書く HOWFAR という名のサブルーティンで記述する。記述には、様々な補助サブプログラムを使用する事ができる。
  - 自分自身でジオメトリルーティンを書くことを望まないユーザに対しては、平板、円筒、球等の補助ジオメトリルーティンが EGS4 システムに含まれている。
  - サブルーティンよりは一般的に実行速度の早いマクロバージョンも用意されている。(i.e., in the EGS4MAC file)
  - HOWFAR 用の MORSE-CG Combinatorial Geometry パッケージも使用可能である。(e.g., see the UCSAMPCG file on the EGS4 Distribution Tape). しかし、CG を使用するとマクロの場合に比べて少なくとも 4 倍以上計算速度が遅くなる。
  - 特別な HOWFAR により磁場中の輸送を扱うことができる。あるいは Mortan3 のマクロを使用してサブルーティン ELECTR を書き換える事により、電場を含んだ磁場をもっと一般的に扱う事ができる。
- ユーザの計算したい情報は、AUSGAB と呼ばれるユーザの書くサブルーティンで処理する。
  - エネルギー保存のチェックやその他のために使用するサブルーティン ECNSV1 も提供されている。
  - サブルーティン NTALLY は、トラック数の情報 (i.e., an event counter) のために提供されている。
  - event-by-event あるいは step-by-step のトラッキングのためにサブルーティン WATCH が提供されている。
- EGS4 には、*importancesampling* 等の “variance reduction techniques” (e.g., leading particle biasing, splitting, path length biasing, Russian roulette, etc.) を組み込む事ができるようになっている。
- 線源条件:
  - オプションとして  $\pi_0$  の崩壊に伴い発生する 2 個の光子を線源とする事ができる。(IQI=2)
  - ユーザは、単一エネルギーの粒子からでも分布したスペクトル (例えば、シンクロトロン放射光) からサンプリングした結果から輸送を始める事ができる。
  - 空間分布や角度分布を持った線源から輸送を始める事もできる。



## 2 EGS4の構成とカスケードの追跡方法

EGS4flow control を第 1 図に示す。EGS4 で計算を行うためには、

### 1. メインプログラム

- (a) 計算形状の定義、
- (b) 使用する物質データの定義と各リージョンへの割当、
- (c) 入射粒子に関する初期データの設定等、
- (d) SUBROUTINE HATCH を CALL することにより PEGS4 で計算しておいた物質データの読み込み、
- (e) SUBROUTINE SHOWER を必要なヒストリー回数 CALL し、
- (f) 得られた結果を編集して出力する。

### 2. ジオメトリを決定する SUBROUTINE HOWFAR、

### 3. 求めようとする結果を蓄積する SUBROUTINE AUSGAB

で構成されるユーザコードが必要である。(これ以外の SUBROUTINE がユーザコードに含まれても良いが、HOWFAR と AUSGAB は、必ずなければならない。無限媒質でジオメトリを考慮する必要が無い場合にも、RETURN; END; だけで構成される HOWFAR が必要である。)

EGS4 のジオメトリの単位は、リージョンである。各リージョンの物質は、メインプログラムで HATCH を CALL する前に行われる。HOWFAR は粒子が移動するたびに CALL され、メインプログラムで定義した形状に関するデータによって、リージョンが変わるかどうかのチェックを行う。(詳細は、「How to Code Geometry in EGS4」を参照) AUSGAB は、標準では、

1: 粒子が移動する前 (IARG=0)

2: 粒子のエネルギーが PEGS4 で設定したカットオフエネルギー以下になった時 (IARG=1)

3: 粒子のエネルギーが EGS4 で設定したカットオフエネルギー以下になった時 (IARG=2)

4: ユーザーが追跡終了の設定をしたとき (IARG=3)

5: 光電吸収の生じる前 (IARG=4)

に CALL される。各反応の生じた割合を知りたい場合等には、その反応に関連する IAUSFL(IARG) の値を 1 にする事により、知りたい反応の発生前または発生後に AUSGAB を CALL するようにする事ができる。

第 1 表に、EGS4 で使用されている IARG の値の意味を示す。

電磁カスケードでは、反応毎に粒子の数が倍に増える。粒子の数の増加をどの様に扱っていくかは計算コードにより異なるが、EGS4 ではスタック数、NP、を用いて次のような方法でカスケードを追跡している。

1. 入射粒子のスタック数を 1 とする。( NP=1; )

2. 反応の結果新しい粒子が発生すると、全エネルギーの小さい方の粒子にスタック数 NP+1 を、大きい方の粒子に NP を割り当てる。

3. スタック数の大きい粒子を先に追跡する。

4. 粒子が、体系外に出たり、カットオフエネルギー以下なる等の理由で discard された場合には、スタック数 NP=NP-1 の粒子を次に追跡する。( 第 2 図参照 )

5. スタック数 NP=1 の粒子が discard された時、そのヒストリーが終了したとする。

### 3 EGS4で使用されている主な変数

EGS4で使用されている変数とその意味については、SLAC-265のAPPENDIX 2で説明されている。ユーザコードを書く上で最低必要と思われる変数とその意味を以下に示す。

#### COMMON/STACK

X(NP), Y(NP), Z(NP) :粒子の位置  
U(NP), V(NP), W(NP) :粒子の方向余弦  
WT(NP) :粒子の重み  
E(NP) :粒子のエネルギー ; 静止質量を含む全エネルギー (MeV)  
IQ(NP) :粒子の電荷  
IR(NP) :粒子のリージョン  
NP :スタック数

#### COMMON/BOUNDS

ECUT :各リージョンでの荷電粒子のカットオフエネルギー  
PCUT :各リージョンでの光子のカットオフエネルギー

#### COMMON/EPCONT

EDEP :吸収エネルギー (MeV)  
TSTEP :反応点までの直線距離  
TVSTEP :移動する実際の距離  
IROLD :前のリージョン  
IRNEW :次のリージョン

#### COMMON/MEDIA

NMED :物質の数 ; ユーザコード中で使用する物質の数  
RLC :物質の radiation length  
RHO :物質の密度

メインプログラムで粒子の初期値は次のように設定し、CALL SHOWERの引き数としてSUBROUTINE SHOWERに渡す。SUBROUTINE SHOWER内でこれらの初期値は、スタック数1の各パラメータに割り振られる。

XI, YI, ZI :入射粒子の位置  
UI, VI, WI :入射粒子の方向余弦  
EI :入射粒子のエネルギー  
IQI :入射粒子の電荷  
IRI :粒子の入射リージョン  
WTI :入射粒子の重み

### 4 得たい情報の収集

EGS4では、ユーザが自分で得たい情報をSUBROUTINE AUSGABで変数や配列に割当て、所定のヒストリーが終了したのちに、集計・出力するという思想を持っている。予め、著者が設定した限りの情報を入力データで選択する方式のコードと異なり、自分の計算目的を明確にし、その目的にあうようにAUSGABを書かなければならないので特に初心者には使いにくいという印象を与えるが、自分の目的とする情報の物理的意味を知った上でコーディングすることは当然のことである。そのためには、どのような変数を使えばよいかという事を知っている必要がある。具体的には、サンプルプログラムを参照してもらうとして、ここでは最低知っておく必要のある基本的な事を

紹介する。

#### 4.1 エネルギー吸収量の計算

最も一般的な物理量は、エネルギー吸収量である。粒子の移動や反応に伴って物質に吸収されるエネルギーは、変数 EDEP である。この量をどのような形で収集するかは計算毎に異なる。例えば、検出器の検出効率を計算している場合や、高エネルギーのカロリメータでは、エネルギー吸収が検出器部に相当するリージョンで発生した時のみに、所定の変数に加えていく。物質中のエネルギー分布の状況を計算している場合には、エネルギー吸収量をリージョン毎に加える。

#### 4.2 粒子束の計算

リージョン境界の面を横切る粒子束を計算するには、まず粒子が面を横切ったかどうかの判断をし ( `IF(IR(NP).NE.IROLD.AND.IARG.EQ.0)` )、その上で、粒子が自分の目的とする粒子であるかどうかの判断をし、所定の配列に加えていく。粒子の横切った面がどの面かについては、ユーザが自分の使用している形状に即して判断する。Z-軸に垂直な無限平板が組み合わされた形状の場合も粒子の進行方向によって同じリージョンにいる場合でも横切る面が異なることに注意する必要がある。光子束の計算の場合には、通常的光子輸送計算コードで行っているように、面に対する粒子の角度を補正する (平面の場合は、その面に対する方向余弦で割る)。面を横切る光子の数を求める場合には、このような補正は行わない。エネルギースペクトルを計算する場合には、更に、粒子の運動エネルギーによって、エネルギー区分を決定し、該当する区分に割り当てる作業が必要である。

#### 4.3 Track length の計算

計算する量が、track length に比例する場合や、光核反応生成量を計算する場合には、粒子の実際の移動距離 TVSTEP を使用する。エネルギー依存の track length の場合には、エネルギースペクトルの場合と同様にエネルギー区分毎の track length を求める。

#### 4.4 ウェイトを使用する場合

通常の計算では、粒子のウェイト WT(NP) は常に 1 であるが、線源条件にウェイトを用いたり、なんらかのバイアス法のより、粒子にウェイトを割り当てる場合には、WT(NP) を掛けた量を保存する必要がある。間違いをなくすためにも、常に WT(NP) を掛けた量を扱うようにする方が良い。

### 5 EGS4 で使用されている配列の大きさ

EGS4 で使用されている配列の大きさの標準値は、EGS4MAC.MORTRAN で次の様に設定されている。

```
$MXMED — 10; ユーザコード中で使用する物質の数  
$MXREG — 2000; リージョン数  
$MXSTACK — 40; スタック数 NP の最大値
```

通常は、これらの大きさを変更する必要は無いが、Mortran 使用の手引きで詳しく紹介されている様に、ユーザコードでマクロを用いて変更する事ができる。

ジオメトリーマクロを使用する場合の平板数等に関連する配列の大きさも次のように標準値が決められており、同様にユーザコードで変更する事ができる。

\$MXPLNS — 100; 平板の最大数

\$MXCYLS — 75; 円筒の最大数

\$MXSPHE — 75; 球の最大数

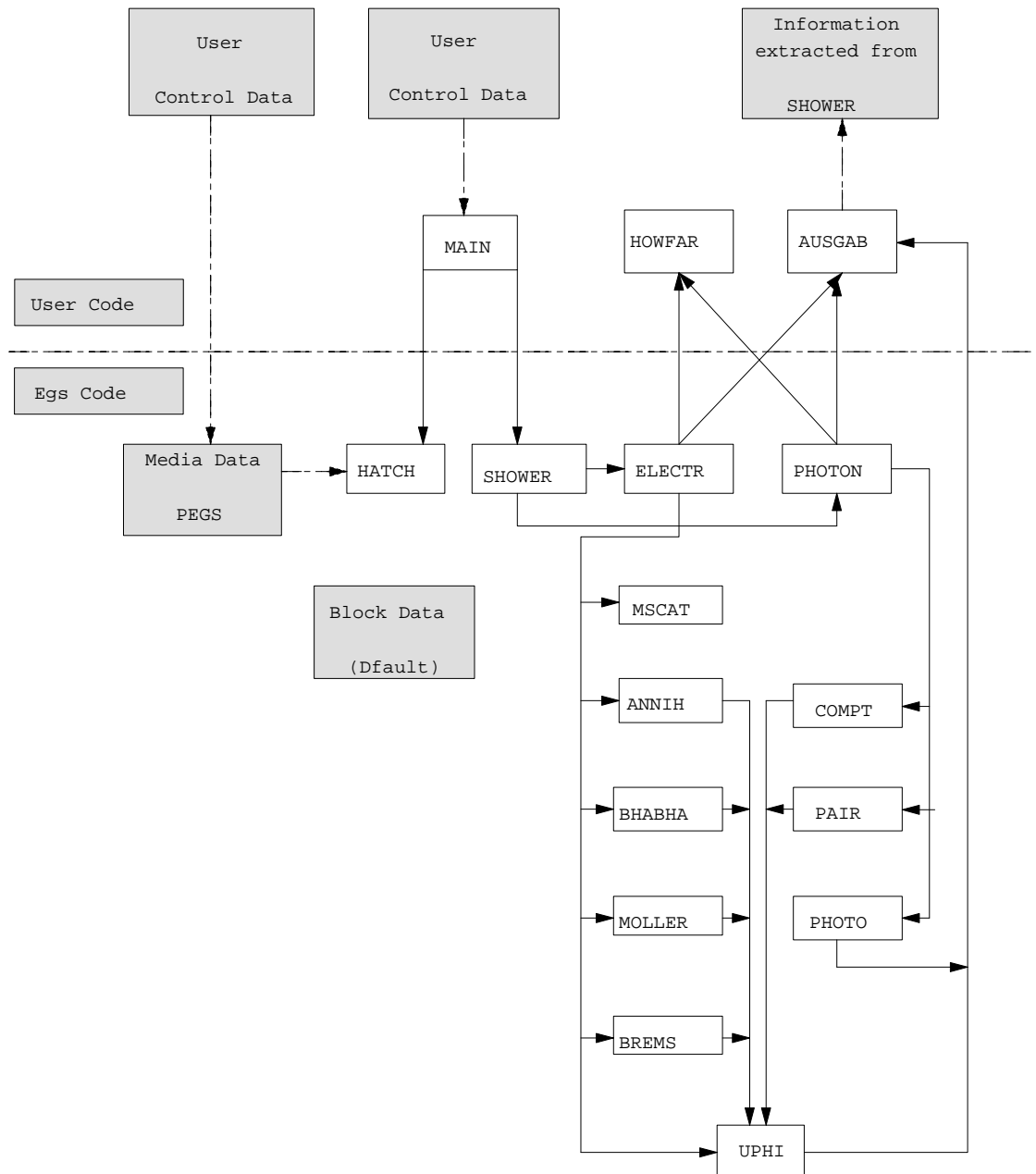
\$MXCONES — 75; 円錐の最大数

**Table 1(a) The value for IARG and corresponding situation.**

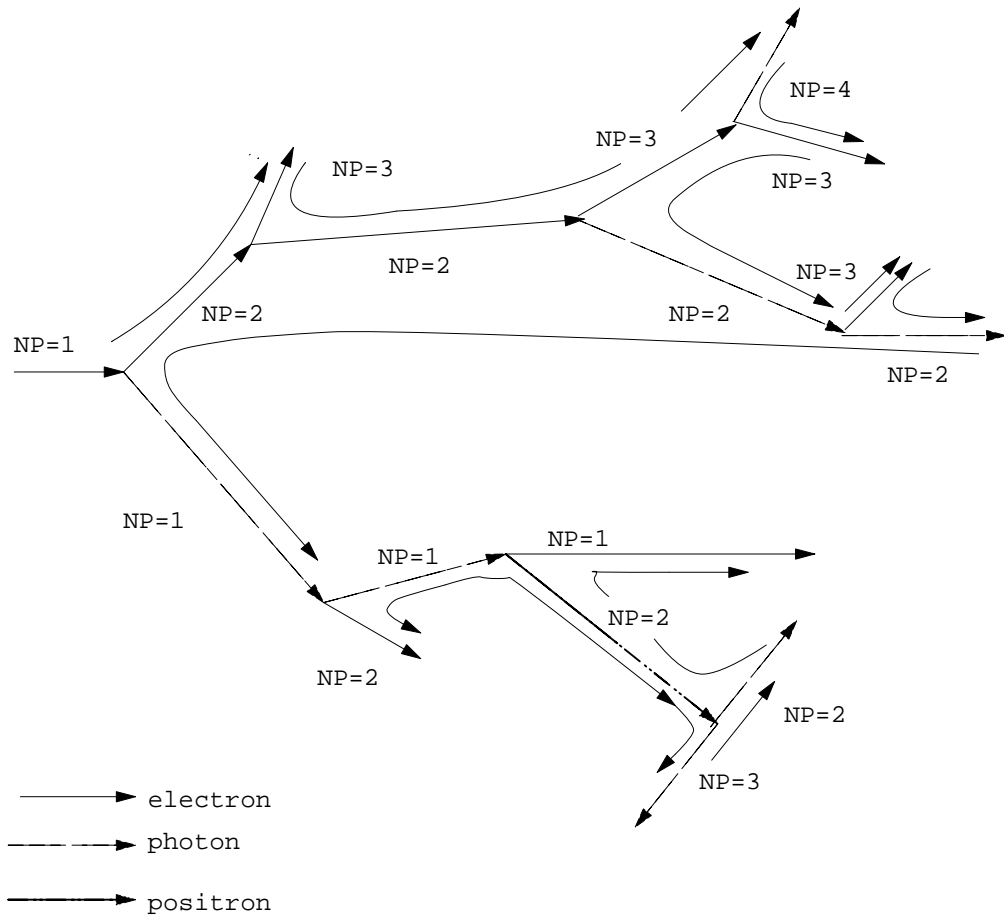
IARG	IAUSFL	Situation
0	1	Particle is going to be transported by distance TVSTEP.
1	2	Particle is going to be discarded because its energy is below the cutoff ECUT (for charged particles) or PCUT (for photons)—but its energy is larger than the corresponding PEGS cutoff AE or AP, respectively.
2	3	Particle is going to be discarded because its energy is below both ECUT and AE (or PCUT and AP).
3	4	Particle is going to be discarded because the user requested it (in HOWFAR usually).
4	5	A photoelectric interaction has occurred and either: a) the energy of the incident photon was below the K-edge binding energy and it is going to be discarded, or b) a (fluorescent) photon is going to be discarded with the K-edge binding energy.

**Table 1(b) The value for IARG and corresponding situation.**

IARG	IAUSFL	Situation
5	6	Particle has been transported by distance TVSTEP.
6	7	A bremsstrahlung interaction is to occur and a call to BREMS is about to be made in ELECTR.
7	8	Return to ELECRA after a call to BREMS was made.
8	9	A Møller interaction is to occur and a call to MOLLER is about to be made in ELECTR.
9	10	Return to ELECRA after a call to MOLLER was made.
10	11	A Bhabha interaction is to occur and a call to BHABHA is about to be made in ELECTR.
11	12	Return to ELECRA after a call to BHABHA was made.
12	13	An in-flight annihilation of positron is to occur and a call to ANIHI is about to be made in ELECTR.
13	14	Return to ELECRA after a call to ANIHI was made.
14	15	A positron has annihilated at rest.
15	16	An pair production interaction is to occur and a call to PAIR is about to be made in PHOTON.
16	17	Return to PHOTON after a call to PAIR was made.
17	18	A Compton interaction is to occur and a call to COMPT is about to be made in PHOTON.
18	19	Return to PHOTON after a call to COMPT was made.
19	20	A photoelectric interaction is to occur and a call to PHOTO is about to be made in PHOTON.
20	21	Return to PHOTON after a call to PHOTO was made (assuming NP is non-zero).
21	22	Subroutine UPHI was just entered.
22	23	Subroutine UPHI was just exited.
23	24	A coherent (Rayleigh) interaction is about to occur.
24	25	A coherent (Rayleigh) interaction has just occurred.



☒ 1: Division between the user-interface and EGS4.



☒ 2: Flow control of cascade in EGS4.

# Mortran3の使い方

## 1 はじめに

Mortran は、SLAC の J. Cook 等によって FORTRAN をより機能的にする目的で開発された FORTRAN の Pre-processor である。EGS4[1] で Mortran を採用しているのは、主に Mortran が Macro 機能を持っているからであるが、それだけでなく Mortran の機能を使用し EGS4 や PEGS4 をコンパクトで見やすくする事も要因の一つとなっている。一方、EGS4 のユーザーからは、EGS4 が Mortran を使用している為に使いにくいという意見が出されている。この原因としては、Mortran の使用経験が無い上に、計算機センター等でも Mortran に対してサポート体制が無いことによるものと考えられる。

Mortran の使用に必要な事は、EGS4 のマニュアルである SLAC-265 の APPENDIX 4 に “EGS4 Users Guide to Mortran3” として書かれている。しかしながら、EGS4 を使用する前に読んで理解しなければならぬという点では実際上、APPENDIX 4 だけでは判りにくいのも事実であると言える。この手引では、基礎編では EGS4 の USER CODE を Mortran で書く上で最低限度知っていなければならない Mortran 3 の文法と発生し易いエラーとその対処法を紹介し、応用編で Macro の機能を紹介すると共に USER CODE で使用できるように EGS4 システムで使用している Macro についても併せて紹介する。

Mortran は、使用する機会が少ないことから最初は敬遠しがちであるが、環境さえ整備されるならば FORTRAN よりはるかに使いやすく便利である。(自分なりに秩序を立てて割り振ってきた文番号を、プログラムの修正時にその秩序を壊さざるを得なかった経験をもっている人は多いと思う。文番号を全く考えなくて良いということだけでも使う価値があると思う。) EGS4 を初めて使用する時、あるいはより高レベルの使い方をする際の有効な手引となる事を願う。

EGS4 ユーザに対する Mortran の手引きとして、1992 年に “Mortran3 使用の手引き” が KEK Internal 92-8 として出版された。KEK Internal 92-8 は、1990 年から開催されてきた EGS4 研究会のテキストとして使用されてきたが、Mortran3 を理解し、より効率的に使用する上で説明の追加が必要になってきた。また、KEK Internal 92-8 出版後も、EGS4 の様々な改良がマクロの形で行われており、EGS4 の利用に関する手引きとしてはこれらに関する記述も求められている。

## 2 基礎編

### 2.1 Statement

Mortran の Statement は ; で終了する。

Statement は行のどこから始めても良いし、1行で終わる必要も無い。1行に Statement をいくつ書いても良い。

Statement の最後に ; を付けなければならない事は、Mortran を使用始めた時に最も忘れ易い事である。FORTRAN の Statement は、最後に ; をつけ、継続行の印を除けば Mortan の Statement として認識される。



## 2.2 コメント

”で囲まれた部分は Mortran ではコメントと認識される。

コメントは、プログラムのどこにでも書くことができる。コメントは複数行にわたって使用する事が可能だが、応用編で述べる %Q1 が使用された以降では、コメント行の終わりの ”を省略できる代わりに複数行にわたるコメントは使えなくなる。このコメントが Statement の途中でも簡単に使える機能は、プログラムを分かりやすく書く点で非常に便利である。自分自身にとっても他のユーザが使う場合でも、コメントが多いものほど判りやすい。

## 2.3 DO-loop

DO-loop は DO I=J,K,N [...] と表現すればよい。

文番号と CONTINUE 文は、Mortran が割り振ってくれる。ブロックを示す為に [ と ] が使用される。J, K, N の意味は FORTRAN の DO-loop と同じである。DO-loop は次の例のように多重に使う事ができる。

```
DO I=1,10 [DO J=1,20 [DO K=1,5 [A(I,J,K)=I*J*K;]]]
```

[ の数と ] の数が一致する様に注意する。

## 2.4 IF Statement

基本となる IF Statement は

```
IF e [...;]
```

である。e は任意の論理表現で、ブロック内には、複数の Statement が書かれてもよい。論理表現は、( ) で囲んでも良いし囲まなくても良い。例えば、

```
IF A.LT.B [ C=D; E=F; ] G=H; でも IF(A.LT.B) [ C=D, E=F; ] G=H; でも良い。
```

IF-ELSE の表現は、

```
IF e [...;] ELSE [...;]
```

と表される。FORTRAN で必要な THEN や ENDIF は書く必要がない。

IF-ELSEIF の表現は、

```
IF p [...] ELSEIF q [...];] ELSEIF r [...] ELSE [...];
```

で表される。DO-loop の場合と同様にブロックを示す記号 [ ] が重要であり、対応が取れている様に注意する。

## 2.5 入出力文

標準的な FORTRAN の入出力文 ( READ(5, ) と WRITE(6, ) はつぎの様に簡単に表現できる。

```
INPUT i-o list; (format list); READ(5, に対応  
例 : INPUT NCASES;(I6);  
OUTPUT i-o list; (format list); WRITE(6, に対応  
例 : OUTPUT IXX;(' IXXST=',I12);
```

FORMAT 文の文番号や FORMAT という言葉は不要である。これらも FORTRAN にコンパイルされる時点で Mortran が設定する。INPUT や OUTPUT を使用する場合には、format list が常に必要である。

その他の入出力 UNIT を使用する場合には、

```
READ(1,;FORMAT1:) i-o list; ;FORMAT1:FORMAT(format list);  
WRITE(7,;FORMAT7:) i-o list;;FORMAT7:FORMAT(format list);
```

の様にすれば良い。:FORMAT1:, :FORMAT7: 等は例であり、ラベルとしては何を使用しても良い。FORTRAN の場合と同様に、一つのプログラム単位内で同じ FORMAT 文を何回も使用する場合には FORMAT 文は一度書くだけでよい。当然の事ながら、READ(5, や WRITE(6, の場合にもこの方法を使用しても良い。

## 2.6 Multiple Assignment

```
/p,q,r,.....,z/=e;
```

Mortran では、上記の式で複数の変数を一度に同じ値や変数に設定する事ができる。各変数の型が違っていてもかまわない。例えば、

```
/I, A(I,K), J/ = SQRT(X/2.0);
```

は、次のような FORTRAN Statements に変換される。

```
I = SQRT(X/2.0)  
A(I,K) = SQRT(X/2.0)  
J = SQRT(X/2.0)
```

この機能は、メインプログラムで様々な変数を初期化する際によく使用される。

## 2.7 Label

```
: で囲まれたものは label と認識される。  
例 :USER-PHOTON-DISCARD:.....;
```

Label の文字数の制限は無い。従って、label 名としては、できるだけ判りやすい名前をつける方がよい。Label は、FORTRAN に変換された時点で文番号に置き換えられる。

## 2.8 Iteration

DO-LOOP 以外にもいくつかの iteration 法が使用できる。

```
WHILE e [.....;]  
LOOP [.....;] WHILE e;  
UNTIL e [.....;]
```

最初の例では、論理表現  $e$  が最初に調べられ、真の時はブロック内の Statement を実行し、偽の時はブロックの次の最初の Statement に制御を移す。LOOP の場合は、ブロック内の Statement を実行した後に論理表現  $e$  を調べる点が最初の例と異なっているだけで他の機能は同じである。UNTIL 表現は、WHILE の逆で  $e$  が真になるまでブロック内の Statement が実行される。WHILE と UNTIL は次の例の様に組み合わせて使用する事もできる。

```
WHILE e [.....;] UNTIL f;  
WHILE e [.....;] WHILE f;  
UNTIL e [.....;] WHILE f;  
UNTIL e [.....;] UNTIL f;
```

次の様な FOR-LOOP も使用できる。

```
FOR v=e TO f BY g [.....;]  
FOR v=e BY g TO f [.....;]  
FOR v=e TO f [.....;]
```

$v$  は制御変数であり、 $e, f, g$  は任意の数値表現である。最後の表現の場合には  $g=1$  として扱われる。特殊な iteration 表現として次のような "forever loop" が用いられる場合がある。

```
LOOP [.....;] REPEAT;  
LOOP [.....;]
```

この様な "forever loop" から抜け出すには

```
GO TO :CHICAGO;;
EXIT;
IF (e) EXIT;
IF e [.....; EXIT;]
```

等の方法がある。最初の方法は、loop の外の :CHICAGO: というラベルを持つ Statement に強制的に制御を移す。他の場合には、loop の次の Statement に制御が移される。

## 2.9 Operator

Relationnal operator としては、FORTRAN で使用される .LT., .EQ., 等の他に次の様な記号を使用する事ができる。

```
<, <=, =<, =, ~=, =>, >=, >
```

また、logical operator としても .AND., .OR. 等の他に

```
<, < , & (and), | (or), ~ (not)
```

という記号を使用する事ができる。しかし、一つの Statement 内で FORTRAN で使用される表現と混合して使用してはならない。例えば (A>B & C>D) や (A.GT.B.AND.C.GT.D) は良いが、(A>B.AND.C>D) の様な使い方は出来ない。

## 2.10 Mortran プログラムの一部を除く方法

一時的に、Mortran プログラムのある部分を除くには、以下の 2 つの方法がある。

- 最も一般的な方法 --- ”記号で囲んで当該部分をコメントとする。
- 他の方法 (マニュアルには記載されていない) --- NONGENERATE コマンドを用いる。

### 2.10.1 NONGENERATE command

NONGENERATE 及び GENERATE コマンドを用いる事により、Mortran プログラムの一部をプログラムから除いたり、含めたりする事ができる。

```
!NEWCONDITIONAL; "Place at top of User Code"
```

```
NONGENERATE; "Remove this code"  
(lines of code)  
ENDGENERATE;
```

```
GENERATE; "Use this code"  
(lines of code)  
ENDGENERATE;
```

上記の例では、NONGENERATE と ENDGENERATE の間の行は、Mortran プログラムから除かれる。  
下記の例のようにすれば、プログラムに含める部分を簡単に変更することができる。

```
$Select_Code_NEW; "Use NEW or OLD code"

REPLACE{$Select_Code_#;} WITH {
  [IF] '{P1}'='NEW' [SET <NewCode>=GENERATE;
                    SET <OldCode>=NONGENERATE;]
  [IF] '{P1}'='OLD' [SET <NewCode>=NONGENERATE;
                    SET <OldCode>=GENERATE;]
}

<OldCode>
(lines of code)
ENDGENERATE;

<NewCode>
(lines of code)
ENDGENERATE
```

この例のように \$Select\_Code\_NEW が定義されると、<OldCode> と ENDGENERATE の間の行が除かれ、逆に \$Select\_Code\_OLD が定義されると、<NewCode> と ENDGENERATE の間の行が除かれる。

## 2.11 Mortran, FORTRAN リスト出力のコントロールコマンド

Mortran や FORTRAN のリスト出力等をコントロールするコマンドにはいろいろなものがあるが、次のコマンドはよく使われるので知っておいた方がよい。% で始まるコマンドは必ず 1 カラムから書かなければならないが、! で始まり ; で終わるコマンドはどの位置に書いてもよい。

**%E** : Mortran のリスト出力において改頁を行う。

**%L** : または !LIST; Mortran のリストを出力する。

**%N** : または !NOLIST; Mortran のリストを出力しない。

**%In** : または !INDENT Mn; Mortran のリストでネスト毎に n 文字分インデントをつける。

**%F** : FORTRAN モードに切り換える。ユーザコードを FORTRAN で書きたい場合には、%F を使えば良い。また、Mortran の Statement と FORTRAN の Statement の混じったプログラミングをすることも次の %M と併せて使用することにより可能である。<sup>1</sup>

**%M** : Mortran モードに戻す。

**!INDENT Fn**; : FORTRAN リストにおいて、ネストレベル毎に n カラムのインデントを行う。

**!COMMENTS**; : Mortran のコメントを FORTRAN のコメントとして FORTRAN リストに出力する。

**!NOCOMMENTS**; : コメント出力を止める意味である。

**%An** : Mortran コメントを FORTRAN の注釈として出力するモードの切り替え。

<sup>1</sup>但し、%F にはバグがあり、%F の直前の Mortran Statement が正しく変換されない。従って、FORTRAN と Mortran を混合して使用する場合には、%F の前に "....."; の様なダミーのコメント行を入れる。通常コメントの後ろには ; は不要であるが、この場合には必ずつけなければならない。

n = 0 FORTRAN リストに出力しない。!NOANNOTATE; も同じ意味。

n = 1 Mortran コメントを FORTRAN のコメントとして 2 から 80 カラムに出力する。!ANNOTATE; も同じ意味。

n = 2 Mortran コメントを FORTRAN のコメントとして 40 から 80 カラムに出力する。1 行の Mortran コメントは 2 行に出力される。

%Qn : Mortran のコメントのコントロール ( 初期値は 0 )

n = 0 コメントは必ず ” で囲まなければならない。 ” で囲めば複数行にわたるコメントも可能である。

n = 1 行の終わりのコメントの最後の ” は無くても良い。Mortran によって自動的にコメントが終了する。従って、複数行にわたるコメントは使用できない。

%E は、サブルーチンの区切りを分かりやすくする為によく使われる。ユーザコードの最後に %N か !NOLIST; を書いておくと、EGS4BLOK や EGS4 の Mortran リストは出力されず EGS4MAC とユーザコードのみの Mortran リストが出力される。%In は、Mortran リストにおいてネストのレベルを分かりやすくする為に用いられる。

## 2.12 Mortran 使用時に発生しやすいエラーとその処理

- ; を忘れる事によるエラー

場所にもよるが、通常は 2 つの Statement が 1 つのものと認識されるため、Mortran 段階ではエラーとならないが、FORTRAN のコンパイル時にエラーとなる。FORTRAN のリストを見て、2 つの Statement がつながっている場合には (;) をチェックする。

- ] や ” の忘れまたは過剰

[ と ] の数は必ず同じでなければならない。数が一致しないと Mortran エラーが発生し、例えば足りない場合には Mortran のリストの最後に UNCLOSED BLOCKS というエラーメッセージが打ち出される。問題の箇所を見つけるためには、Mortran のリストの左側に打ち出されるネストレベルを調べる。足りない場合には最後の値が 0 より大きい数になり、] が多すぎる場合には 0 より小さい値になっている。本来ネストのレベルが 0 となるべき場所を順番に調べていけば、どこで忘れたかがわかる。その際、各プログラム単位の最後に %E を書いておくと、どのプログラム単位で間違いが生じたかが判るので便利である。すなわち、画面で %E を検索し、ネストレベルが 0 になっているかどうかを見る。最初に、ネストレベルが 0 以外で終了しているプログラム単位に間違いがある。

” を閉じるのを忘れた場合には、コメント行が継続していると認識され、Mortran のリストは途中で終了し、FATAL STRING OR STATEMENT TOO LONG というエラーメッセージが打ち出される。この場合には、ネストレベルの表示の左側に ” が印字され、コメント行の継続を表示している。

- ブロック記号と ; の関係

; は Statement の最後につけなければならないので、必ず ] の前に書かなければならない。.....]; の様に ] の後に ; が来るとエラーになる。

### 3 応用編

Mortran には、基礎編で紹介した以外にマクロという便利な機能があり、この機能を使用するとプログラミングが容易になる。本章では、マクロの定義の方法および EGS4 のマクロとして組み込まれているものの内主なものについて紹介する。

#### 3.1 マクロ

##### 3.1.1 マクロの定義

Mortran のマクロは次のように表現する。

```
REPLACE pattern WITH replacement
```

Mortran で書かれたプログラム中に pattern と一致する表現があると、replacement 部に書かれたものに置き換えた後、FORTRAN に変換する。Replacement は数値でも、変数でも statement (複数行も可) でもかまわない。例えば、

```
REPLACE {ARRAYSIZE} WITH {50}
```

というマクロがあると、

```
DIMENSION X(ARRAYSIZE);.....DO J=1,ARRAYSIZE [...]
```

は、

```
DIMENSION X(50);.....DO J=1,50 [.....]
```

と同じ FORTRAN に変換される。

プログラム中で pattern と一致する表現を捜す場合にテキスト中のブランクは無視される。従って、

```
REPLACE {SIGMA(1)} WITH {SIGMA1}
```

というマクロに対して

```
SIGMA (1) も SIGMA(1)
```

も有効である。一方、pattern 中に置かれたブランクは意味がある。例えば、

```
REPLACE {DUMP X;} WITH {OUTPUT X; (F10.2);}
```

というマクロは

```
DUMP X;
```

には有効であっても

```
Y = DUMPX;
```

に対しては有効でない。

pattern 部の引き数は、# で表現され、  
replacement 中では {P1}, {P2}.. の様に表現される。

{P1} は 1 番目の引き数、{P2} は 2 番目の引き数である。例えば、

```
REPLACE {PLUS #;} WITH {{P1}={P1}+1;}
```

というマクロがあると、

```
PLUS A(I,J,K);
```

は、

```
A(J,J,K)=A(I,J,K)+1;
```

と同じ FORTRAN statement に変換される。

Mortran3 のマクロでは、「大文字と小文字は区別されている」ので、注意が必要である。EGS4 で使用されているマクロを含めて基本的にマクロは大文字で書かれているので、EGS4 のユーザーコードでマクロを使用する場合には、大文字を使用する方が安全である。Mortran の statement がそのまま残って Fortran 段階でエラーが出る場合には、マクロを「小文字」で使用していないかチェックする必要がある。

マクロは、Mortran のプログラムにおいて、同じ pattern については、後で現れたものの方が適用される。従って、EGS4 において使用されている EGS4MAC 中のマクロと同じ pattern をユーザーコード中で別のマクロに再定義すると、ユーザーコード中のマクロが適用される事になる。(このために、EGS4MAC, ユーザーコード, EGS4BLOK, EGS4 というファイルの接続順番が重要になる。)この機能を使うことにより、EGS4 を様々に変更するでき、しかも変更した内容は、ユーザーコード中のマクロを見れば直ちに判るという事が可能になる。EGS4 では、マクロである事がよく判る様に、ほとんどの場合マクロの先頭に \$ をつけている。

## 3.2 EGS4MAC で使用されているマクロ

### 3.2.1 PARAMETER マクロ

```
REPLACE {PARAMETER #=#;} WITH { REPLACE {{P1}} WITH {{P2}}}
```

により、

```
PARAMETER $MXMED=10;
```

は、テキスト中に \$MXMED が現れると 10 に置き換えて変換される。この例の様に、replacement 中にマクロ表現を使用することも可能である。

配列の引数を PARAMETER マクロにより定義した変数(例えば \$MXMED)で定義しておく、その値を変更する場合に、下記の例の様に PARAMETER マクロでの設定値だけを変更すれば良いので便利である。

```
PARAMETER $NBATCH=5;  
PARAMETER $NDET=11;  
....  
....  
COMMON/TOTALS/PHEI($NDET,150),TRLG($NDET,150),DELE;  
....
```



### 3.2.2 COMIN マクロ

EGS4MAC では、次のマクロが定義されている。:

```
REPLACE {;COMIN/#,#/;} WITH {;COMIN/{P1}/;COMIN/{P2}/;}
```

ユーザーコードや EGS4 システム中で

```
;COMIN/BOUNDS,EPCONT,STACK/;
```

のような文が見つかったらそれは、

```
;COMIN/BOUNDS/; COMIN/EPCONT,STACK/;
```

に変換され、更に

```
;COMIN/BOUNDS/; COMIN/EPCONT/; COMIN/STACK/;
```

に変換される。それぞれの COMIN は、それぞれについて定義された内容に従って、名前付 COMMON に変換される。例えば、COMIN/BOUND/; は、下記のマクロにより COMMON/BOUNDS/ECUT(\$MXREG), PCUT(\$MXREG), VACDST; に変換される。

```
REPLACE {;COMIN/BOUNDS/;} WITH  
  {;COMMON/BOUNDS/ECUT($MXREG),PCUT($MXREG),VACDST;}
```

COMIN マクロは、ユーザーコードと EGS4 が情報をやりとりするのに使用される COMMON をメインプログラムやサブルーチンで定義するために使用される。例えば、

```
COMIN/STACK,BOUNDS/;
```

は、FORTRAN に変換された時点で

```
COMMON/STACK/.....  
COMMON/BOUNDS/.....
```

の様にラベル付き COMMON に置き換えられる。COMIN マクロはプログラム中で COMMON 文が置かれる場所に書けば良い。重要な COMMON BLOCK の名前とそこに含まれている変数及びその意味を第 1 表に示す。

ユーザーコードのメインプログラムとサブルーチン間で用いられる名前付き COMMON を COMIN マクロで定義しておく、EGS4 の COMMON と同様に COMIN で引用する事が出来る。ユーザーコードを作成した後に、COMMON を変更する事はよくあるが、この様にしておけば 1 箇所の変更でその COMMON を使用している全ての COMMON を変更する事ができる。

```
REPLACE {;COMIN/TOTALS/;} WITH  
  {;COMMON/TOTALS/PHEI($NDET,150),TRLG($NDET,150),DELE;}  
  PARAMETER $NDET=10;
```

というマクロを定義しておく、

```
;COMIN/BOUNDS,EPCONT,TOTALS,USEFUL/;
```

が

```
COMMON/BOUNDS/ECUT.....  
COMMON/EPCONT/EDEP.....  
COMMON/TOTALS/PHEI(10,150),TRLG(10,150),DELE  
COMMON/USEFUL/MEDIUM.....
```

の様に変換される。

### 3.2.3 \$LGN (List-Generator) マクロ

いわゆる *list-generator* マクロは、EGS4 システム中で数多く使用されている。

例えば、*list-generator* マクロ

```
$LGN(A,B,C(123));
```

は、

```
A(123),B(123),C(123)
```

という文を作り出す。

\$LGN は BLOCK COMMON でよく使用される。例えば、

```
;COMMON/STACK/$LGN(E,X,Y,Z,U,V,W,DNEAR,WT,IQ,IR($MXSTACK)),NP;
```

は、最終的に以下のような FORTRAN 文となる。:

```
COMMON/STACK/E(40),X(40),Y(40),Z(40),U(40),V(40),W(40),DNEAR(40),W  
*T(40),IQ(40),NP
```

### 3.2.4 \$RANDOMSET マクロ

\$RANDOMSET マクロは、計算スピードのためにインラインで乱数を発生させるマクロである。

\$RANDOMSET マクロは、COMIN/RANDOM/; とペアで使用される。汎用の大型計算機では、

```
REPLACE {;COMIN/RANDOM;} WITH {;COMMON/RANDOM/IXX; INTEGER IX(2);  
REAL*8 DRN; EQUIVALENCE(IX(1),DRN); DATA IX(1)/Z46000000/;}  
REPLACE {$RANDOMSET #;} WITH {IXX=IXX*663608941;IX(2)=IXX;  
{P1}=DRN+0.000;}
```

が使用され、UNIX, VAX あるいは PC では、

```
REPLACE {;COMIN/RANDOM;} WITH {;COMMON/RANDOM/IXX;}  
REPLACE {$RANDOMSET #;} WITH {IXX=IXX*663608941;  
{P1}=0.5+IXX*0.23283064E-09;}
```

が使用される。

\$RANDOMSET は以下のように使用する。

```
$RANDOMSET RN; "Sample RN uniformly on (0,1)"  
PHI=TWOPI*RN; "Obtain azimuthal angle"
```

は、UNIX 等の場合、次のようなインラインの文となる。:

```
IXX=IXX*663608941  
RN=0.5+IXX*0.23283064E-9  
PHI=TWOPI*RN
```

上記の乱数は RAN6 と呼ばれるもので、標準の EGS4 システムに含まれている。

Marsaglia-Zaman random number generator と呼ばれる極めて周期が長く ( $2^{144} \sim 10^{43}$ )、32 ビットのどのような計算機でも使用可能な乱数 [3] も使用可能になっている。

### 3.2.5 デバッグのためのマクロ

- \$TRACE #; \$S1TRACE #; \$S2TRACE #; \$ITRACE #; \$IS1TRACE #; \$IS2TRACE #;  
デバッグの為のマクロで、\$TRACE は、引き数を持たない実数型の変数の場合に、\$S1TRACE は引き数を一つ、\$S2TRACE は引数を 2 つ持つ実数型の変数の場合に使用する。整数型の変数にたいしては、\$ITRACE、\$IS1TRACE と \$IS2TRACE を使用する。

\$S2TRACE #; \$ITRACE #; \$IS1TRACE #; \$IS2TRACE #: は、\$TRACE #; \$S1TRACE #; を基に KEK で作成したマクロで、kek4mac.mor に含まれている。

これらの、マクロを使用する時には、メインプログラムの COMIN マクロ中に DEBUG を含め、Statement 中で QDEBUG=.TRUE.; とする。プログラム実行中に、対象となる変数が計算される毎にどのような式で計算されたかという情報と共に値が出力される。QDEBUG=.FALSE.; の時は、これらのデバッグは出力されないが、FORTRAN のプログラム中には多くの IF 文が残るので、デバッグが不要になった時は、QDEBUG で制御するのではなくこれらのマクロを削除するか、コメントにする方が良い。

- \$CALLTRACE;  
サブルーチンコールがコールされる毎に、サブルーチン名を出力する。デバッグの目的でプログラムの流れを知りたい場合に使用する。但し、ラベルの直後のサブルーチンコールに対しては適用されないので注意する必要がある。

### 3.2.6 The SPECIFY and \$SETINTERVAL#,#;

“specifier”は、認識関数名で 次のような SPECIFY 文によって定義する。

```
SPECIFY specifier AS specification;
```

EGS4MAC.MOR では、以下の SPECIFY マクロが使われている。:

```
SPECIFY SNAME AS ['SINC'|'BLC'|'RTHR'|'RTHRI'];
```

SNAME は、マクロの “pattern” 部で使用するために定義する。\$EVALUATE マクロでの使用例を示す。:

```
REPLACE {$SETINTERVAL#,#;}  
  WITH {[IF] '{P2}'=SNAME [L{P1}={P2}1*{P1}+{P2}0;]  
        [ELSE] [L{P1}={P2}1(MEDIUM)*{P1}+{P2}0(MEDIUM);]}
```

MEDIUM は、計算対象の物質番号である。

### 3.2.7 \$EVALUATE Macro

\$EVALUATE macro の内の一つは:

```
REPLACE {$EVALUATE#USING#(#);} WITH {[IF] '{P2}'='SIN'  
  [{P1}={P2}1(L{P3})*{P3}+{P2}0(L{P3});] [ELSE]  
  [{P1}={P2}1(L{P3},MEDIUM)*{P3}+{P2}0(L{P3},MEDIUM);]}
```

と定義されている。

```
$RANDOMSET RNN038;  
PHI=RNN038*TWOPAI;  
$SET INTERVAL PHI,SINC;  
$EVALUATE SINPHI USING SIN(PHI);
```

というマクロを見ると、2番目と3番目のマクロは、

```
LPHI=SINC1*PHI + SINCO;  
SINPHI=SIN1(LPHAI)*PHI + SIN0(LPHI);
```

に展開される。LPHIは、“sin”関数の表を“piecewise-linear fit”した時の対応する区分を示し、SINPHIは、PHIとLPHIを用いて“sin”の値を求めるものである。

### 3.2.8 \$TRANSFERPROPERTIESTO#FROM#;

粒子が新しく生成した時に、生成前の粒子の位置、ウエイト等のパラメータを新しく生成した粒子に移すマクロであり、以下の様に定義されている。

```
REPLACE {$TRANSFERPROPERTIESTO#FROM#;} WITH {X{P1}=X{P2};Y{P1}=Y{P2};  
Z{P1}=Z{P2};IR{P1}=IR{P2};WT{P1}=WT{P2};DNEAR{P1}=DNEAR{P2};}
```

例えば、SUBROUTINE SHOWERの最初で用いられている

```
$TRANSFERPROPERTIES TO (1) FROM I;
```

によりメインプログラムでXの初期値をXIで与えておくとX(1)に初期値が設定される。

### 3.2.9 ユーザが様々な機能を組み込む為に用意されているマクロ

- \$KERMA-INSERT;: KERMAの計算をする際に使用。DEFAULTはNULL。
- \$DE-FLUCTUATION;: エネルギーロスの計算にLandau Fluctuation等を組み込む為のマクロ。DEFAULTはNULL。
- \$POSITRON-ECUT-DISCARD;: エネルギーEの陽電子が $AE < E < ECUT$ でDISCARDされた時に、消滅ガンマ線をコントロールするマクロ。

```
REPLACE {$POSITRON-ECUT-DISCARD;} WITH {EDEP=PEIE-PRM};
```

が

DEFAULTのマクロであるので消滅ガンマ線が発生するが、EDEP=PEIE+PRMに変更すると消滅ガンマ線が発生しなくなる。

- \$PARTICLE-SELECTION-ELECTRA; \$PARTICLE-SELECTION-PHOTON; 等:  
全ての電子・陽電子と光子の反応にも置かれているマクロで、Leading ParticleやSplitting等機能を組み込む為のマクロ。(例えば\$PARTICLE-SELECTION-COMPT;)DEFAULTはNULL。
- \$SELECT-ELECTRON-MFP;: 電子のmfpを決めるマクロ。

```
REPLACE {$SELECT-ELECTRON-MFP;} WITH {$RANDOMSET RNNE1;  
IF(RNNE1.EQ.0.0) [RNNE1=1.E-30;]  
DEMFP=AMAX1(-ALOG(RNNE1),$EPSEMFP);}
```

がDEFAULTのマクロであり、別のやり方をする際にはこのマクロを変更する。

- \$SELECT-PHOTON-MFP;: 光子のmfpを決めるマクロ。

```
REPLACE {$SELECT-PHOTON-MFP;} WITH {$RANDOMSET RNNE1;
IF(RNNO35.EQ.0.0) [RNN035=1.E-30;] DEMFP=-ALOG(RNNO35);}
```

が DEFAULT のマクロであり、Exponential Transform 等を組み込む際にはこのマクロを変更する。

- \$USER-RANGE-DISCARD;: 電子・陽電子の Range が小さい場合などに、ユーザの判断で DISCARD する機能を組み込むマクロ。DEFAULT は NULL。
- \$SET-TUSTEP-EM-FIELD; \$SET-USTEP-EM-FIELD; \$VACUUM-TRANSPORT-EM-FIELD; \$SET-ANGLES-EM-FIELD; \$SET-TVSTEP-EM-FIELD; \$ADD-WORK-EM-FIELD;: 電磁場中で荷電粒子の輸送を組み込む為のマクロで、DEFAULT はすべて NULL。

## 4 EGS4MAC等のユーザコードでの応用例

### 4.1 EGS4で使用されている mfp 等のパラメータの使用

EGS4 の各サブルーチンでは PEGS4 で計算して得られたデータから粒子の輸送計算に必要なパラメータを求めている。ユーザコードにおいても、これらのパラメータを使用したい場合がある。その場合には、\$SET INTERVAL と \$EVALUATE マクロを次の例のように使用すれば良い。

```
EIG=EI;
$SET INTERVAL GLE,EIG;"SET PWLF INTERVAL"
MEDIUM=MED(2);
GLE=ALOG(EIG);
$EVALUATE GMFPRO USING GNFP(GLE);
```

この例では、リージョン番号 2 の物質の光子の mfp を計算している。EGS4 の中で MEDIUM という変数を使用されているので、EVALUATE マクロを使用する場合には、

```
EIG=E(NP);
MEDOLD=MEDIUM;
MEDIUM=MED(2);
.....
MEDIUM=MEDOLD;
```

のように計算前の状態を保存しておき、終了後に元に戻しておかなければならない。

### 4.2 NEGATIVE-USTEP

NEGATIVE-USTEP が発生した場合に、\$USER-CONTROLS-NEGATIVE-USTEP; を次の例のようにユーザコードでオーバーライドすることにより必要な変数の値を出力し、原因解明をしやすい事ができる。

```
REPLACE{$USER-CONTROLS -NEGATIVE-USTEP;} WITH
{;IF(USTEP.LT.-1.E-4) [OUTPUT USTEP;(' NEGATIVE USTEP=',G20.10);
OUTPUT NP,IR(NP),IQ(NP);(' NP,IR,IQ=',3I5);
OUTPUT X(NP),Y(NP),Z(NP),U(NP),V(NP),W(NP);
(' X,Y,Z,U,V,W=',6G10.4); STOP;]}
```

3-5 行が追加した部分である。

### 4.3 LATCH の使用例

粒子の種類、位置、方向及び重み以外に粒子の属性として、LATCH が COMIN/STACK/; に組み込まれている。LATCH を使用する事により、要因別の計算を行うことができる。

- リージョンに入った時の、粒子の種類を保存して置きたい時 (例えば、周囲に別の物質があり検出器 (リージョン 4) に入った時の粒子の種類毎のエネルギー吸収を知りたい時) SUB-ROUTINE AUSGAB で、

```
IRL=IR(NP);
IF(IRL.NE.IROLD.AND.IRL.EQ.4) [LATCH(NP)=IQ(NP);
                                "Particle enters into detector"]
IF(LATCH(NP).EQ.0) [EPDE(IRL)=EPDE(IRL)+EDEP*DPWT;]
ELSE [EEDE(IRL)=EEDE(IRL)+EDEP*DPWT;]
```

2 番目の Statement で粒子が新しいリージョンに入ったかどうかの判断をし、もしリージョンが変わっていれば、その時の粒子の種類に置き換える。

- 検出器中で光子が多重散乱を経由したエネルギー吸収と直接光電吸収が発生した場合を分けてスコアしたい場合には IAUSFL(18)=1; とすることにより、コンプトン散乱が発生した場合に CALL AUSGAB(IARG); が行われる様に設定した上で、AUSGAB の先頭に、

```
IF(IARG.EQ.17) [LATCH(NP)=LATCH(NP)+1; RETURN;]
```

という statement を置けば良い。これにより、LATCH には、その粒子がこれまでに経験したコンプトン散乱の回数が記録される事になる。実際にエネルギー吸収が生じる可能性がある  $IARG \leq 4$  の時の処理において、

**No Compton** : LATCH(NP)=0;

**Single Compton** : LATCH(NP)=1;

**Multiple Compton** : LATCH(NP)>1;

として、分けてスコアすれば良い。

IAUSFL は、COMIN/EPCONT/; に含まれているので、メインの COMIN に EPCONT を忘れずに加えておかなければならない。

### 4.4 新しい属性の追加

入射後の時間及び飛跡長を知りたい場合の様に 2 つ以上の新たな属性が必要な時は、下記の例のように COMIN/STACK/ と \$TRANSFERPROPERTIES マクロを変更すれば良い。

```
REPLACE {;COMIN/STACK/;} WITH
  {;COMMON/STACK/$LGN(E,X,Y,Z,U,V,W,DNEAR,WT,TTLSS,TIMESS,IQ,
  IR($MXSTACK)),NP;$ENERGY PRECISION E;}
REPLACE {$TRANSFERPROPERTIESTO#FROM#;} WITH
  {X{P1}=X{P2};Y{P1}=Y{P2};Z{P1}=Z{P2};IR{P1}=IR{P2};
  WT{P1}=WT{P2};DNEAR{P1}=DNEAR{P2};
  [IF] '{P2}'='I' [TTLSS(1)=0.0;TIMESS(1)=0.0;]
  [ELSE] [TTLSS{P1}=TTLSS{P2};TIMESS{P1}=TIMESS{P2};]}
```

TTLSE は入射後の飛跡長、TIMES は入射後の時間である。AUSGAB でこれらのデータを使用する際には、AUSGAB の COMIN に STACK を加えておかなければならないことは言うまでもない。また、飛跡長、時間は、粒子が移動する度に CALL される AUSGAB で次のように計算する。

```
TTLSS(NP)=TTLSS(NP)+TVSTEP;
IF(IQ(NP).EQ.0) [ SPEED=2.998E+1;]
ELSE [SPEED=SQRT(BETA2)*2.998E+01;
      "Speed of charged particle in cm/nsec"]
TIMESS(NP)=TIMESS(NP)+TVSTEP/SPEED;
```

TVSTEP は、電子・陽電子の場合、多重散乱の効果を含めた飛跡長である。BETA2 は、 $\beta^2$  で COMMON/EPCONT/ に含まれているので、AUSGAB の COMIN には EPCONT を加えておく。

以上の例のように、EGS4 で計算可能な量であれば標準では得られない情報もマクロの変更によって計算できるようにする事が出来る。

## 4.5 Mortran 及び EGS4 実行時のエラーとその処理

### 4.5.1 Mortran エラーで原因が判らない時

!TRACE 2; を Mortran エラーが発生している Statement の数行前に挿入し、!TRACE 1; を エラーメッセージの数行後ろに挿入する。!TRACE 2 と !TRACE 1 の間に Mortran マクロがあると、Mortran のリストにマクロがどの様に適用されたかが出力される。その結果を見て、原因を調べる事が出来る。例えば

```
!TRACE 2;
$SET INTERVAL GLE,GE;
$EVALUATE GMFPR1 USING GMFP(GLE);
!TRACE 1;
```

となっていると、Mortran リストには、

```
0 !TRACE 2;
REDUCED ;
0 $SET INTERVAL GLE,GE;
MATCHED $SETINTERVALp,p;
ARG 1 GLE
ARG 2 GE
REDUCED LGLE=GE1(MEDIUM)*GLE+GEO(MEDIUM);
0 $EVALUATE GMFPR1 USING GMFP(GLE);
MATCHED $EVALUATEpUSINGp(p);
ARG 1 GMFPR1
ARG 2 GMFP
ARG 3 GLE
REDUCED GMFPR1=GMFP1(LGLE,MEDIUM)*GLE+GMFPO(LGLE,MEDIUM);
0 !TRACE 1;
```

の様に出力される。

### 4.5.2 Large negative “USTEP” が発生した時

USTEP が負の値でその絶対値が  $10^{-4}$  より大きい場合には USTEP が出力される。EGS4 はその後、USTEP を 0 にして計算を続行する。通常の計算では、USTEP の絶対値が  $10^{-3}$  程度であれば特に問題にする必要は無いが、 $10^{-2}$  より大きい場合には Geometry の扱いに間違いがあると考えられるので調べる必要がある。(小さな negative “USTEP” は、計算機の精度から生じる現象である。)

### 4.5.3 Subroutine HATCH に関連したエラー

```
END OF FILE ON UNIT 12
PROGRAM STOPPED IN HATCH BECAUSE THE
FOLLOWING NAME WERE NOT RECOGNIZED:
.....
```

というメッセージを出力してプログラムが停止した場合には、出力された物質名が PEGS4 で作成した物質名と違ってないか調べる。タイプミスをしている事が多い。

```
STOPPED IN HATCH:REQUESTED RAYLEIGH OPTION FOR MEDIUM ##
BUT RAYLEIGH DATA NOT INCLUDED IN DATA CREATED BY PEGS
```

というメッセージを出力してプログラムが停止した場合は、## の番号の物質データに COHERENT データが含まれていないためなので、RAYLEIGH OPTION の使用を止めるか、COHERENT DATA を含む物質データをつくり直す。

#### 4.5.4 配列オーバのエラーが発生した時

プログラムには、間違いがないのに配列オーバのエラーが出る時は、物質データのエネルギー上限値と入射粒子のエネルギーをチェックする。特に、電子・陽電子の場合は、全エネルギーで表されているのにも係わらず、エネルギーの上限値は光子の上限値と同じにすることが多い。このような場合、入射光子のエネルギーが光子の上限値に等しいと、物質データの上限値以上のエネルギーの電子・陽電子が発生することがあり、この時には、配列オーバのエラーが発生する。このような事態をさけるには、ユーザコードで、入射エネルギーを決めた後に、プログラム中で使用されている全ての物質の上限値と入射エネルギーを比較し、入射エネルギーが上限値を超えていたり、上限値を超える粒子が生成する可能性のある場合は、その旨の印字をしプログラムを停止するのが良い。

#### 4.5.5 LOOP が発生してヒストリーが終了しない場合

- (a) CALL SHOWER の直前に、各ヒストリーの最初の乱数シード IXX を出力する OUTPUT IXX; (' IXXST=',I12); を挿入し、プログラムを実行する。
- (b) IXXST を最後の IXX に変更し、ヒストリー数を 1 に変更する。  
\$TRACE と \$S1TRACE ( 及び \$ITRACE, \$IS1TRACE ) を用いて粒子の位置、エネルギー、方向、リージョン等を出力出来るようにする。  
メインプログラムの COMIN に DEBUG を加える。  
実行 Statement 中に QDEBUG=.TRUE.; を挿入する。  
その後、プログラムを実行する。
- (c) 打ち出された結果を調べて、LOOP の原因を見つける。  
\$TRACE 等を使用すると出力が多くなるので、途中で強制的に止める等の処置が必要である。



## 参考文献

- [1] W. R. Nelson, H. Hirayama and D. W. O. Rogers, "THE EGS4 CODE SYSTEM", SLAC-265, December 1985.
- [2] A.J.Cook, "Mortran3 User's Guide", SLAC Computational Research Group Technical Memorandum Number CGTM 209 (1983).
- [3] G.Masaglia and A.Zaman, "A New Class of Random Number Generator", *Annals of Applied Probability* 1, 462-480(1991).

**Table 1. Names of the COMMON blocks important to the user with a brief description of their functions.**

COMMON BLOCK	VARIABLE	FUNCTION
BOUNDS	ECUT	Array of regions' charged particle cutoff energies in MeV.
	PCUT	Array of regions' photon cutoff energies in MeV.
EPCONT	VACDST	Distance to transport in vacuum (default=1.E8).
	EDEP	Energy deposited in MeV (Double Precision).
	TSTEP	Distance to next interaction (cm)
	TUSTEP	Total (curved) step length requested.
	USTEP	User (straight line) step length requested and granted.
	TVSTEP	Actual total (curved) step length to be transported.
	VSTEP	Actual (straight line) step length to be transported.
	IDISC	User discard request flag (to be set in HOWFAR. IDISC > 0 means user requests immediate discard, IDISC < 0 means user requests discard after completion of transport, and IDISC=0 (default) means no user discard requested.
	IROL	Index of previous region.
	IRNEW	Index of new region.
	RHOF	Value of density correction (default=1)
	EOLD	Charged particle (total) energy at beginning of step in MeV.
	ENEW	Charged particle (total) energy at end of step in MeV.
	BETA2	Beta squared for present particle.
	IAUSFL	Array of flags for turning on various calls to AUSGAB.
MEDIA	EKE	Kinetic energy of charged particle in MeV.
	ELKE	Natural logarithm of EKE.
	GLE	Natural logarithm of photon energy.
	TSCAT	See Eq.2.14.82 in SLAC-265.
	NMED	Number of media being used (default=1).
	MEDIA	Array containing names of media (default is NaI).
	IRAYLM	Array of flags for turning on (=1) coherent (Rayleigh) scattering in various media. Set in HATCH based on values of IRAYLR.
	RLC	Array containing radiation length of the media in cm.
	RLDU	Array containing radiation length of the media in distance units established by DUNIT.
	RHO	Array containing density of the media in g/cm <sup>3</sup> .
	MED	Array containing medium index for each region.
	DUNIT	The distance unit to be used: DUNIT=1 (default) establishes all distances in cm; whereas, DUNIT=2.54 establishes all distances in inches.
	KMPI	FORTTRAN unit number (default=12) from which to read material data.
	KMPOI	FORTTRAN unit number (default=8) on which to "echo" material data (e.g., printed output, "dummy" output, etc.).
	MISC	

**Table 1. Names of the COMMON blocks important to the user with a brief description of their functions (continued).**

COMMON BLOCK	VARIABLE	FUNCTION
MISC	RHOR	Array containing the density for each region ( $\text{g}/\text{cm}^3$ ). If this is different than the default density for the medium for that region, the cross section and stopping power (with the exception of the density effect) are scaled appropriately.
	NOSCAT	Number of times multiple scattering has been bypassed in subroutine MSCAT (initialized to 0 in BLOCK DATA).
	IRAYLR	Array of flags for turning on (=1) coherent (Rayleigh) scattering in various regions (default=0).
RANDOM STACK	IXX	Random number generator seed (default=123456789).
	note:	This COMMON block contains the information about the particles currently in the shower. All of the following variables are arrays except NP.
	E	Total energy in MEV (Double precision).
	X, Y, Z	Position of particle in units established by DUNIT.
	U, V, W	Direction cosines of particle (not necessarily normalized — see Section A2.4.1c of SLAC-265).
	DNEAR	A lower bound of distance from (X,Y,Z) to nearest surface of current region.
	WT	Statistical weight of current particles (default=1.0), To be used in conjunction with variance reduction techniques as determined by user.
	IQ	Integer charge of particle (+1, 0, -1).
	IR	Index of particle's current region.
	NP	The stack pointer (i.e., the particle currently pointed to). Also, the number of particles on the stack.
THRESH	RMt2	Twice the electron rest mass energy in MeV.
	RMSQ	Electron rest mass energy squared in MeV-squared.
	AP	Array containing PEGS lower photon cutoff energy for each medium in MeV.
	UP	Array containing PEGS upper photon cutoff energy for each medium in MeV.
	AE	Array containing PEGS lower charged particle cutoff energy for each medium in MeV.
	UE	Array containing PEGS upper charged particle cutoff energy for each medium in MeV.
	TE	Same as AE except kinetic energy rather than total energy.
	THMOLL	Array containing the Moller threshold energy (THMOLL=AE+TE) for each medium in MeV.

**Table 1. Names of the COMMON blocks important to the user with a brief description of their functions (continued).**

COMMON BLOCK	VARIABLE	FUNCTION
UPHIOT	THETA	Collision scattering angle (polar).
	SIN THE	Sine of THETA.
	COS THE	Cosine of THETA.
	SIN PHI	Sine of PHI (the azimuthal scattering angle of the collision).
	PI	Pi.
USEFUL	TWOPI	Twice pi.
	MEDIUM	Index of current medium. If vacuum, then MEDIUM=0.
	MEDOLD	Index of previous medium.
	RM	Electron rest mass energy in MeV.
	PRM	“Precision” electron rest mass energy in MeV (Double precision).
	PRMT2	Twice PRM (Double precision).
	IBLOBE	Flag indicating if photon is below binding energy (EBINDA) after a photoelectric interaction 8yes=1).

# PEGS4の使い方

PEGS4はEGS4用物質データを作成するためのプログラムである。PEGS4を使用するためには入力データファイルを作成する。対象物質が「元素」、「化合物」または「混合物」のどれであるかによって、入力データファイルは3種類に分類できるので、この3種類に分けて入力データファイル作成法を述べる。本文では、具体例と誤りやすい点を中心に述べる。一方、PEGS4の使用法については、SLAC-265のA3.3“PEGS4 Options and Input Specifications”に網羅的な記述がある。PEGS4使用時には、本文と上記マニュアルの両方を参照されたい。

## 1 元素用入力ファイル

例えば、鉄の物質データを10 keVから50 MeVの範囲で作成するための入力ファイルは、

```
ELEM
  &INP IAPRIM=1,IRAYL=1 &END
FE-RAYLEIGH                FE
FE
ENER
  &INP AE=0.521,UE=50.511,AP=0.01,UP=50.0 &END
TEST
  &INP &END
PWLF
  &INP &END
DECK
  &INP &END
```

とする。

- 1行目の「ELEM」は物質が単一の元素からなる事を示している。(第1カラムから入力する。)
- 2、6、8、10そして12行目は「namelist入力」というIBM拡張FORTRANの入力である<sup>1</sup>。
- 「IAPRIM=1」は電子の放射阻止能がICRU Report 37に一致するように制動輻射断面積を規格化するオプションである<sup>2</sup>。このオプションを使用の方が正確なデータが得られるので、このオプションは常時使用するべきである。
- 「IRAYL=1」は、光子の弾性散乱であるレイリー散乱を含めるオプションである。
- 「FE-RAYLEIGH」は作成中の物質データの名称である。EGS4のユーザーコードで各領域に物質データを割り当てる際、この名前を指定する。この名称は、24文字以内でユーザーが自由につける事ができる。物質名が24文字未満の場合には、物質名の後に空白を指定する。空白の長さは、「24-物質名の文字数」である。
- 3行目 第31カラムからの「FE」は密度効果に関するSternheimer-Seltzer-Berger係数(以下、「SSB係数」)を指定する変数である。
  - 対象とする物質がSLAC-265の表2.13.2に含まれる場合には、その表中の名称を入力する。(必ず第31カラム目から入力する事。)

<sup>1</sup>「namelist入力」は、プログラム中に変数の規定値を与えておき、必要に応じて値を上書き入力するための入力形式である。「namelist入力」では、&INPと&ENDの間に「変数名=値」という組みを「,」で区切りながら入力する。複数行に渡ってもよい。MS-FORTRANの場合には、「&END」の代わりに「/」を用いる。

<sup>2</sup>NRCCにおける拡張機能の一つ。したがって、SLAC-265には関連する記述がない。

- 対象とする物質が同表にない場合には、この部分は無視され、Sternheimer-Peierls の一般式により密度効果の計算を行う。
- 4行目の「FE」は、元素名を指定するための原子記号である。PEGS4はこの記号によって元素を決め、密度などを自動的に設定する。
- 5行目の「ENER」は物質データを作成するエネルギー範囲の設定開始を指示し、6行目の namelist 入力で具体的にエネルギー範囲を入力する。
  - AE、UE は電子に対する物質データの下限および上限エネルギーである ( MeV 単位 )、AE、UE は静止エネルギーを含む全エネルギーで指定する
    - \* AE<0.511 を入力すると負の運動エネルギーを指定したことになり、PEGS4は誤動作し混乱した出力を行う。この場合、原因特定に時間がかかることが多いので、このような入力をしないよう注意する必要がある。
    - \* UE=50.0 と入力すると運動エネルギーでは 49.489 MeV までのデータを作成した事になる。その時に EGS4 で運動エネルギー 50 MeV の電子を入射粒子とすると物質データのエネルギーの範囲外となり、異常動作が起こる。これを防ぐため、電子のエネルギー範囲の指定では静止質量の加算に注意する事。
  - AP、UP は光子に対する物質データの下限および上限エネルギーである ( MeV 単位 )、

エネルギー範囲を過大に広く設定すると、内そうの精度が少し悪くなる。一方、エネルギー範囲をシミュレーションで使用するエネルギー範囲と厳密に同じにしておくと、シミュレーションの条件を変える毎に PEGS4 で物質データを作成し直す必要があり不便である。したがって、通常使用するエネルギー範囲より多少広いエネルギー範囲で物質データを作成すると便利である。

- 「TEST」はラインプリンターイメージによるデータのプロットを指示する。(通常変更しない)
- 「PWLF」は区間毎に直線内そうする事を指示する。(通常変更しない)
- 「DECK」は、直線内そう結果を出力する事を指示する。(通常変更しない)

次に、単原子(非分子)気体用の入力データについて述べる。この例として、Xeガス(1気圧、0°C)用入力の最初の4行を示す<sup>3</sup>。(ENER以降は、前の例と同じ書式であるので省略する。)

```
ELEM
&INP RHO=5.89E-3,GASP=1.,IAPRIM=1 &END
XENON-GAS                XE-GAS
XE
```

- 「RHO」は、密度 (g/cm<sup>3</sup>) である。但し、この例のように対象物質が気体の場合には、標準状態 (Standard Temperature and Pressure= 0 °C、1 気圧) での密度を入力する。
- 2行目の namelist 入力中で「GASP=ガス圧力」という入力を行うと PEGS4 は、対象物質が気体であると認識する。ガス圧力は、標準温度 (0 °C) での値を「気圧」単位で入力する。気体の温度が標準温度と異なる場合には、その温度の気体を体積を一定に保ったまま、標準温度に変更した場合の圧力を計算し、これを入力する。

<sup>3</sup>SLAC265 の表 A3.3.1 には、ELEM2 の入力として GASP を示していないが、ELEM に対しても GASP を使用できる。

- 「XENON-GAS」は作成中の物質データの名称である。EGS4のユーザーコードで各領域に物質データを割り当てる際、この名前を指定する。この名称は、24文字以内でユーザーが自由につける事ができる。
- 「XE-GAS」は、SLAC-265の表 2.13.2 中の Xeガスの名称である。
- 4行目の「XE」は、元素名を指定するための原子記号である。

## 2 化合物用入力ファイル

化合物物質データ作成用入力の例として、アクリル (PMMA : ポリメタクリル酸メチル) 用入力ファイルを示す。

```
COMP
  &INP NE=3,RHO=1.19,PZ=5.,8.,2.,IAPRIM=1 &END
PMMA-IAPRIM          PMMA
C H O
ENER
  &INP AE=0.521,UE=50.511,AP=0.01,UP=50.0 &END
TEST
  &INP &END
PWLF
  &INP &END
DECK
  &INP &END
```

- 1行目で「COMP」と入力すると、化合物用入力となる。
- 「NE=3」は化合物中の元素が3種類である事を指定している。
- 「RHO」は、密度 (g/cm<sup>3</sup>) である。
- PZは、化合物中の原子数の比である。
- 3行目は、ELEMの場合と同じである。SSB係数指定のための第31カラムからの入力は、対象とする化合物がSLAC-265の表 2.13.2 に含まれている場合には、その中から選んで入力する。この場合のようにこの表にない場合には、この入力欄は無視されるので、空白でもよい。
- 4行目では、元素記号をPZに対応する順に、(A2,1X)書式で入力する。
- 5行目以降は、ELEMの場合と同じである。

物質が気体で化合物である場合の例として、CO<sub>2</sub>ガスに対する入力の最初の4行を示す。(5行目以降は、ここまでの例と同じであるので省略する。)0°C、1気圧の場合には、

```
COMP
  &INP NE=2,RHO=1.977E-3,GASP=1.,PZ=1.,2.,IAPRIM=1 &END
CO2          CO2-GAS
C O
```

0°C、10気圧の場合には、

```

COMP
&INP NE=2,RHO=1.977E-3,GASP=10.,PZ=1.,2.,IAPRIM=1 &END
CO2-10atm          CO2-GAS
C 0

```

20 °C、1 気圧の場合には、

```

COMP
&INP NE=2,RHO=1.977E-3,GASP=0.93174,PZ=1.,2.,IAPRIM=1 &END
CO2-20degree      CO2-GAS
C 0

```

となる。

- RHO は常に STP 状態での密度を使用する。
- 最後の例では、20 °C、1 気圧の気体を体積を一定に保って 0 °C に冷却した場合の圧力、0.93174 気圧 (=273 °C/293 °C) を GASP の値として入力する。
- 「CO2-10atm」、「CO2-20degree」は、ユーザーが自由に付けるデータ名である。
- 「CO2-GAS」は、SLAC-265 の表 2.13.2 中の CO<sub>2</sub> ガスの名称である。

次に、水素ガス (0 °C、1 気圧) の場合には、

```

COMP
&COMP NE=2,RHO=8.99E-5,GASP=1.,IAPRIM=1,PZ=1.,1. &END
H2-GAS            H2-GAS
H H

```

とする。

- PEGS4 では水素ガスなどの単一元素分子気体を化合物として扱う<sup>4</sup>。
- NE=1 はエラーになるので、NE=2 として入力する。
- 「H2-GAS」は密度効果表 (SLAC265 表 2.13.2) 中の水素ガスの名称。

### 3 混合物用入力ファイル

混合物物質データ作成用入力の例として、鉛ガラス用入力ファイルの最初の 4 行を示す (5 行目以降は元素用入力を参照。)

```

MIXT
&INP NE=5,RHO=3.61,RHOZ=41.8,21.0,29.0,5.0,2.2,IAPRIM=1 &END
LEAD GLASS
PB SI O K NA

```

- 1 行目で「MIXT」と入力すると、混合物用入力となる。
- 「NE=5」は混合物に 5 種類の元素が含まれる事を示す。

<sup>4</sup>Sternheimer-Peierls の一般式では、分子気体を化合物として扱っている。このため、SSB 係数が与えられてない単元素分子気体を扱う場合、SP の一般式を用いるので、その物質を化合物とする必要がある。一方、この例のように SSB 係数を用いる場合には、単元素分子気体を単体としても化合物としてもどちらでもよい。



- RHO は、混合物の密度 ( $\text{g}/\text{cm}^3$ )。
- RHOZ は、混合物中の元素の質量比。
- 4 行目で RHOZ と同順で元素名を入力する。

物質が気体で混合物の場合の例として、空気 (20 °C、1 気圧) 用の入力の最初の 5 行を示す。

```
MIXT
&INP NE=3,RHO=1.2929E-3,GASP=0.93174,RHOZ=0.75575,0.23143,0.01282,
IAPRIM=1 &END
AIR                AIR-GAS
N 0 AR
```

- RHO の値は、STP 状態での密度 ( $\text{g}/\text{cm}^3$ ) である。
- 「AIR-GAS」は、SLAC265 の表 2.13.2 中の空気の名称である。
- Ar は 100 keV 以下の低エネルギーの計算において重要である。

## 4 CALL オプション

CALL オプションは、物質データ中の特定の評価値を出力させるための入力オプションである。例えば、次の入力では、49.99 MeV 光子の平均自由行程を出力する。

```
ELEM
&INP IAPRIM=1 &END
PB
PB
CALL
&INP XP(1)=49.99 &END
GMFP
```

この結果の出力は、

```
OPT=CALL
FUNCTIONCALL:    1.95522 = GMFP  OF 49.9900
```

となる。ここで、GMFP は cm 単位ではなく、放射長 (r.l.) 単位で出力される事に注意する必要がある。

## 5 国内で改良した PEGS4 の使用方法

- PHOTX データに基づく光子断面積<sup>5</sup>。
  - KEK 版 EGS4 システムに含まれている pgs4phtx.dat を使用する。 pgs4.mortran 内の
 

```
OPEN(8,file='pegs4pepr.dat');
```

 をコメントアウトし、
 

```
OPEN(8,file='pegs4phtx.dat');
```

<sup>5</sup>坂本 幸夫 “PEGS4 用光子断面積 PHOTX データ”、第 3 回 EGS4 研究会、KEK Proceedings, 93-15 (Dec 1993) 参照。

をコメントアウトしないように変更する。

- 通常の計算では、pegs4pepr.dat よりも pegas4phtx.dat を使用する方がよい。
- Sternheimer、Berger と Seltzer による 1984 年版 Sternheimer 密度効果係数<sup>6</sup>。
  - 下記のファイル内の pegas4.mortran を使用する。  
`ftp.kek.jp:/kek/kek_egs4/pegs4/pegs4.tar.Z`
  - 表 1 に示す物質名を PEGS4 入力ファイルの 3 行目の 31 カラムから指定する。(表 1 の物質名はこの拡張版の PEGS4 のみに有効であり、標準の PEGS4 では SLAC-265 の表 2.13.2 に含まれる物質名のみが使用できる事に注意。)
- 束縛電子コンプトン散乱断面積、インコヒーレント散乱関数とコンプトンプロファイル<sup>7</sup>。
  - 下記のファイル内の pegas4nb.mortran を使用する。  
`ftp.kek.jp:/kek/kek_egs4/lscat/lscat_1.1.tar.Z`
  - 100 keV 以下の低エネルギー光子輸送の場合に重要。

---

<sup>6</sup>H. Hiramaya, “Revision of the Sternheimer Density Effect Coefficients in PEGS4”, KEK Internal 95-17 (1995). 参照

<sup>7</sup>Y. Namito et al “LSCAT: Low-Energy Photon-Scattering Expansion for the EGS4 Code”, KEK Internal 95-10 (1995). 参照

**Table 1 Identifier of medium name for Sternheimer-Seltzer-Berger Coefficients in PEGS4**

LABEL	LABEL	LABEL
H2-GAS	RH	PU
H2-LIQUID	PD	AM
HE-GAS	AG	CM
LI	CD	BK
BE	IN	A-150 TE PLASTIC
B	SN	ACETONE
C-2.265 G/CM**3	SB	ACETYLENE
C-2.00 G/CM**3	TE	ADENINE
C-1.70 G/CM**3	I	ADIPOSE TISSUE
N2-GAS	XE-GAS	AIR-GAS
O2-GAS	CS	ALANINE
F	BA	ALUMINIUM OXIDE
NE-GAS	LA	AMBER
NA	CE	AMMONIA
MG	PR	ANILINE
AL	ND	ANTHRACENE
SI	PM	B-100 BONE-EQ. PLASTIC
P	SM	BAKELITE
S	EU	BARIUM FLUORIDE
CL	GD	BARIUM SULFATE
AR-GAS	TB	BENZENE
K	DY	BERYLLIUM OXIDE
CA	HO	BISMUTH GERMANIUM OXIDE
SC	ER	BLOOD (ICRP)
TI	TM	BONE, COMPACT (ICRU)
V	YB	BONE CORTICAL (ICRP)
CR	LU	BORON CARBIDE
MN	HF	BORN OXIDE
FE	TA	BRAIN (ICRP)
CO	W	BUTANE
NI	RE	N-BUTYL ALCHOL
CU	OS	C-552 AIR-EQ. PLASTIC
ZN	IR	CADMIUM TELLURIDE
GA	PT	CADMIUM TUNGSTATE
GE	AU	CALCIUM CARBONITE
AS	HG	CALCIUM FLUORIDE
SE	TL	CALCIUM OXIDE
BR	PB	CALCIUM SULFATE
KR-GAS	BI	CALCIUM TUNGSTATE
RB	PO	CARBON DIOXIDE
SR	RN-GAS	CARBON TETRACHLORIDE
Y	RA	CELLOPHANE
ZR	AC	CELLULOSE ACETATE BUTYRA
NI	TH	CELLULOSE NITRATE
MO	PA	CERIC SURFARE DOSIMETER
TC	U	CESIUM FLUORIDE
RU	NP	CESIUM IODIME

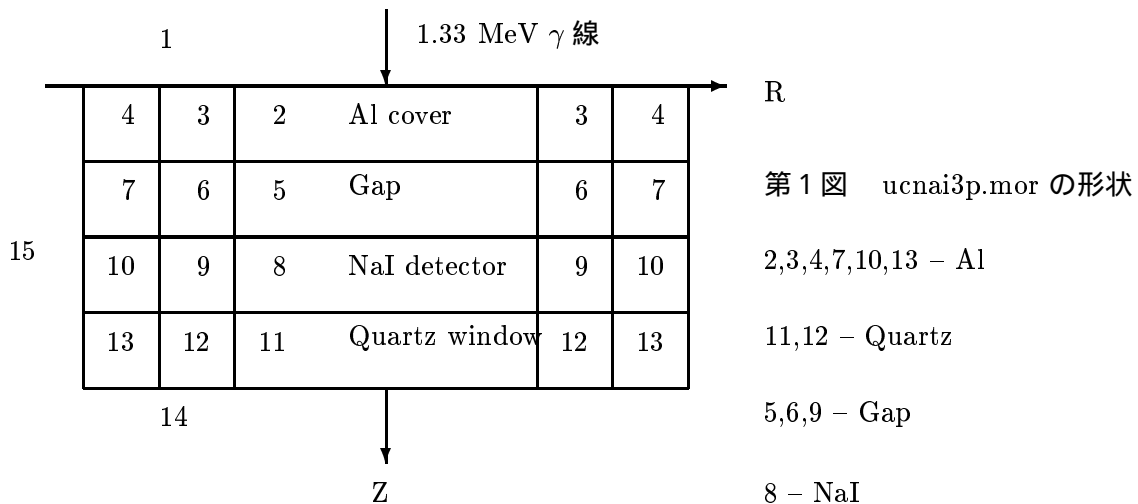
Table 1 Identifier of medium name for Sternheimer-Seltzer-Berger Coefficients in PEGS4

LABEL	LABEL	LABEL
CHLOROBENZENE	LITHIUM TETRABORATE	POTASSIUM IODINE
CHLOROFORM	LUNG (ICRP)	POTASSIUM OXIDE
CONCRETE, PORTLAND	M3 WAX	PROPANE
CYCLOHEXANE	MAGNESIUM CARBONATE	PROPANE, LIQUID
1,2-DICHLOROBENZENE	MANESIUM FLUORIDE	N-PROPYL ALCOHOL
DICHLORODIETHYL ETHER	MAGNESIUM OXIDE	PYRIDINE
1,2-DICHLOROETHANE	MAGNESIUM TETRABORATE	RUBBER, BUTYL
DIETHYL ETHER	MERCURIC IODIDE	RUBBER, NATURAL
N,N-DIMETHYL FORMAMIDE	METHANE	RUBBER, NEOPRENE
DIMETHYL SULFOXIDE	METHANOL	SILICON DIOXIDE
ETHANE	MIX D WAX	SILVER BROMIDE
ETHYL ALCOHOL	MS20 TISSUE SUBSTITUTE	SILVER CHLORIDE
ETHYL CELLULOSE	MUSCLE, SKELETAL (ICRP)	SILVER HALIDES IN EMUL.
ETHYLENE	MUSCLE, STRIATED (ICRU)	SILVER IODIDE
EYE LENS (ICRP)	MUSCLE-EQ. LIQ. W SUCROS	SKIN (ICRP)
FERRIC OXIDE	MUSCLE-EQ. LIQ. W/O SUCR	SODIUM CARBONATE
FERROBORIDE	NAPHTHALENE	SODIUM IODIDE
FERROUS OXIDE	NITROBENZENE	SODIUM MONOXIDE
FERROUS SULFATE DOSIMETE	NITROUS OXIDE	SODIUM NITRATE
FREON-12	NYLON, DU PONT	STILBENE
FREON-12B2	NYLON, TYPE 6 AND 6/6	SUCROSE
FREON-13	NYLON, TYPE 6/10	TRRPHENYL
FREON-13B1	NYLON, TYPE 11	TESTES (ICRP)
FREON-13I1	OCTANE, LIQUID	TETRACHLOROETHTLENE
GADOLINIUM OXYSULFIDE	PARAFFIN WAX	THALLIUM CHLORIDE
GALLIUM ARSENIDE	N-PENTANE	TISSUE, SOFT (ICRP)
GEL IN PHOTOGRAPHIC EMUL	PHOTOGRAPHIC EMULSION	ICRU FOUR-COMP. TISSUE
GLASS (PYREX)	PLASTIC SCINT.	TISSUE-EQ. GAS (METHANE)
GLASS, LEAD	PLUTONIUM DIOXIDE	TISSUE-EQ. GAS (PROPANE)
GLASS, PLATE	POLYCRYLONITRILE	TITANIUM DIOXIDE
GLUCOSE	POLYCARBONATE	TOLUEN
GLUTAMINE	POLYCHLOROSTYRWNE	TRICHLOROETHYLENE
GLYCEROL	POLYETHYLENE	TRIETHYL PHOSPHATE
GUANINE	MYLAR	TUNGSTEN HEXAFLUORIDE
GYPNUM, PLASTER OF PARIS	LUCITE	URANIUM DICARBIDE
N-HEPTANE	POLYOXYMETHYLENE	URANIUM MONOCARBIDE
N-HEXANE	POLYPROPYLENE	URANIUM OXIDE
KAPTON, POLYIMIDE FILM	POLYSTYRENE	UREA
LANTHANUM OXYBROMIDE	TEFLON	VALINE
LANTHANUM OXYSULFIDE	POLYTRIFLUOROCHLOROETHY.	VITON
LEAD OXIDE	POLYVINYL ACETATE	WATER, LIQUID
LITHIUM AMIDE	POLYVINYL ALCOHOL	WATER VAPOR
LITHIUM CARBONATE	POLYVINYL BUTYRAL	XYLENE
LITHIUM FLUORIDE	POLYVINYL CHLORIDE	
LITHIUM HYDRIDE	SARAN	
LITHIUM IODIDE	PLOYVINYLIDENE FLUORIDE	
LITHIUM OXIDE	POLYVINYL PYRROLIDONE	

# ユーザコードの書き方

## 1 はじめに

ユーザコードは、SLAC-265 の説明に従って自分で組み立ててもよいが、間違いを起こす可能性があるため、自分の計算したい内容に近いユーザコードを基にして作り替えるのが最も現実的である。ここでは、ユーザコードの基本的な構成を、最も良く使用される形状(円筒平板形状: 第1図)で NaI 検出器のレスポンスを計算するユーザコード ucnai3p.mor を用いて紹介する。



## 2 STEP 1

EGS4で使用しているマクロをユーザが変更したい場合には、ここに over-ride するマクロを挿入する。Mortran は、同じパターンのマクロがある場合、最も新しいものを採用する。EGS4で使用するマクロのファイル egs4mac.mortran は、ユーザコードの前に置かれるので、ユーザコードに書かれたマクロはパターンが同じ場合には、EGS4のマクロを変更し、従ってEGS4そのものを変更する事ができる。Over-ride マクロは、他の場所に置くことも出来るが、まとめておいた方がわかりやすい。

最初に置かれているマクロは、PRESTAで必要なリージョン境界までの最短距離を計算するものである。使用する形状により計算方法が異なるので、PRESTAを使用する場合には、その形状に合うように変更する必要がある。

```

REPLACE{ $CALL-HOWNEAR(#); } WITH
  { $CALL-HOWNEAR-FOR-SLAC-CYLINDRICAL-GEOMETRY({P1}); }
"          ****"
"THIS IS THE MACRO THAT SHOULD RETURN THE CLOSEST PERPENDICULAR"
"DISTANCE TO ANY SURFACE WHICH FORMS A BOUNDARY FOR THE CURRENT"
"REGION. IN THIS APPLICATION IT IS REPLACED BY THE MACRO FOLLOWING"
"WHICH IS SPECIALIZED FOR THE SLAC"
"          ****"
"CYLINDRICAL-PLANE GEOMETRY."
"IT IS THE USER'S RESPONSIBILITY TO PROVIDE THIS MACRO FOR HIS OWN"
"GEOMETRY."
"++++"
; "BUFFER FLUSH"
REPLACE{ $CALL-HOWNEAR-FOR-SLAC-CYLINDRICAL-GEOMETRY(#); } WITH

```

```

"      ====="
  {;ZL=Z(NP);RL=SQRT(X(NP)**2+Y(NP)**2);
  $GET-JR-JZ(IRL);
  ZLEFT=ZL-PCOORD(3,JZ);ZRIGHT=PCOORD(3,JZ+1)-ZL;ROUT=CYRAD(JR)-RL;
  {P1}=MIN(ZLEFT,ZRIGHT,ROUT);
  IF(JR.NE.1)[RIN=RL-CYRAD(JR-1);{P1}=MIN({P1},RIN);]}
"THIS ROUTINE IS INTENDED TO BE USED TO CALCULATE THE MINIMUM      "
"PERPENDICLAR TO THE NEAREST BOUNDING SURFACE. THIS VERSION IS    "
"SPECIALY DESIGNED FOR THE SLAC CYLINDRICAL GEOMETRY PACKAGE.    "
"A DIFFERENT VERSION IS NEEDED FOR OTHER                          "
"****                                                                "
"GEOMETRY PACKAGES.                                             "

; "BUFFER FLUSH"
"SLAC MACRO TO GET CYLINDER (JR) AND SLAB (JZ) ZONES FROM REGION  "
"NUMBER"
"*****
REPLACE {$GET-JR-JZ(#);} WITH
"      ====="
  {;JZ=({P1}-2)/NCYLP+1;JR={P1}-1-NCYLP*(JZ-1);}
; "BUFFER FLUSH"
"GEOMETRICAL INFORMATION"
"SLAC DEFINITION OF /GEOM/ AND RE-DEFINITIONS FOR /CYLDTA/ AND  "
"/PLADTA/                "

```

上記のマクロ中で使用されている変数を SUBROUTINE ELECTR で使用するために関連する COMIN を変更する。

```

"****                                                                "
REPLACE {;COMIN/GEOM/;} WITH {;COMIN/CYLDTA,PLADTA/;}
"      ====="

REPLACE {;COMIN/CYLDTA/;} WITH
"      ====="
  {;COMMON/CYLDTA/CYRAD($MXCYLS),CYRAD2($MXCYLS),NCYLP;}
"NOTE: CYRAD MUST BE PROVIDED IN ADDITION TO CYRAD2 (USUALLY IN  "
"MAIN)"

REPLACE {;COMIN/PLADTA/;} WITH
"      ====="
  {;COMMON/PLADTA/PCOORD(3,$MXPLNS),PNORM(3,$MXPLNS);}

```

EGS4 では使用していないが、ユーザコードを簡単にするために定義したマクロがあれば同じくこの Step 1 に書いて置く方がよい。ucnai3p.mor では、AUSGAB で score する変数関係の COMMON/TOTAL、ジオメトリに関する COMMON/PASSIT、途中結果の打ち出しに関する COMMON/LINES を COMIN マクロを用いて定義している。

```

"COMMON to define variables to score at AUSGAB"
"DEPE:deposited energy inside the detector"
"DELTAE:energy bin width in MeV"
"SPG:Gamma spectrum, SPE:Electron spectrum, SPP:Positron spectrum"

REPLACE {;COMIN/TOTALS/;} WITH
{;COMMON/TOTALS/DEPE,DELTAE,SPG($NDET,$NEBIN),SPE($NDET,$NEBIN),
SPP($NDET,$NEBIN);}

"COMMON of print-out parameter"

REPLACE {;COMIN/LINES/;} WITH
{;COMMON/LINES/NLINES,NWRITE,NCOUNT,ILINES;}

"COMMON of geometry related parameter"

```

```
REPLACE {;COMIN/PASSIT/;} WITH
      {;COMMON/PASSIT/NREG,NPLAN;}
```

\$PARAMETER 文は、プログラム中で使用される変数の値を定義している。\$PARAMETER 文を用いて配列の大きさを定義しておく、変更したい場合に、\$PARAMETER 文で指定した値の変更のみで済むので便利である。ucnai3p.mor で使用している\$PARAMETER の内、

```
PARAMETER $MXPLNS=5; "NUMBER OF PLANE"
PARAMETER $MXCYLS=3; "NUMBER OF CYLINDER"
PARAMETER $NBATCH=50; "Number of batch"
PARAMETER $NEBIN=50; "Number of energy bin"
PARAMETER $NDET=1; "Number of detector"
```

が、配列の大きさに関連したものである。

次に、メインプログラムで使用する COMMON と DIMENSION 関係の宣言文を置く。EGS4 の COMMON は、マクロの形で定義されているので、必要な COMMON の名前を

```
;COMIN/DEBUG,BOUNDS,BREMPR,EDGE,ELECIN,ETALY1,GEOM,LINES,MEDIA,
MISC,NTALY1,PASSIT,RANDOM,STACK,THRESH,TOTALS,UPHIOT,USEFUL,USER/;
```

の様に記載する。LINES, PASSIT, TOTALS は、このユーザコードで定義したものである。

次に、PEGS4 で計算した物質データを読み込むのに必要な定義を行う。物質名は、PEGS4 で物質データを作る時に指定した物質名で 24 文字で指定するので、MADARR の最初の配列は、必ず 24 でなければならない。2 番目の配列はユーザコード中で使用する物質の数に対応している。この例では 3 種類であるので 3 となっている。

UNIT 6, 8 の OPEN 文については、この例に限る事なく各自の好きな名前として良いが、UNIT 12 については、egs4run を用いて実行する場合にはこの例の様に mortjob.xsec (mortjob.xse:PC の場合) でなければならない。

### 3 STEP 2

以下のように、平板、円筒の数、リージョン数と共に SUBROUTINE HATCH を CALL する前に定義しておかなければならない変数を定義する。NCYLP は、PRESTA で使用するために定義する変数である。NMED は、ユーザコードで使用する物質の数で、MEDARR の 2 番目の配列に対応する。次に、リージョン数、各リージョンへの物質 (物質番号) ECUT, PCUT, K X-ray の対象となる原子番号の割当や設定を行う。

```
NMED=3; "NUMBER OF MEDIA"

DO J=1,NMED [
DO I=1,24 [MEDIA(I,J)=MEDARR(I,J);]]

NPLAN=$MXPLNS; "NUMBER OF PLANES"
NCYL=$MXCYLS; "NUMBER OF CYLINDER"
NCYLP=NCYL; "*PRESTA*"

NREG=(NPLAN-1)*NCYL+3; "NUMBER OF REGIONS (INCLUDING OUTSIDE VACUUM"
" REGION) "
IRZ=NREG-3;
"SET MEDIUM INDEX FOR EACH REGION"

/MED(1),MED(NREG-1),MED(NREG)/=0; "VACUUM REGIONS"

/MED(2),MED(3),MED(4),MED(7),MED(10),MED(13)/=2;
```

```

"                                     Al region"
MED(8)=1;  "NaI(Tl) detector region"
/MED(11),MED(12)/=3;  "Quartz region"
/ECUT(2),ECUT(3),ECUT(4),ECUT(7),ECUT(8),ECUT(10)/=0.561;
/ECUT(11),ECUT(12)/=0.561;

/MED(5),MED(6),MED(9)/=0;  "Vacuum region inside case"

IEDGFL(8)=53;  "53:Atomic number of I"
"                                     0:K-X ray of I is not produced"

```

PEGS4で物質データを作成する場合に、AEおよびAPでカットオフエネルギーを決めている。粒子のエネルギーがAEやAP以下になると追跡を止め、粒子の持つエネルギーがその場所で吸収された扱いになる。またAE以下の二次電子を生じる散乱とAP以下の光子を生じる制動輻射は、個別の粒子として扱わず電子/陽電子の移動に伴う連続エネルギー減衰に含めている。ユーザコードにおいてECUT、PCUTを定義しない場合には、自動的にAE、APが割り当てられる。AE、APよりも高い値にECUT、PCUTを設定した場合には「連続減衰についてはAE、APを適用するが、粒子がECUT、PCUT以下になるとその場で計算を終了する」事になる。

EGS4では、化合物や混合物も単一のエレメントの様に扱っているため、これらのK X-rayの扱いは一般的には出来ない。近似的な方法として、K X-rayの発生に最も寄与すると考えられる元素の原子番号を指定する方法がある。例えば、この例のようなNaIの場合には、計算上問題となるK X-rayはIからのものであり、NaからのX-rayの影響は小さいため、IEDGFL(4)=53;とIの原子番号を設定している。

### 3.1 オプションの設定

EGS4 がリリースされて以降様々な改良がなされ、マクロの形で組み込まれている。多くの場合、先のIEDGFLの様に当該フラグの変数に値を設定する事により適用されるようになっている。本ユーザコードでは、適用していないが、重要なものについて、簡単に紹介する。

#### 1. 制動輻射の放出角度 [1]

デフォルトでは、クリティカルアングル ( $= m_0/E_0$  radian;  $m_0$ :電子の静止エネルギー、 $E_0$ :電子のエネルギー) に放出される様になっている。クリティカルアングル内でサンプリングするには、IBRDST=1; とすれば良い。

```

"THE FOLLOWING REPLACES THE EGS4 DEFAULT $SET-BREMS-ANGLE MACRO  "
"IT'S USE REQUIRES AN ASSOCIATE MACRO $SET-BREM-REJECTION-FUNCTION"
"DEFINED BELOW                                                    "
"                                                                    "
"USAGE:  IBRDST=0 => EGS4 DEFAULT ANGLE SELECTION                "
"         IBRDST=1 => KOCH AND MOTZ (1959) EQ. 2BS ANGLE SELECTION  "

```

このフラグを設定するためには、メインルーティンにCOMIN/BREMPRR/;が含まれていなければならない。

#### 2. 電子対生成で発生する電子・陽電子の角度分布 [2]

電子対生成で発生する電子・陽電子の放出角度についても、制動輻射と同じ扱いがされている。より、厳密な扱いを行う場合には、IPRDSTに該当する値を設定する。

```

"USAGE:  IPRDST=0 => EGS4 DEFAULT ANGLE SELECTION                "
"         IPRDST=1 => LOWEST ORDER ANGULAR DISTRIBUTION          "
"                                                                    "
"          d(Probability)          sin(theta)                      "
"          ----- = -----"

```



```

"          d(theta)          2*P[E_total - P*cos(theta)]**2      "
"
"          IPRDST=2 => MOTZ, OLSEN AND KOCH (1969) EQ. 3D-2003    "
"          IF IPRDST IS NON-ZERO AND E_PHOTON < $BHPAIR          "
"          THE IPRDST=1 DISTRIBUTION IS USED                      "

```

このフラグを設定するためには、メインルーティンに COMIN/BREMPRR/; が含まれていなければならない。

### 3. 光電子の角度分布 [3]

デフォルトでは、光電子は元の光子と同じ方向に放出される。光電子の放出角度分布を厳密に扱いたい場合には、IPHTER を設定する。IPHTER の場合は、IEDGFL と同様、リージョン毎に設定する。

```

"          IPHTER          REGION DEPENDENT ARRAY FOR SWITCHING ON  "
"          PHOTOELECTRON ANGULAR DISTRIBUTION                    "
"          DEFAULT(0)-NO SAMPLING, (1)-SAMPLING                  "

```

このフラグを設定するためには、メインルーティンに COMIN/USER/; が含まれていなければならない。

### 4. 制動輻射 Splitting[1]

制動輻射の発生頻度が低いとその寄与が重要な問題の場合には、制動輻射 Splitting を使用すると有効な場合がある。制動輻射 Splitting を適用するには、IBRSPL=1; に設定すると共に、Splitting 数,NBRSPL, を指定する必要がある。

```

"THIS MACRO PLACES ADDITIONAL BREMSSTRAHLUNG PHOTONS ON THE STACK  "
"RESETTING PARTICLE WEIGHTS TO MAKE THE GAME FAIR. THREE USER INPUTS"
"ARE REQUIRED:                                                         "
"                                                                       "
"IBRSPL = 0 => NO ADDITIONAL BREMSSTRAHLUNG PHOTONS (DEFAULT)      "
"  = 1 => PERFORM BREMSSTRAHLUNG SPLITTING                          "
"NBRSPL = NUMBER OF BREMSSTRAHLUNG PHOTONS CREATED/INTERACTION     "
"FBRSPL = 1/NBRSPL (USED TO ADJUST THE PARTICLE WEIGHTS)           "
"                                                                       "
"NBRSPL AND FBRSPL ARE CHANGED DYNAMICALLY IF STACK OVERFLOW MIGHT "
"OCCUR                                                                "
"                                                                       "
"THIS MACRO IS INVOKED AFTER THE FIRST CALL THE SUBROUTINE BREMS  "

```

このフラグを設定するためには、メインルーティンに COMIN/BREMPRR/; が含まれていなければならない。

## 4 STEP 3

SUBROUTINE HATCH を CALL して、物質データを読み込むのがこの step の機能である。使用している物質のデータ(名前、密度、放射長、エネルギーの下限、上限)の打ち出し、各リージョンの物質名、カットオフエネルギーの打ち出しは、必ずしも必要ではないが、確認のために打ち出した方がよい。ただし、リージョン数が非常に多い場合には、全てのリージョンについて打ち出すのではなく、指定した通りに物質が割り当てられているかが確認できる様な打ち出しにする等の工夫が必要である。

K-X ray を発生させる SUBROUTINE PHOTO 等が組み込まれている場合には、X ray 発生のフラグがセットされているかどうかの判断をし、セットされている場合には、必要なデータを整える SUBROUTINE EDGSET を CALL する文を入れておく。

```
DO I=1,NREG [IF(IEDGFL(I).NE.0) [CALL EDGSET(NREG); EXIT;]]
```

## 5 STEP 4

幾何形状を規定する各種のデータを設定する。この例では、円筒平板形状なので、各平板の位置と角度 (PCOORD(I, J), PNORM(I, J))、円筒の半径とその自乗 (CYRAD(I), CYRAD2(I)) を設定している。各値は、直接設定しても良いが、計算出力の打ち出し時に、形状に関する情報を出力しやすいように各平板の厚さ等を変数として定義しておく方が良い。

## 6 STEP 5

AUSGABでの使用する変数の初期化を行う。エネルギー保存のチェックや粒子の統計をとるサブルーチンである ECNSC1 や NTALLY を使用している場合には例にならってここで初期化しておく。

```
CALL ECNSV1(0,NREG,TOTKE);" INITIALIZE ESUM ARRAY FOR ENERGY"
" CONSERVATION CALCULATION."
" NREG=NUMBER OF REGIONS"
" TOTKE=TOTAL KE (DUMMY VARIABLE HERE)"
" (MUST BE REAL*8)"

CALL NTALLY(0,NREG);

NCOUNT=0; "PARTICLE HISTORY COUNTER"
ILINES=0; "INITIALIZE LINE-OUTPUT COUNTER"
DEPE=0.DO; "ZERO THE ENERGY DEPOSITION AT SCINTILATOR"
/PEF,TEF/=0.0; "Zero the efficiency"

DO J=1,$NEBIN [PH(J)=0.0;] "Zero the pulse-heght"
DO ND=1,$NDET [
DO J=1,$NEBIN [
/SPG(ND,J),SPE(ND,J),SPP(ND,J)/=0.0; "Zero the spectrum"
]]
```

上記の内、NCOUNT, ILINES は、途中結果の打ち出しの制御のための変数である。

## 7 STEP 6

以下のように入射粒子のパラメータを設定する。

```
IQI=0; "INCIDENT PARTICLE"

EI=1.33 +ABS(IQI)*PRM; "TOTAL ENERGY OF PARTICLE (MEV) "

AVAILE=EI + IQI*PRM; "AVAILABLE K.E. (MEV) (MUST BE REAL*8)"
EKIN=AVAILE;
ECUTMN=ECUT(4); EKO=EKIN; "*PRESTA*"
$PRESTA-INPUTS; "INPUT THE *PRESTA* VARIABLES"

DELTA E=0.05; "Energy bin of response"

XI=0.0; YI=0.0; ZI=0.0; "STARTING COORDINATES (CM)"
UI=0.0; VI=0.0; WI=1.0; "INCIDENT DIRECTION COSINES"
IRI=2; "ENTRANCE REGION DEFINITION"
WTI=1.0; "WEIGHT FACTOR OF UNITY"

IDINC=-1; "AN IDENTIFIER (LIKE IARG) TO MARK INCIDENT PARTICLES"

IXXST=17847465;

IXX=IXXST; "INITIALIZED RANDOM NUMBER WITH STARTING SEED"
```

```

NWRITE=10; "NUMBER OF INCIDENT CASES TO PRINT OUT"

NCASES=$NCASES; "MAXIMUM NUMBER OF INCIDENT CASES TO RUN"
NBATCH=$NBATCH; "NUMBER OF BATCH"
NCASPB=NCASES/NBATCH; "NUMBER OF CASES PER BATCH"
NOFBAT=0; "NUMBER OF BATCH FINISHED"

NLINES=15; "NUMBER OF LINES TO PRINT OUT"

```

IQI:粒子の種類、 EI:全エネルギー、 XI,YI,ZI:入射位置、  
 UI,VI,WI:入射方向の方向余弦、 IRI:入射位置のリージョン番号  
 WTI:入射粒子のウェイト (通常は 1)

ECUTMN=ECUT(4); EKO=EKIN; は、\$PRESTA-INPUTSにより PRESTAのパラメータを初期化するために必要な変数である。

入射粒子がエネルギー分布を持っている場合のスペクトル情報、方向や位置が分布している場合のその情報の設定もこの Stepで行う。

その他に、途中結果の打ち出しで入射粒子であることを示すための変数 (IDINC=-1;) や、乱数の最初のシード (IXXST)、途中結果の打ち出し条件、ヒストリー数 (NCASES) 等の設定を行う。この例の様にバッチに分けて計算する場合には、バッチ数 (NBATCH) やバッチ当りのヒストリー数の指定も行う。

## 8 STEP 7

設定したヒストリー数だけ SUBROUTINE SHOWER を CALL し、EGS4 を使用する部分である。入射粒子がエネルギー、位置、方向等について分布している場合には、CALL SHOWERの前にそれらのパラメータを決定するルーチンを入れる。位置や方向が分布しており、入射粒子毎に入射のリージョン番号が変わる場合には、IRI を忘れずに変えて置かなければならない。

### 8.1 統計誤差

$x$  をモンテカルロ計算で計算したい量 (スコアする量) とする。モンテカルロ計算の結果には、その統計誤差が必要である。モンテカルロ計算の統計誤差を評価する方法として、以下の2つの方法がある。

#### 8.1.1 MCNPで使用している方法

- ヒストリー数を  $N$  とする。
- $x_i$  を  $i$  番目のヒストリーの結果とする。
- $x$  の平均値を計算する:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \quad (1)$$

- $x_i$  の分散値を以下の式から求める。:

$$s^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \simeq \overline{x^2} - \bar{x}^2 \quad (\overline{x^2} = \frac{1}{N} \sum_{i=1}^N x_i^2). \quad (2)$$

- $\bar{x}$  の分散値は、

$$s_{\bar{x}}^2 = \frac{1}{N} s^2 \simeq \frac{1}{N} [\overline{x^2} - \bar{x}^2] \quad (3)$$

となる。

- 統計誤差として、

$$R = s_{\bar{x}}/\bar{x} \simeq \left[ \frac{1}{N} \left( \frac{\overline{x^2}}{\bar{x}^2} - 1 \right) \right]^{1/2} \quad (4)$$

を用いる。

### 8.1.2 MORSE-CG で使用している方法

- ヒストリー数を  $N$  とする。
- “ $N$ ” ヒストリーをそれぞれ  $N/n$  ヒストリーの  $n$  バッチに分割する。各バッチで得られた値を  $x_i$  とする。

- $x$  の平均値を

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (5)$$

より求める。

- 求め  $x_i$  の分散を以下の式で計算する。

$$s_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i^2 - \bar{x}^2) \quad (6)$$

- $\bar{x}$  の統計誤差は、

$$s_{\bar{x}}^2 = \frac{s_x^2}{n} \quad (7)$$

となる。

- FSD(fractional standard deviation) として

$$\text{FSD} = s_{\bar{x}}/\bar{x} \quad (8)$$

を用いる。

### 8.1.3 統計誤差評価のために行っている ucna3p.mor での処理

ucna3p.mor では MORSE-CG の方法で FSD を求めている。

全ヒストリー数を NBATCH に分け、各バッチサイクル中で NCASPB(=NCASES/NBATCH) ヒストリーの計算を行う。各ヒストリー毎の検出器中でのエネルギー吸収量に基づくレスポンス、ピーク及び全検出効率は、CALL SHOWER の後で求めると共に、関連する変数を初期化する。

```
"If some energy is deposited inside detector add pulse-height"
"and efficiency"
```

```
IF(DEPE.GT.0.DO) [
IE=DEPE/DELTAE+1;
IF(IE.LE.$NEBIN) [PH(IE)=PH(IE)+1.0;]
IF(DEPE.GE.EI*0.999) [PEF=PEF+1.0;]
TEF=TEF+1;]
DEPE=0.DO;
```

NCASPB ヒストリーが終了すると、バッチ毎の平均値 ( $x_i$ ) を求める。

```
"Calcurate average value for this BATCH"
DO IE=1,$NEBIN [
  PHPB(IE,NOFBAT)=PH(IE)/NCASPB;
  PH(IE)=0.0;
]
PEFPB(NOFBAT)=PEF/NCASPB;
TEFPB(NOFBAT)=TEF/NCASPB;
/PEF,TEF/=0.0;

DO ND=1,$NDET [
  DO IE=1,$NEBIN [
    SPGPB(ND,IE,NOFBAT)=SPG(ND,IE)/NCASPB; "Gamma spectrum for this BATCH"
    SPEPB(ND,IE,NOFBAT)=SPE(ND,IE)/NCASPB; "Electron spectrum for this BATCH"
    SPPPB(ND,IE,NOFBAT)=SPP(ND,IE)/NCASPB; "Positron spectrum for this BATCH"
    /SPG(ND,IE),SPE(ND,IE),SPP(ND,IE)/=0.0;
  ]
]
```

## 9 STEP 8

得られた結果を処理して打ち出す処理を行う。この部分での出力だけで形状や入射粒子の種類、エネルギー等がわかるようにこれらの情報についても出力するようにしておく方が良い。

バッチ毎の結果から、それぞれの量の平均値及び  $S_{\bar{x}}$  を求め、出力する。検出器のレスポンスや、粒子のスペクトルの表現にはいろいろな方法がある。例では、線源光子 1 個あたりの MeV 当たりに対応や粒子数にしているため、エネルギービン幅である DELTAE で割った結果を出力している。

```
"Calculate average and its deviation"

/AVPE,DESCI2/=0.0;
DO J=1,NBATCH [
  AVPE=AVPE+PEFPB(J)/NBATCH;
  DESCI2=DESCI2+PEFPB(J)*PEFPB(J)/NBATCH;
]
SIGPE=SQRT((DESCI2-AVPE*AVPE)/(NBATCH-1));
AVPE=AVPE*100.0;
SIGPE=SIGPE*100.0;
OUTPUT AVPE,SIGPE;(' Peak efficiency =',G15.5,'+-',G15.5,' %');

/AVTE,DESCI2/=0.0;
DO J=1,NBATCH [
  AVTE=AVTE+TEFPB(J)/NBATCH;
  DESCI2=DESCI2+TEFPB(J)*TEFPB(J)/NBATCH;
]
SIGTE=SQRT((DESCI2-AVTE*AVTE)/(NBATCH-1));
AVTE=AVTE*100.0;
SIGTE=SIGTE*100.0;
OUTPUT AVTE,SIGTE;(' Total efficiency =',G15.5,'+-',G15.5,' %');

OUTPUT ;('/ Pulse height distribution ');
DO IE=1,$NEBIN [
  ELOW=DELTAIE*(IE-1);
  EUP=DELTAIE*IE;
  IF(ELOW.GT.EKIN) [EXIT;]

/AVPH,DESCI2/=0.0;
DO J=1,NBATCH [
  AVPH=AVPH+PHPB(IE,J)/NBATCH;
  DESCI2=DESCI2+PHPB(IE,J)*PHPB(IE,J)/NBATCH;
]
```

```

SIGPH=SQRT((DESCI2-AVPH*AVPH)/(NBATCH-1));
AVPH=AVPH/DELTAЕ;
SIGPH=SIGPH/DELTAЕ;
OUTPUT EUP,AVPH,SIGPH;
(' E (upper-edge --',G10.4,' MeV )=' ,G15.5,'+-',G15.5,
' counts/MeV/incident');
]

```

ECNSV1 や NTALLY を使用している場合には、それらからの出力を行う文が必要である。

## 10 SUBROUTINE AUSGAB

AUSGAB は、ユーザが求める情報をスコアするサブルーチンである。例では、NaI 検出器部での吸収エネルギー及び NaI の部分に入射する各粒子のスペクトルを求めている。

```

IF(MED(IRL).EQ.1) ["particle is inside the detector"
DEPE=DEPE+EDEP*DPWT; "Add energy deposition"
IF(IRL.NE.IROLD.AND.IARG.EQ.0) [ "particle enters into detector"
IF(IQ(NP).EQ.0) ["photon"
IE=E(NP)/DELTAЕ+1;
IF(IE.LE.$NEBIN) [SPG(1,IE)=SPG(1,IE)+DPWT;]]
ELSEIF(IQ(NP).EQ.-1) ["Electron"
IE=(E(NP)-RM)/DELTAЕ;
IF(IE.LE.$NEBIN) [SPE(1,IE)=SPE(1,IE)+DPWT;]]
ELSE ["Positron"
IE=(E(NP)-RM)/DELTAЕ;
IF(IE.LE.$NEBIN) [SPP(1,IE)=SPP(1,IE)+DPWT;]]
] "end of entering to detector"
] "end of inside detector"

```

粒子が検出器内かどうかの判断を最初に行っている

```
IF(MED(IRL).EQ.1) ["particle is inside the detector"
```

の代わりに

```
IF(IRL.EQ.8) ["particle is inside the detector"
```

としても良いが、平板等の数を変更する等により NaI のリージョン番号が変わる度に変更が必要となるので、先のやり方の方が良い。

```
IF(IRL.NE.IROLD.AND.IARG.EQ.0) [ "particle enters into detector"
```

が、粒子がある平面を横切ったかどうかの判定の方法である。この表現では、ギャップ部(リージョン 5 から NaI に入る粒子と共に、後方の quartz 窓等から後方散乱されてくる粒子も記録される。機器の前面からの粒子のみを求める場合には、

```
IF((IRL.NE.IROLD.AND.IARG.EQ.0).AND.(IROLD.EQ.5))
[ "particle enters into detector"
```

とすれば良い。

```
"KEEP TRACK OF THE ENERGY DEPOSITION---FOR CONSERVATION PURPOSES"
ESUM(IQ(NP)+2,IRL,IARG+1)=ESUM(IQ(NP)+2,IRL,IARG+1)+EDEP*DPWT;
NSUM(IQ(NP)+2,IRL,IARG+1)=NSUM(IQ(NP)+2,IRL,IARG+1) + 1;
```

は、ECNSV1 と NTALLY のために必要な処理であり、

```
IF(NCOUNT.LE.NWRITE.AND.ILINES.LE.NLINES) [
OUTPUT E(NP),X(NP),Y(NP),Z(NP),U(NP),V(NP),W(NP),
IQ(NP),IRL,IARG; (7G15.7,3I5);
ILINES=ILINES+1;]
```

は、途中結果を打ち出す処理である。

## 11 SUBROUTINE HOWFAR

HOWFARは、粒子の進行方向でのリージョン境界までの距離を計算し、反応点までの距離との比較をし、境界までの距離の方が短い場合には粒子の移動距離を境界までの距離に置き換え、リージョンが変わるという処理を行う。

その他に、HOWFARでは、ユーザが粒子の追跡を止める設定を行う。(IDISC=1;) 通常は、粒子が、検討している領域の外に出て追跡を終了する場合にこの設定を行う。

```
IRL=IR(NP); "SET LOCAL VARIABLE"

IF(IRL.LE.1.OR.IRL.GE.IRZ+2) [IDISC=1; RETURN;]

NSLAB=(IRL-2)/NCYL + 1 ; "SLAB NUMBER"
NANNU=IRL-1-NCYL*(NSLAB-1); "ANNULUS NUMBER"
NPL1=NSLAB+1; NPL2=NSLAB;
IF(NSLAB.LT.NPLAN-1) [NRG1=IRL+NCYL;]
ELSE [NRG1=IRZ+2;]
IF(NSLAB.GT.1) [NRG2=IRL-NCYL;]
ELSE [NRG2=1;]

$PLAN2P(NPL1, NRG1, 1, NPL2, NRG2, -1);

IF(NANNU.LT.NCYL) [NRG2=IRL+1;]
ELSE [NRG2=IRZ+3;]
IF(NANNU.GT.1) [NRG1=IRL-1; NCL2=NANNU;]
NCL1=NANNU-1;
$CYL2(NCL1, NRG1, NCL2, NRG2); RETURN;]

$CYLNDR(1, 1, IHIT, TCYL);
IF(IHIT.EQ.1) [
$CHGTR(TCYL, NRG2);]

RETURN;
END; "END OF SUBROUTINE HOWFAR"
```

## 参考文献

- [1] A. F. Bielajew, R. Mohan and C. S. Chui, "Improved bremsstrahlung photon angular sampling in the EGS4 code system," National Reserach Council of Canada Report PIRS-0203(1989).
- [2] A. F. Bielajew, "Improved angular sampling for pair production in the EGS4 code system," National Reserach Council of Canada Report PIRS-0287(1991).
- [3] A. F. Bielajew and D. W. O. Rogers, "Photoelectron angular distribution in the EGS4 code system," National Reserach Council of Canada Report PIRS-0058(1986).

# EGS4におけるジオメトリの扱い

## 1 SUBROUTINE HOWFAR の機能

EGS4 ユーザコードには、

- 求めたい量を記録する SUBROUTINE AUSGAB
- ジオメトリに関する情報を EGS に伝える SUBROUTINE HOWFAR

が必要である。

この内 HOWFAR について理解するには文献 1-3 が有益である。本小冊子の記述の基本的な部分は文献 3 によっている。

SUBROUTINE HOWFAR で重要な変数は、USTEP, IDISC, IRNEW である。これらの変数は、COMMON /EPCONT/ に含まれている。

EGS4 では、計算対象となる領域を「リージョン」と呼んでいる。リージョン毎に物質が異なる場合もあるし、同じ物質の領域がいくつかのリージョンに分かれている場合もある。HOWFAR の機能は、粒子の進行方向でのリージョン境界までの距離を計算し、断面積から決められた粒子の飛程距離 (USTEP) と比較し、反応点が現在のリージョン内にあるかどうかの判定を行う事である。

境界までの距離の方が小さい場合には、新しいリージョン番号を IRNEW とし、粒子を境界まで移動させる。従って、HOWFAR で最も重要な事は、リージョン境界までの距離を計算する事である。当然の事ながら、計算方法は使用する幾何形状により異なる。

粒子の追跡を止めたい領域 (discard region) に粒子が達した場合等のようにユーザの判断でその粒子の追跡を止めたい時には IDISC を 1 にすることにより、以後の追跡を止める (USER DISCARD) 事ができる。これも HOWFAR の重要な機能である。

以下では EGS4 で標準的に用いている Geometry マクロ の紹介と、それらを NaI(Tl) 検出器のピーク検出効率とレスポンスを計算するユーザコードに適用した例を通じて EGS4 での幾何形状の扱い方を紹介する。

EGS4 システムに含まれている geometry SUBROUTINE 及び マクロ を第 1 表に示す。SUBROUTINE と マクロ は機能としては同じであるが、マクロの方がインラインであるために計算時間が短い。

## 2 \$PLANE1 マクロ

```
$PLANE1(NPLAN,ISIDE,IHIT,TPLN);  
NPLAN: ID number of plane to be checked.  
ISIDE: 1 region is between origin and outer normal.  
       -1 region is not between origin and outer normal.  
IHIT: 1 means particle trajectory will hit plane.  
       2 means particle trajectory is parallel to plane.  
       0 means particle trajectory is away from plane.  
TPLN: distance to plane if IHIT=1.
```

平面は、(PNORM(I,J),I=1,3)(normal vector) と (PCOORD(I,J),I=1,3) (normal vector と平面の交点の座標) で定義する。両変数は、COMMON/PLADTA/ に含まれている。

図 1 の様に、2 つの平行な平面で囲まれた 3 つのリージョンを考える。それぞれのリージョンのリージョン番号は、22, 23, 24 であり、平面の番号は 6, 7 である。この場合、PNORM と PCOORD は、次のように定義する。



PCOORD(1,6)=0.0; PCOORD(2,6)=0.0; PCOORD(3,6)=30.0;  
 PNORM(1,6)=0.0; PNORM(2,6)=0.0; PNORM(3,6)=1.0;

PCOORD(1,7)=0.0; PCOORD(2,7)=0.0; PCOORD(3,7)=45.0;  
 PNORM(1,7)=0.0; PNORM(2,7)=0.0; PNORM(3,7)=1.0;

SUBROUTINE 名	機 能	MACRO
PLANE1	Determines if the particle trajectory strikes a plane surface Returns trajectory distance (TPLN).	\$PLANE1
CYLNDR	Determines if the particle trajectory strikes a cylindrical surface. Returns trajectory distance (TCYL).	\$CYLNDR
CONE	Determines if the particle trajectory strikes a conical surface. Returns trajectory distance (TCONE).	\$CONE
SPHERE	Determines if the particle trajectory strikes a spherical surface. Returns trajectory distance (TSPH).	\$SPHRE
CHGTR	Change USTEP and IRNEW whenever USTEP is larger than the trajectory distance (TPLN, TCYL, TCONE, TSPH).	\$CHGTR
FINVAL	Determines the coordinates of the particle trajectory at the point of an intersection with a given surface.	\$FINVAL
PLAN2P	Determines the intersection point for two parallel planes by calling PLANE1 twice (when necessary) and CHGTR if a plane is hit.	\$PLAN2P
PLAN2X	Determines the intersection point for two crossing planes by calling PLANE1 twice (always) and CHGTR if a plane is hit.	\$PLAN2X
CYL2	Similar to PLAN2P, but for concentric cylinders.	\$CYL2
SPH2	Similar to PLAN2P, but for concentric spheres	\$SPH2
CON2	Similar to PLAN2P, but for concentric cone.	\$CON2

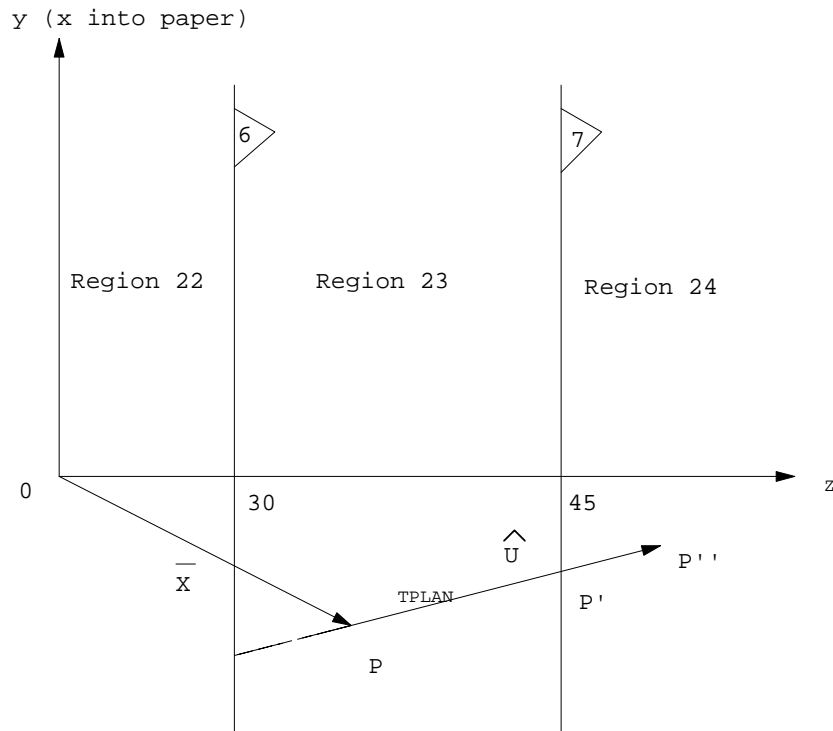


図 1

粒子が、23 以外のリージョンに移動した時は、追跡を終了するとした場合の、HOWFAR は、次のように書くことができる。

```

SUBROUTINE HOWFAR;
COMIN/EPCONT,PLADTA,STACK/;

IRL=IR(NP);
IF(IRL.NE.23) [IDISC=1; "Discard particles outside region 23"]
ELSE ["Track particles within region 23"
  $PLANE1(7,1,IHIT,TPLN); "Check upstream plane first"
  IF(IHIT.EQ.1) ["Surface is hit---make change if necessary"
    $CHGTR(TPLN,24);]
  ELSEIF (IHIT.EQ.0) ["Heading backwards"
    $PLANE1(6,-1,IHIT,TPLN); "To get TPLN-value (IHIT=1, amust)"
    $CHGTR(TPLN,22); "Make changes if necessary"
  ]
]
RETURN;
END;

```

\$CHGTR は、リージョン境界までの距離と、USTEP を比較し、USTEP の方が大きい場合には、USTEP を 境界までの距離に置き換え、IRNEW を次のリージョンに置き換えるという一連の処理をするマクロである。

```

REPLACE{$CHGTR(,#,#);} WITH
  {;IF({P1}.LE. USTEP) [USTEP={P1}; IRNEW={P2};]}

```

### 3 \$PLAN2P マクロ

\$PLAN2P マクロを使用すると先の HOWFAR を以下のように簡略化する事ができる。

```

SUBROUTINE HOWFAR;
COMIN/EPCONT,PLADTA,STACK/;

IRL=IR(NP);
IF(IRL.NE.23) [IDISC=1; "Discard particles outside region 23"]
ELSE ["Track particles within region 23"
  $PLAN2P(7,24,1,6,22,-1);
]
RETURN;
END;

```

\$PLAN2P の各パラメータの意味は、以下の通りである。

```

$PLAN2P(NPL1,NRG1,ISIDE1,NPL2,NRG2,ISIDE2);
  NPL1: ID number of first plane to be checked.
  NRG1: region to go into if first plane is intersected by particle.
  ISIDE1: 1 or -1 (same with ISIDE in PLANE1)
  NPL2: ID number of second plane to be checked.
  NRG2: region to go into if second plane is intersected by particle.
  ISIDE2: 1 or -1 (same with ISIDE in PLANE1)

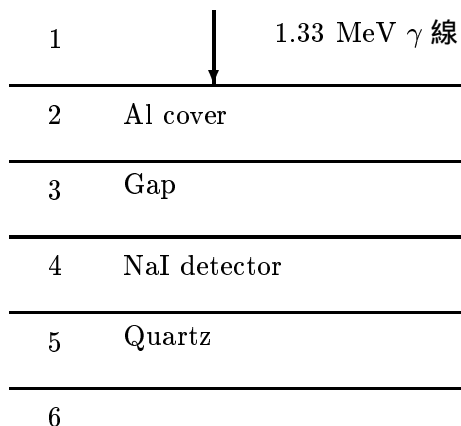
```

2 つの平板が平行でない場合には、\$PLAN2P の代わりに \$PLAN2X を使用する。

### 3.1 Multislab 形状

\$PLAN2P の使用により、多数の平行平板からなる形状を扱う事ができる。

#### 3.1.1 半無限平板形状



第 2 図 半無限平板形状

第 2 図に示すよう半無限平板の組み合わせからなる NaI 検出器を考える。NaI 検出器の前には、厚さ 1 mm のアルミニウムカバーがあり、カバーと検出器の間には 5 mm のギャップがある。この領域は真空とする。検出器の後ろには、5mm の厚さの石英ガラスがある。

各平板を以下のように定義する。

```
"DEFINE VARIOUS THICKNESSES/DISTANCES"

TCOV=0.1;  "Thickness of Al case in cm "
TGAP=0.5;  "Gap between case and detector in cm"
TDE=7.62;  "Thickness of detector in cm"
TQUARTZ=0.5;"Thickness of quartz window in cm"

"DEFINITION OF PLANES"
"SET ALL COORDINATES AND NORMALS TO ZERO TO BEGIN WITH"
DO J=1,NPLAN [
  PCOORD(1,J)=0.0;  PCOORD(2,J)=0.0;  PCOORD(3,J)=0.0;
  PNORM(1,J)=0.0;  PNORM(2,J)=0.0;  PNORM(3,J)=1.0;
]

"NOW PUT IN THE EXCEPTIONS"

PCOORD(3,2)=PCOORD(3,1)+TCOV;
PCOORD(3,3)=PCOORD(3,2)+TGAP;
PCOORD(3,4)=PCOORD(3,3)+TDE;
PCOORD(3,5)=PCOORD(3,4)+TQUARTZ;

この例では平面は X-Y 平面に平行なので、PNORM(3,J) のみが 1.0 で他は 0.0 である。
HOWFAR は、以下のように簡単に記述できる。

SUBROUTINE HOWFAR;
;COMIN/DEBUG,EPCONT,PLADTA,STACK/;
COMMON/PASSIT/NREG,NPLAN;

IRL=IR(NP);  "SET LOCAL VARIABLE"

IF(IRL.LE.1.OR.IRL.EQ.NREG) [IDISC=1; RETURN;]
```

```

NPL1=IRL; NPL2=IRL-1;
NRG1=IRL+1;
NRG2=IRL-1;

$PLAN2P(NPL1, NRG1, 1, NPL2, NRG2, -1);

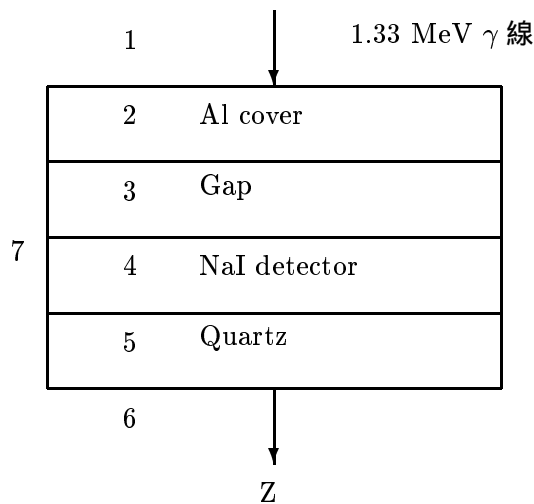
RETURN;
END; "END OF SUBROUTINE HOWFAR"

```

半無限平板形状のユーザコードとして、ucnail.mor と ucnailp.mor が、サンプルユーザコードに含まれている。ucnailp.mor は、PRESTA を用いていること以外は ucnail.mor と同じである。

### 3.1.2 有限平板形状

X 及び Y 方向も有限な第 3 図に示す形状を考える。Z 方向は変わらないが、X 及び Y 方向にそれぞれ 2 つの平面を設定する。その有限平板の外側をリージョン”7”とする。



第 3 図 ucnai2.mor の形状

平面の定義は次のように行う。

```

"DEFINE VARIOUS THICKNESSES/DISTANCES"

TCOV=0.1; "Thickness of Al case in cm "
TGAP=0.5; "Gap between case and detector in cm"
TDE=7.62; "Thickness of detector in cm"
TQUARTZ=0.5; "Thickness of quartz window in cm"
XHALF=3.81; "Half width, X-direction in cm"
YHALF=3.81; "Half width, Y-direction in cm"

"DEFINITION OF PLANES"

"SET ALL COORDINATES AND NORMALS TO ZERO TO BEGIN WITH"
DO J=1,NPLAN [
  PCOORD(1,J)=0.0; PCOORD(2,J)=0.0; PCOORD(3,J)=0.0;
  PNORM(1,J)=0.0; PNORM(2,J)=0.0; PNORM(3,J)=0.0;
]

"NOW PUT IN THE EXCEPTIONS"

"Z-direction"

/PNORM(3,1),PNORM(3,2),PNORM(3,3),PNORM(3,4),PNORM(3,5)/=1.0;
PCOORD(3,2)=PCOORD(3,1)+TCOV;

```

```
PCOORD(3,3)=PCOORD(3,2)+TGAP;
PCOORD(3,4)=PCOORD(3,3)+TDE;
PCOORD(3,5)=PCOORD(3,4)+TQUARTZ;
```

```
"Y-direction"
/PNORM(2,6),PNORM(2,7)/=1.0;
PCOORD(2,6)=-YHALF;
PCOORD(2,7)=YHALF;
```

```
"X-direction"
/PNORM(1,8),PNORM(1,9)/=1.0;
PCOORD(1,8)=-XHALF;
PCOORD(1,9)=XHALF;
```

X, Y 方向の平面を定義するために、それぞれの方向の半値幅 (XHALF, YHALF) を与える。PNORM は、Z 方向の平面 (平面 1 ~ 5) では PNORM(3, J) が、Y 方向の平面 (平面 6, 7) では PNORM(2, J) が、X 方向の平面 (平面 8, 9) では PNORM(1, J) が 1.0 である。

有限平板形状の HOWFAR は、次のようになる。

```
SUBROUTINE HOWFAR;
;COMIN/DEBUG,EPCONT,PLADTA,STACK/;
COMMON/PASSIT/NREG,NPLAN;

IRL=IR(NP); "SET LOCAL VARIABLE"

IF (IRL.LE.1.OR.IRL.GE.NREG-1) [IDISC=1; RETURN;]

NPL1=IRL; NPL2=IRL-1;
NRG1=IRL+1;
NRG2=IRL-1;

$PLAN2P(NPL1,NRG1,1,NPL2,NRG2,-1);

$PLAN2P(7,7,1,6,7,-1);

$PLAN2P(9,7,1,8,7,-1);
```

新たに計算を終了する ( IDISC=1 をセットする ) リージョンが増えたので、

```
IF (IRL.EQ.1.OR.IRL.EQ.NREG) [IDISC=1;RETURN;]
```

が

```
IF (IRL.EQ.1.OR.IRL.GE.NREG-1) [IDISC=1;RETURN;]
```

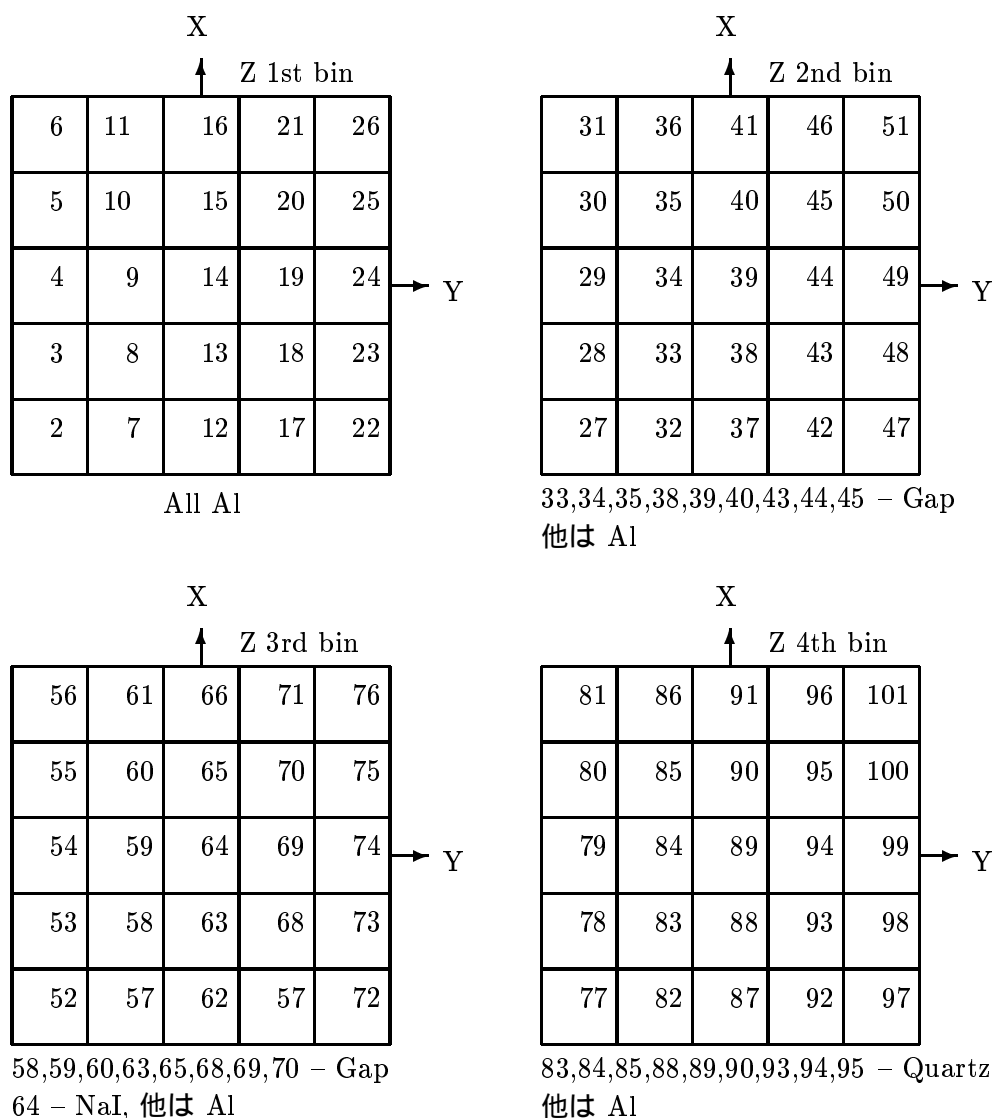
に変更されている。又、Y 方向、X 方向について調べるマクロが追加されている。ここで考慮している体系は、X, Y の平面の外はすべてリージョン 7 (NREG) となっているので、どの面にたいしても次のリージョンは 7 としている。

(ucnai2.mor 及び ucnaip.mor 参照)

### 3.1.3 3-Dimensional Cartesian Geometry

\$PLAN2P を使うことにより、3次元の形状を扱う事も可能である。X, Y, Z 方向の平面で構成される 3-dimensional cartesian geometry を扱う例を示す。リージョン番号を第 7 図の様に割り当てる。一般に、この形状はリージョンへの物質の割当などが複雑になるので、頭の中だけで考え

るのではなく実際に7図の様な絵を書いて番号を割り当てる方がよい。リージョン番号1は、Z方向の最初の平面の手前、Z方向最後の平面の先が IXYZ+2、Y方向最初の平面の手前が IXYZ+3、Y方向最後の平面の先が IXYZ+4、X方向最初の平面の手前が IXYZ+5、X方向最後の平面の先が IXYZ+6(NREG)である。



第4図 ucna14.mor の形状

各リージョンの物質指定も複雑になる。第4図の場合の設定例を示す。

```
PARAMETER $NXBIN=5; "NUMBER OF BINS IN X-DIRECTION"
PARAMETER $NYBIN=5; "NUMBER OF BINS IN Y-DIRECTION"
"$NXBIN and $NYBIN must be odd"
PARAMETER $NZBIN=4; "NUMBER OF BINS IN Z-DIRECTION"
PARAMETER $MXREG=106; "Number of region"
```

平板数、リージョン数と各リージョンの物質を次のように割り当てる。

```
NPLAN=$NXBIN+$NYBIN+$NZBIN+3; "ACTUAL NUMBER OF PLANES"
```

```
IXY=$NXBIN*$NYBIN; IXYZ=IXY*$NZBIN;
```

```
NREG=IXYZ+6; "ACTUAL NUMBER OF REGION USED"
```

```

"SET MEDIUM INDEX FOR EACH REGION"
DO I=1,NREG [MED(I)=0;] "Set all region to vacuum at first"

DO IZP=1,$NZBIN [
DO IYP=1,$NYBIN [
DO IXP=1,$NXBIN [
IIZP=(IZP-1)*IXY+(IYP-1)*$NXBIN+IXP+1;
IF(IZP.EQ.1) [MED(IIZP)=2; "Al region"
ECUT(IIZP)=0.561;]
ELSEIF(IZP.EQ.$NZBIN) [MED(IIZP)=2; "set to Al at first"
IF((IXP.GT.1.AND.IXP.LT.$NXBIN).AND.(IYP.GT.1.AND.IYP.LT.$NYBIN)) [
MED(IIZP)=3; "Quartz region"
ECUT(IIZP)=0.561;]
ELSEIF(IZP.EQ.2) [
IF(IYP.EQ.1.OR.IYP.EQ.$NYBIN.OR.(IXP.EQ.1.OR.IXP.EQ.$NXBIN)) [
MED(IIZP)=2; "Al region"
ECUT(IIZP)=0.561;]]
ELSE [
IF(IYP.EQ.1.OR.IYP.EQ.$NYBIN.OR.(IXP.EQ.1.OR.IXP.EQ.$NXBIN)) [
MED(IIZP)=2; "Al region"
ECUT(IIZP)=0.561;]
ELSE [
IF((IYP.GE.3.AND.IYP.LE.$NYBIN-2).AND.
(IXP.GE.3.AND.IXP.LE.$NXBIN-2)) [
MED(IIZP)=1; "NaI(Tl) detector region"
IEDGFL(IIZP)=53; "53:Atomic number of I"
" O:K-X ray of I is not produced"
]]]
]]]

```

各平面を以下のように定義する。

```

"DEFINE VARIOUS THICKNESSES/DISTANCES"

TCOV=0.1; "Thickness of Al case in cm "
TGAP=0.5; "Gap between case and detector in cm"
TDE=7.62; "Thickness of detector in cm"
TQUARTZ=0.5;"Thickness of quartz window in cm"
XWIDTH=7.62; "Width, X-direction in cm"
YWIDTH=7.62; "Width, Y-direction in cm"
"DEFINITION OF PLANES"

"SET ALL COORDINATES AND NORMALS TO ZERO TO BEGIN WITH"
DO J=1,NPLAN [
PCOORD(1,J)=0.0; PCOORD(2,J)=0.0; PCOORD(3,J)=0.0;
PNORM(1,J)=0.0; PNORM(2,J)=0.0; PNORM(3,J)=0.0;
]

ID1=$NZBIN+1; ID2=ID1+1; ID3=ID2+$NYBIN;
ID4=ID3+1; ID5=ID4+$NXBIN;
DO I=1,ID1 [PNORM(3,I)=1.0;]
DO I=ID2,ID3 [PNORM(2,I)=1.0;]
DO I=ID4,ID5 [PNORM(1,I)=1.0;]

"NOW PUT IN THE EXCEPTIONS"

"Z-direction"

/PNORM(3,1),PNORM(3,2),PNORM(3,3),PNORM(3,4),PNORM(3,5)/=1.0;
PCOORD(3,2)=PCOORD(3,1)+TCOV;
PCOORD(3,3)=PCOORD(3,2)+TGAP;
PCOORD(3,4)=PCOORD(3,3)+TDE;
PCOORD(3,5)=PCOORD(3,4)+TQUARTZ;

```

"Y-direction"

```
PCOORD(2, ID2)=-TCOV-TGAP-0.5*YWIDTH;  
PCOORD(2, ID2+1)=PCOORD(2, ID2)+TCOV;  
PCOORD(2, ID2+2)=PCOORD(2, ID2+1)+TGAP;  
PCOORD(2, ID2+3)=PCOORD(2, ID2+2)+YWIDTH;  
PCOORD(2, ID2+4)=PCOORD(2, ID2+3)+TGAP;  
PCOORD(2, ID2+5)=PCOORD(2, ID2+4)+TCOV;
```

"X-direction"

```
PCOORD(1, ID4)=-TCOV-TGAP-0.5*XWIDTH;  
PCOORD(1, ID4+1)=PCOORD(1, ID4)+TCOV;  
PCOORD(1, ID4+2)=PCOORD(1, ID4+1)+TGAP;  
PCOORD(1, ID4+3)=PCOORD(1, ID4+2)+XWIDTH;  
PCOORD(1, ID4+4)=PCOORD(1, ID4+3)+TGAP;  
PCOORD(1, ID4+5)=PCOORD(1, ID4+4)+TCOV;
```

平面番号は、Z, Y, X の順に割り当てる。Z 方向の平面は PNORM(3, J) が、Y 方向の平面は PNORM(2, J) が、X 方向の平面は PNORM(1, J) が 1.0 となるようにする。この例では、X-Y 平面の中心に Z 軸が来るようにしているが、これは必ずしも必要な事ではない。

以上のように定義した形状を扱う SUBROUTINE HOWFAR は、次のようになる。

```
SUBROUTINE HOWFAR;  
;COMIN/DEBUG, EPCONT, PLADTA, STACK/;  
COMMON/PASSIT/NREG, NPLAN, IXY, IXYZ;  
  
IRL=IR(NP); "SET LOCAL VARIABLE"  
  
IF(IRL.EQ.1.OR.IRL.GE.IXYZ+2) [IDISC=1; RETURN;]  
  
I=(IRL-2)/IXY+1; "SLAB NUMBER"  
J=(IRL-2-IXY*(I-1))/NXXBIN+1; "COLUMN NUMBER"  
K=IRL-1-IXY*(I-1)-NXXBIN*(J-1); "ROW NUMBER"  
  
NPL1=I+1; NPL2=I;  
IF(I.LT.NNZBIN) [NRG1=IRL+IXY;] ELSE [NRG1=IXYZ+2;]  
IF(I.GT.1) [NRG2=IRL-IXY;] ELSE [NRG2=1;]  
$PLAN2P(NPL1, NRG1, 1, NPL2, NRG2, -1);  
  
NPL2=NXXBIN+1+J; NPL1=NPL2+1;  
IF(J.LT.NNYBIN) [NRG1=IRL+NXXBIN;] ELSE [NRG1=IXYZ+3;]  
IF(J.GT.1) [NRG2=IRL-NXXBIN;] ELSE [NRG2=IXYZ+4;]  
$PLAN2P(NPL1, NRG1, 1, NPL2, NRG2, -1);  
  
NPL2=NXXBIN+NNYBIN+2+K; NPL1=NPL2+1;  
IF(K.LT.NNXBIN) [NRG1=IRL+1;] ELSE [NRG1=IXYZ+5;]  
IF(K.GT.1) [NRG2=IRL-1;] ELSE [NRG2=IXYZ+6;]  
$PLAN2P(NPL1, NRG1, 1, NPL2, NRG2, -1);  
  
RETURN;  
END;
```

粒子を囲む Z, Y, X の各 2 つの平面までの距離を調べ最短のものと USTEP との比較を行う。追跡を終了するのは、いずれか方向の最初または最後の平面の外にでた場合であるので、

```
IF(IRL.EQ.1.OR.IRL.GE.IXYZ+2) [IDISC=1; RETURN;]
```

となっている。

(具体的な例としては、ucnai4.mor 及び ucna4p.mor を参照)



## 4 \$CYLNDR 及び \$CYL2 マクロ

中心軸が Z-軸である円筒を扱うマクロは、

```
$CYLNDR(ICYL,ISIDE,IHIT,TCYL);  
  ICYL: ID number of cylinder to be checked.  
  ISIDE: 1 means particle is inside cylinder.  
         0 means particle is outside cylinder.  
  IHIT: 1 means particle intersects surface.  
        0 means particle misses surface.  
  TCYL: distance to surface if IHIT=1.
```

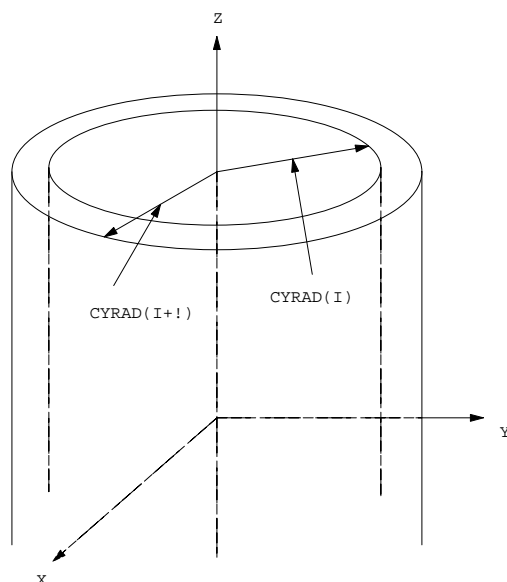
である。\$CYLNDR マクロを使用する場合には、CYRAD(I) 及び CYRAD2(I) を定義して置かなければならない。CYRAD は第 5 図に示すように、円筒の半径、CYRAD2 は、半径の 2 乗でいづれも COMOON/CYLDTA; に含まれている。

\$CYLNDR1 実行の結果 IHIT=1 が得られた場合には、\$CHGTR を使用して処理を行うことは \$PLANE1 の場合と同じである。

\$PLAN2P 等に対応する粒子が 2 つの円筒に挟まれている場合に使用できるマクロとして \$CYL2 がある。

```
$CYL2(NCY1,NRG1,NCY2,NRG2);  
  NCY1: ID number of first cylinder to be checked.  
        particle must be outside the first cylinder.  
  NRG1: region to go into if first cylinder is intersected by particle.  
  NCY2: ID number of second cylinder to be checked.  
        particle must be inside the second cylinder.  
  NRG2: region to go into if second cylinder is intersected by particle.
```

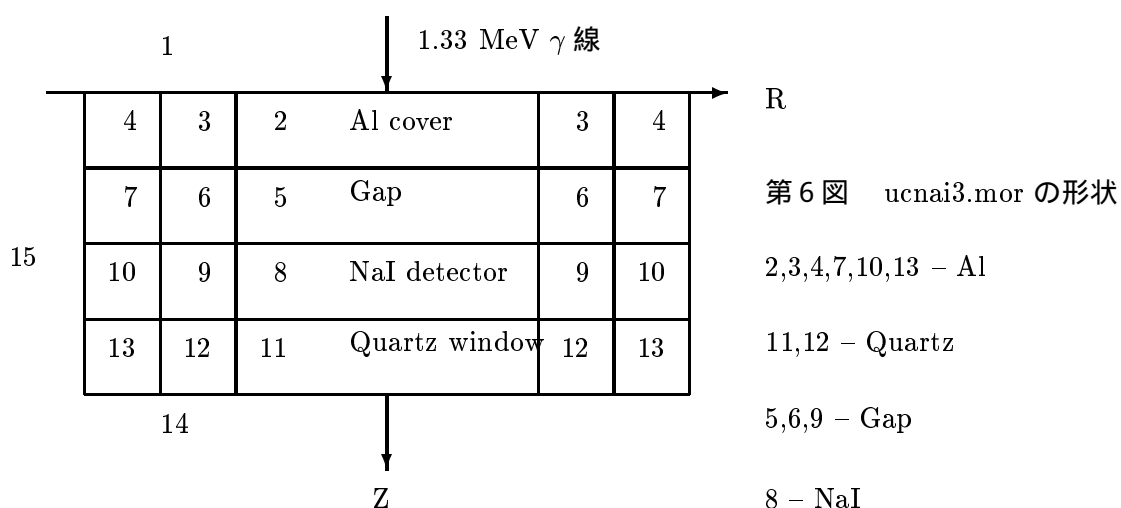
\$PLAN2P の場合と異なり、\$CYL2 の場合は、最初に現在に粒子より内側の円筒について調べ、その後粒子の外側の円筒を調べるようになっている事に注意する必要がある。



$$\text{CYRAD2(I)} = \text{CYRAD(I)} * \text{CYRAD(I)};$$

図 5

## 4.1 円筒平板形状



円筒と平板の組み合わせである円筒平板形状は、最も良く使用される形状である。第 6 図に示すように、Z-方向の 5 つの平面、R- 方向の 3 つの円筒により構成される形状を考える。平面、円筒の数を次の様に指定する。

```
PARAMETER $MXPLNS=5; "NUMBER OF PLANE"
PARAMETER $MXCYLS=3; "NUMBER OF CYLINDER"
```

円筒を使用するので、円筒の数を指定するマクロが追加されている。

円筒に関するジオメトリ データを含む COMIN/CYLDTA; を COMIN に追加する。COMMON/CYLDTA; には円筒の半径 (CYRAD(\$MXCYLS)) 及びその 2 乗 (CYRAD2(\$MXCYLS)) が含まれている。

```
;COMIN/DEBUG, BOUNDS, BREMPR, CYLDTA, EDGE, ELECIN, ETALY1, MEDIA, MISC,
NTALY1, PLADTA, RANDOM, STACK, THRESH, UPHIOT, USEFUL, USER/;
```

各リージョンの物質を次のように割り当てる。

```
NPLAN=$MXPLNS; "NUMBER OF PLANES"
NCYL=$MXCYLS; "NUMBER OF CYLINDER"

NREG=(NPLAN-1)*NCYL+3; "NUMBER OF REGIONS (INCLUDING OUTSIDE VACUUM"
" REGION) "
IRZ=NREG-3;
"SET MEDIUM INDEX FOR EACH REGION"

/MED(1),MED(NREG-1),MED(NREG)/=0; "VACUUM REGIONS"
/MED(2),MED(3),MED(4),MED(7),MED(10),MED(13)/=2;
" Al region"
/MED(11),MED(12)/=3; "Quartz window region"
MED(8)=1; "NaI(Tl) detector region"

/ECUT(2),ECUT(3),ECUT(4),ECUT(7),ECUT(8),ECUT(10)/=0.561;
/ECUT(11),ECUT(12)/=0.561;

/MED(5),MED(6),MED(9)/=0; "Vacuum region inside case"
```

平面の定義は、半無限平板や有限平板の場合と同じである、円筒は、次のように定義する。

```

"DEFINE THE CYLINDER RADII"

RDET=3.81; "Radius of detector in cm"
RGAP=0.5; "Gap between detector and case in cm"
RTCOV=0.1; "Cover thickness in cm"

CYRAD(1)=RDET;
CYRAD(2)=CYRAD(1)+RGAP;
CYRAD(3)=CYRAD(2)+RTCOV;

"PRINT OUT THE CYLINDER RADII"
OUTPUT ;(///,' CYLINDER RADII',/);
DO I=1,NCYL [
CYRAD2(I)=CYRAD(I)*CYRAD(I);
OUTPUT I,CYRAD(I);(' I=',I5,5X,' CYRAD(I)=',G15.5,'CM');
]

```

円筒平板形状を扱う SUBROUTINE HOWFAR は、以下のように書く。

```

SUBROUTINE HOWFAR;
;COMIN/DEBUG,CYLDTA,EPCONT,PLADTA,STACK/;
COMMON/PASSIT/NREG,NPLAN,NCYL,IRZ;

IRL=IR(NP); "SET LOCAL VARIABLE"

IF(IRL.LE.1.OR.IRL.GE.IRZ+2) [IDISC=1; RETURN;]

NSLAB=(IRL-2)/NCYL + 1 ; "SLAB NUMBER"
NANNU=IRL-1-NCYL*(NSLAB-1); "ANNULUS NUMBER"
NPL1=NSLAB+1; NPL2=NSLAB;
IF(NSLAB.LT.NPLAN-1) [NRG1=IRL+NCYL;]
ELSE [NRG1=IRZ+2;]
IF(NSLAB.GT.1) [NRG2=IRL-NCYL;]
ELSE [NRG2=1;]

$PLAN2P(NPL1,NRG1,1,NPL2,NRG2,-1);

IF(NANNU.LT.NCYL) [NRG2=IRL+1;]
ELSE [NRG2=IRZ+3;]
IF(NANNU.GT.1) [NRG1=IRL-1; NCL2=NANNU;
NCL1=NANNU-1;
$CYL2(NCL1,NRG1,NCL2,NRG2); RETURN;]

$CYLNDR(1,1,IHIT,TCYL);
IF(IHIT.EQ.1) [
$CHGTR(TCYL,NRG2);]

RETURN;
END;

```

(具体的な例としては、ucnai3.mor 及び ucnai3p.mor を参照)

## 4.2 DELCYL マクロ

\$CYLNDR には、2次曲面に最も近づきうる距離を表す DELCYL と呼ばれるパラメータが含まれている。

有限な精度の計算機においては、EGS4 で採用している 2次曲面の解法では粒子が表面にトラップされる現象のために、DELCYL の必要性が起きる。

このようなトラップ現象の発生する確率は小さいけれど、発生した場合には無限ループが生じるので問題が大きい。

この問題は、CERN の Stevenson によって指摘されたものである。\$CYLNDR に DELCYL を追加し、多くの問題に適用する値として 1.0E-4cm を採用したのも彼である。

DELCYL は、\$DELCYL マクロとして EGS4MAC (デフォルト値：1.E-4) に含まれている。

SLAC では、\$CYLNDR 中の \$DELCYL を変数 DELCYL で置き換え HOWFAR で以下の様にダイナミックの変えられるようにしている。

```
IF(CYRAD(IRL).LT.0.1) [DELCYL=1.E-6;]
ELSEIF(CYRAD(IRL).LT.1.0) [DELCYL=1.E-5;]
ELSEIF(CYRAD(IRL).LT.10.) [DELCYL=1.E-4;]
ELSE [DELCYL=CYRAD({P1})*1.E-5;]
```

KEK では、同じ事を \$CYLNDR に含むようにマクロを変更した。kekmac.mor に、変更したマクロが含まれている。

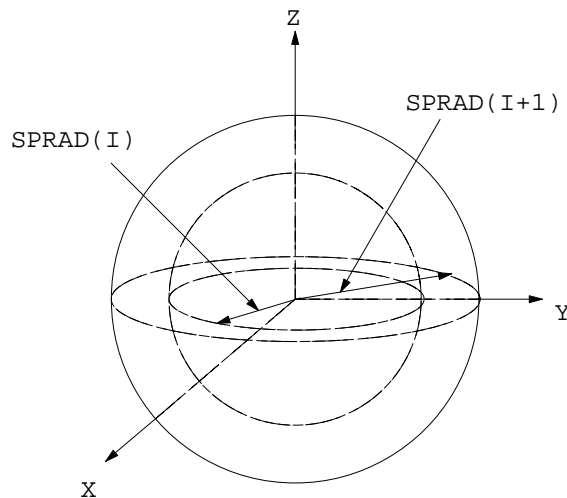
## 5 \$SPHERE 及び \$SPH2 マクロ

原点が中心である球に関連するマクロは、以下のものである。

```
$SPHERE(ISPH,ISIDE,IHIT,TSPH);
  ISPH: ID number of sphere to be checked.
  ISIDE: 1 means particle is inside sphere.
         0 means particle is outside sphere.
  IHIT: 1 means particle intersects surface.
         0 means particle misses surface.
  TSPH: distance to surface if IHIT=1.

$SPH2(NSPH1,NRG1,NSPH2,NRG2);
  NSPH1: ID number of first sphere to be checked.
         particle must be outside the first sphere.
  NRG1: region to go into if first sphere is intersected by particle.
  NCY2: ID number of second sphere to be checked.
         particle must be inside the second sphere.
  NRG2: region to go into if second sphere is intersected by particle.
```

である。\$SPHERE マクロや \$SPH2 マクロを使用する場合には、球の半径 SPRAD(I) とその二乗である SPRAD2(I) を定義して置かなければならない。(いずれも COMOON/SPHDTA; に含まれている。)



$$\text{SPRAD2}(I)=R(I)*R(I);$$

## 6 円錐

円錐は、これまでの形状と較べると複雑である。中心軸が Z-軸である円錐は、Z-軸と円錐面との角度、 $\theta_i$ , ( $COTAL(I)=\cot(\theta_i)$ ,  $COTAL2(I)=COTAL(I)*COTAL(I)$ ) と円錐の頂点の Z-座標である  $SMALL(I)$  で定義する。(第 6 図参照) 両変数は、COMMON/CONDTA に含まれている。

```

$CONE(ICONE,ISIDE,IHIT,TCONE);
  ICONE: ID number of cone to be checked.
  ISIDE: 1 means particle is inside cone.
         0 means particle is outside cone.
  IHIT: 1 means particle intersects surface.
         0 means particle misses surface.
  TCONE: distance to surface if IHIT=1.

```

2つの円錐に挟まれた粒子に対しては、

```

$CON2(NCON1,NRG1,NCON2,NRG2);
$CON21(NCON1,NRG1,NCON2,NRG2);
  NCON1: ID number of first cone to be checked.
         particle must be outside the first cone.
  NRG1: region to go into if first cone is intersected by particle.
  NCON2: ID number of second cone to be checked.
         particle must be inside the second cone for $CON2.
         particle must be outside the second cone for $CON21.
  NRG2: region to go into if second cone is intersected by particle.

```

$\$CON2$  は、第 6 図の (a) 点の場合に使用される。一方、(b) 点の領域に粒子が存在している場合には、 $\$CON21$  を使用する必要がある。

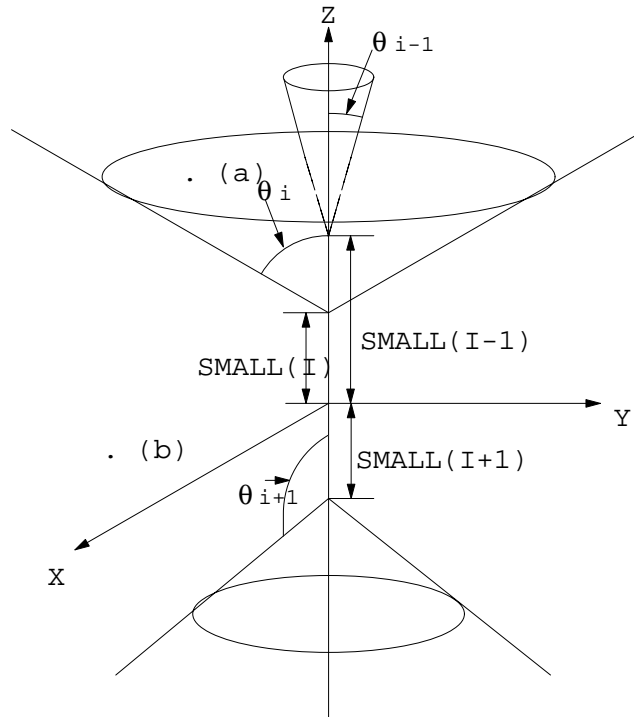


Fig. 6

## 7 \$FINVAL マクロ

その他、形状に関連する便利なマクロに  $\$FINVAL$  がある。

```

$FINVAL (DIST,XCOORD,YCOORD,ZCOORD);
DIST : the distance traveled.
XCOORD: X-coordinate after travel.
YCOORD: Y-coordinate after travel.
ZCOORD: Z-coordinate after travel.

```

\$FINVAL マクロは、DISTだけ粒子が移動した後の粒子の位置を計算するマクロである。移動後の粒子の位置から情報を引き出す場合に便利なマクロである。

## 8 ジオメトリーマクロの修正、作成

EGS4に組み込まれているジオメトリ関係のマクロはこれまでに説明したものであるが、これらのマクロを修正したり、新たなものを作る事もできる。ここでは、その1例として、中心軸のX-,Y-座標が原点からずれているような円筒を扱う様に\$CYLNDRを変更したマクロを紹介する。オリジナルな\$CYLNDRは、

```

REPLACE {$CYLNDR(##,##,##,##)} WITH
  {; IF(CYRAD({P1}).LT.0.1) [DELACYL=1.E-6;]
  ELSEIF(CYRAD({P1}).LT.1.0) [DELACYL=1.E-5;]
  ELSEIF(CYRAD({P1}).LT.10.) [DELACYL=1.E-4;]
  ELSE [DELACYL=1.E-3;]
  NOFCY={P2}; {P3}=1; {P4}=0.0; ACYL=SQRT(U(NP)*U(NP)+V(NP)*V(NP));
  IF(ACYL.EQ.0.0) [{P3}=0;] ELSE [
  BCYL=(X(NP)*U(NP)+Y(NP)*V(NP))/ACYL; CCYL=X(NP)*X(NP)+Y(NP)*Y(NP)
  -CYRAD2({P1}); ARGCY=BCYL*BCYL-CCYL;
  IF(ARGCY.LT.0.0) [{P3}=0;] ELSE [
  IF(ABS(CCYL).LT.DELACYL.AND.NOFCY.EQ.0.AND.BCYL.GE.0.0) [{P3}=0;]
  ELSEIF(ABS(CCYL).LT.DELACYL.AND.NOFCY.EQ.1.AND.BCYL.LT.0.0) [
  {P4}=-2.0*BCYL/ACYL;]
  ELSE [ IF(NOFCY.EQ.1.AND.CCYL.GE.0.0) [{P3}=1; {P4}=DELACYL;]
  ELSEIF(NOFCY.EQ.0.AND.CCYL.LE.0.0) [{P3}=1; {P4}=DELACYL;]
  ELSE [ ROOTCY=SQRT(ARGCY); IF(CCYL.LT.0.0)
  [{P4}=(-BCYL+ROOTCY)/ACYL;]
  ELSEIF(BCYL.LT.0.0) [{P4}=(-BCYL-ROOTCY)/ACYL;]
  ELSE [{P3}=0;]]]]}

```

である。中心軸のX-,Y-座標を示すXOP, YOPを導入し、以下の様に変更する事により、目的とする円筒を扱うマクロにする事ができる。

```

REPLACE {$CYLNDR(##,##,##,##)} WITH
  {; IF(CYRAD({P1}).LT.0.1) [DELACYL=1.E-6;]
  ELSEIF(CYRAD({P1}).LT.1.0) [DELACYL=1.E-5;]
  ELSEIF(CYRAD({P1}).LT.10.) [DELACYL=1.E-4;]
  ELSE [DELACYL=1.E-3;]
  NOFCY={P2}; {P3}=1; {P4}=0.0; ACYL=SQRT(U(NP)*U(NP)+V(NP)*V(NP));
  XNP=X(NP)-XOP({P1}); YNP=Y(NP)-YOP({P1});
  IF(ACYL.EQ.0.0) [{P3}=0;] ELSE [
  BCYL=(XNP*U(NP)+YNP*V(NP))/ACYL; CCYL=XNP*XNP+YNP*YNP
  -CYRAD2({P1}); ARGCY=BCYL*BCYL-CCYL;
  IF(ARGCY.LT.0.0) [{P3}=0;] ELSE [
  IF(ABS(CCYL).LT.DELACYL.AND.{P2}.EQ.0.AND.BCYL.GE.0.0) [{P3}=0;]
  ELSEIF(ABS(CCYL).LT.DELACYL.AND.{P2}.EQ.1.AND.BCYL.LT.0.0) [
  {P4}=-2.0*BCYL/ACYL;]
  ELSE [ IF(NOFCY.EQ.1.AND.CCYL.GE.0.0) [{P3}=1; {P4}=DELACYL;]
  ELSEIF(NOFCY.EQ.0.AND.CCYL.LE.0.0) [{P3}=1; {P4}=DELACYL;]
  ELSE [ ROOTCY=SQRT(ARGCY); IF(CCYL.LT.0.0)
  [{P4}=(-BCYL+ROOTCY)/ACYL;]
  ELSEIF(BCYL.LT.0.0) [{P4}=(-BCYL-ROOTCY)/ACYL;]
  ELSE [{P3}=0;]]]]}

```

XOP,YOP は、メインで定義し、COMMON/CYLDTA; を介して HOWFAR で使用される。そのため、COMMON/CYLDTA; についても下記のように変更する。

```
"CYLDTA---COMMON BLOCK FOR $CYLNDR MACRO"
REPLACE {;COMIN/CYLDTA/;} WITH {;COMMON/CYLDTA/CYRAD2($MXCYLS),CYRAD($MXCYLS),
XOP($MXCYLS),YOP($MXCYLS);}
```

## 9 標準のジオメトリの一部変更

大部分のリージョンは、標準のジオメトリで良いが一部特別な形状として扱わなければならない場合がある。第7図の形状がその例で、大部分は、円筒平板形状であるが、リージョン3、4で円錐形状を扱う必要がある。このような場合には、リージョン3、4だけを別な扱いにするように変更すれば良い。

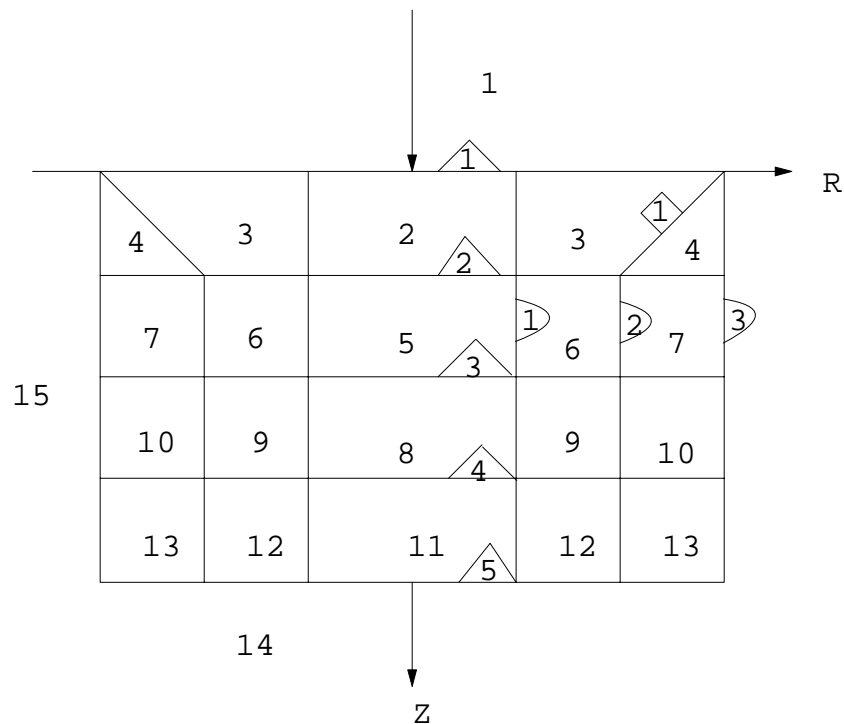


Fig. 7 Cylinder-Slab geometry including a cone

この例のような場合の HOWFAR は、以下のようにすれば良い。

```
IRL=IR(NP); "SET LOCAL VARIABLE"
IF(IRL.LE.1.OR.IRL.GE.IRZ+2) [IDISC=1; RETURN;]

NSLAB=(IRL-2)/NCYL + 1 ; "SLAB NUMBER"
NANNU=IRL-1-NCYL*(NSLAB-1); "ANNULUS NUMBER"
NPL1=NSLAB+1; NPL2=NSLAB;
IF(NSLAB.LT.NPLAN-1) [NRG1=IRL+NCYL;]
ELSE [NRG1=IRZ+2;]
IF(NSLAB.GT.1) [NRG2=IRL-NCYL;]
ELSE [NRG2=1;]

$PLAN2P(NPL1, NRG1, 1, NPL2, NRG2, -1);

IF(IRL.NE.3.AND.IRL.NE.4) [
```

```

IF(NANNU.LT.NCYL) [NRG2=IRL+1;]
ELSE [NRG2=IRZ+3;]
IF(NANNU.GT.1) [NRG1=IRL-1; NCL2=NANNU;
NCL1=NANNU-1;
$CYL2(NCL1,NRG1,NCL2,NRG2); RETURN;]

$CYLNDR(1,1,IHIT,TCYL);
IF(IHIT.EQ.1) [
$CHGTR(TCYL,NRG2);]
]

ELSEIF(IRL.EQ.3) [
$CYLNDR(NANNU-1,0,IHIT,TCYL);
IF(IHIT.EQ.1) [
$CHGTR(TCYL,IRL-1);]
$CONE(1,1,IHIT,TCONE);
IF(IHIT.EQ.1) [
$CHGTR(TCONE,4);]
]

ELSE [
$CYLNDR(NANNU,1,IHIT,TCYL);
IF(IHIT.EQ.1) [
$CHGTR(TCYL,IRZ+3);]
$CONE(1,0,IHIT,TCONE);
IF(IHIT.EQ.1) [
$CHGTR(TCONE,3);]
]

RETURN;

```

平板については、全て同じであるのでそのまま使用し、円筒についての検討部分でリージョン 3、4 を他のリージョンとは異なった検討を行う。具体的には、それぞれのリージョン境界にある円筒及び円錐までの距離を計算し、平板を含めて最も短い距離が USTEP より短い場合には、リージョンが変わるとし、USTEP の値をその距離に置き換えれば良い。

この例のように、一部だけ標準と異なる形状である場合には、その部分だけを別な扱いをする事により処理が可能である。

## 参考文献

- [1] W. R. Nelson and T. M. Jenkins, "Writing SUBROUTINE HOEFAR for EGS4", *SLAC-TN-87-4*, 31 August 1988/Rev.
- [2] W. R. Nelson and T. M. Jenkins, "Geometry Methods and Packages", Chapter 17 in *Monte Carlo Transport of Electron and Photons* Plenum Press, 1988.
- [3] W. R. Nelson, "HOEFAR:How to Code Geometry", EGS4 Lecture Note.
- [4] G. R. Stevenson, "A 3-Dimensional Cartesian Geometry Package for HOWFAR in the EGS Electron Gamma Shower Program", *HS-RP/TM/80-60*.



# 線源ルーチンの作り方

## 1 サンプルング法

### 1.1 Direct method

$f(x)$  が連続な関数で、 $x$  が  $x$  と  $x+dx$  の間にある確率が  $f(x)dx$  であり、 $x$  が  $a \leq x < b$  の範囲の変数で、

$$\int_a^b f(x)dx = 1 \quad (1)$$

である時、 $f(x)$  を確率密度関数と呼ぶ。 $\zeta$  を 0 - 1 の一様乱数とすると、

$$\zeta = F(x) = \int_a^x f(x)dx \quad (2)$$

から、 $x$  を一義的に決める事ができる。 $F(x)$  を累積分布関数と呼ぶ。

$x$  が  $x_i$  の時、確率  $p_i$  であるような離散的な値をとり、

$$F(x_N) = \sum_{i=1}^N p_i = 1 \quad (3)$$

であるならば、一様乱数  $\zeta$  が

$$F(x_{i-1}) = \sum_{j=1}^{i-1} p_j \leq \zeta < F(x_i) = \sum_{j=1}^i p_j \quad (4)$$

の時、 $x$  は  $x_i$  と決める事ができる。

この様に、累積分布関数から直接  $x$  を決定するサンプルングを direct method (直接法) という。

### 1.2 Combination of of “composition” and “rejection” techniques

$f(x)$  が

$$f(x) = \sum_{i=1}^n \alpha_i f_i(x) g_i(x), \quad (\alpha_i : \text{positive real number}) \quad (5)$$

$$\int_a^b f_i(x)dx = 1, \quad (6)$$

$$0 \leq g_i(x) \leq 1.0 \quad (7)$$

の形で表される時、

1.

$$\sum_{j=1}^{i-1} \alpha_j < \zeta_1 \leq \sum_{j=1}^i \alpha_j \quad (8)$$

の条件を満たす  $i$  を選ぶ。

2.  $f_i(x)$  から次の関係式を解くことにより、 $x$  を決める。

$$\int_a^x f_i(x)dx = \zeta_2 \quad (9)$$

3.  $\zeta_3 < g_i(x)$  であれば、 $x$  を採用する。そうでなければ、ステップ 1 に戻ってやり直す。

$\zeta_1, \zeta_2, \zeta_3$  は  $0 - 1$  の一様乱数。

このようにして  $x$  をサンプリングする方法を Combination of the “composition” and “rejection” techniques という。全ての  $i$  について  $g_i = 1$  の時 “composition” 法、 $N=1$  の時が “rejection” 法である。

### 1.3 von Neumann method

Rejection method の一種である von Neumann method と呼ばれる以下の方法で簡単に  $x$  を決める事ができる場合がある。

この方法は、 $f_1(x) = 1/(b - a)$ ,  $g(x) = y = f^*(x) = f(x)/(\text{Maximum value of } f(x))$  であり、以下のようにして  $x$  を決定する。

点  $(\xi, \eta)$  を、 $x=a, x=b, y=0, y=1$  で囲まれた区画内で一様に選び、もし選ばれた点が  $y = f^*(x)$  より上に位置する場合には、サンプリングをやり直し、下に位置する場合には  $\zeta_1$  を  $x$  として採用する。具体的には、

1.  $\xi = a + \zeta_1(b - a)$ ;  $\zeta_1 = \int_a^\xi dx / (b - a) = (\xi - a) / (b - a)$
2.  $y = f^*(\xi)$
3.  $\zeta_2 < y$  ならば、 $x = \xi$
4.  $\zeta_2 > y$  ならば、(1) に戻ってやり直す。

ここで、 $\zeta_1, \zeta_2$  は、共に  $0 - 1$  の一様乱数である。

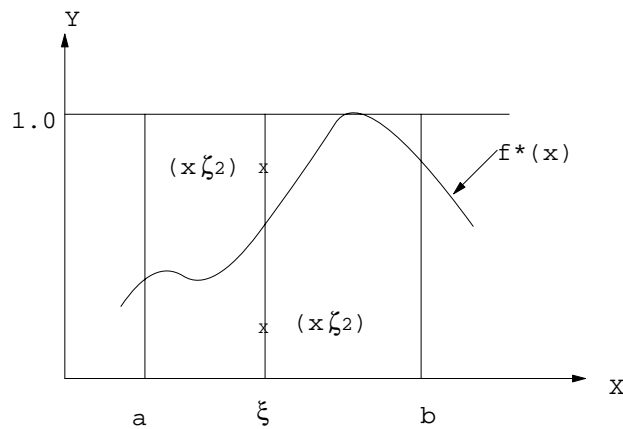


図 1: von Neumann's method.

## 2 線源ルーチン

以下に示す線源ルーチンは、いづれも CALL SHOWER; の前に置き、各ヒストリー毎にサンプリングを行う。

## 2.1 線源位置の決定

### 2.1.1 線線源

線源が  $a \leq x < b$  の区間に一様に分布している様な線源を考える。確率密度関数は、

$$f(x)dx = dx/(b-a); \int_a^b f(\xi)d\xi = 1 \quad (10)$$

である。従って、

$$\zeta = F(x) = \int_a^x f(x)dx = (x-a)/(b-a) \quad (11)$$

となり、 $x = a + \zeta(b-a)$  により、線源の位置を決定する事ができる。

プログラム例

```
$RANDOMSET RNO;
XI=XMIN+RNO*(XMAX-XMIN);
```

### 2.1.2 半径 $R_0$ と $R_1$ の領域に一様に分布した面線源

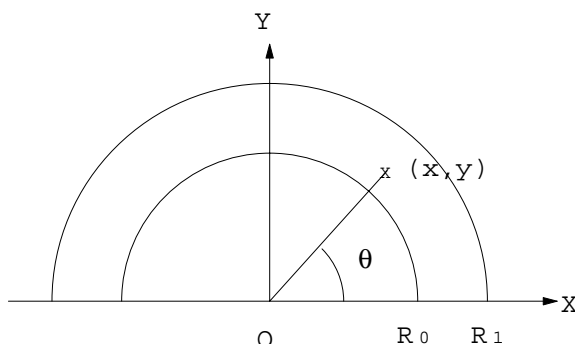


図 2: Uniform source on a annulus of radii  $R_0 < R_1$ .

第 2 図に示すように、原点を中心とする半径  $R_0$  と  $R_1$  の X-Y 平面上に一様に線源が分泌している面線源を考える。半径に関する確率密度関数は、

$$f(r) = 2\pi r dr / \pi(R_1^2 - R_0^2) = 2r dr / (R_1^2 - R_0^2); \int_{R_0}^{R_1} f(r)dr = 1 \quad (12)$$

である。従って、

$$\zeta_1 = F(r) = \int_{R_0}^r f(r)dr = (r^2 - R_0^2)/(R_1^2 - R_0^2) \quad (13)$$

となり、 $r = \sqrt{R_0^2 + \zeta_1(R_1^2 - R_0^2)}$  により、 $r$  を決定する事ができる。

X-Y 平面上での角度  $\phi$  は、 $0 - 2\pi$  に一様であるので、

$$f(\phi) = d\phi/2\pi \quad (14)$$

となり、

$$\zeta_2 = F(\phi) = \int_{-\pi}^{\phi} f(\phi)d\phi = (\phi + \pi)/2\pi \quad (15)$$

から、 $\phi = \pi(2\zeta_2 - 1)$  により決定する事ができる。x, y は

$$x = r \cos \phi \quad (16)$$

$$y = r \sin \phi \quad (17)$$

となる。

#### プログラム例

```
$RANDOMSET RN1;
R02=R0*R0; R12=R1*R1;
RR=SQRT(R02+RN1*(R12-R02));
RANDOMSET RN2;
PHAI=PI*(2.0*RN2-1.0);
" COMIN/UPHIOT/ must be included to use PI"
XI=RR*COS(PHAI);
YI=RR*SIN(PHAI);
```

$R_0=0$  すなわち、半径  $R_1$  の面線源の場合は、 $r = R_1\sqrt{\zeta_1}$  と簡単になる。  
 この場合、rejection 法では以下により x, y の位置を決定する事ができる。 $x=-1.0, x=1.0, y=-1.0, y=1.0$  で囲まれる正方形の区画を考え、この中で点  $(\xi, \eta)$  を一様にサンプリングする。

$$\zeta_1 = \int_{-1.0}^{\xi} d\xi/2.0, \quad \xi = 2\zeta_1 - 1 \quad (18)$$

$$\zeta_2 = \int_{-1.0}^{\eta} d\eta/2.0, \quad \eta = 2\zeta_2 - 1 \quad (19)$$

により決定された点が半径 1.0 の円内にある時、すなわち

$$\sqrt{\xi^2 + \eta^2} \leq 1.0 \quad (20)$$

の場合には、 $x=\xi * R_1, y=\eta * R_1$  を線源位置とし、そうでない場合には、サンプリングをやり直す。

#### プログラム例

```
:SAMPLING:$RANDOMSET RN3;
RANDOMSET RN4;
XIO=2.0*RN3-1;
YIO=2.0*RN4-1;
RR=SQRT(XIO*XIO+YIO*YIO);
IF(RR.GT.1.0) [GO TO :SAMPLING:;
XI=R1*XIO; YI=R1*YIO;
```

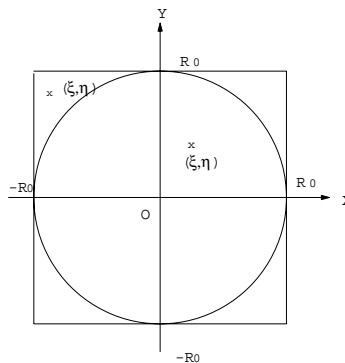


図 3: Determines position by the rejection method.

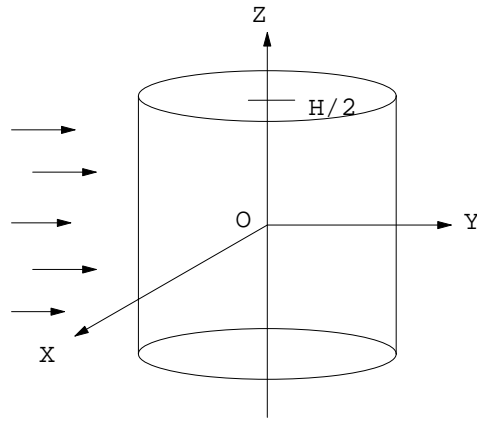


図 4: Parallel-beam source incident on lateral surface on cylinder.

### 2.1.3 直円筒の側面に平行ビームが当たる場合

第 4 図に示すように、Z-軸を中心軸とし、半径  $R_1$  で、 $-H/2$  から  $H/2$  の直円筒の側面に Y-軸の方向に平行なビームがくる場合を考える。粒子が円筒表面に達した位置を線源位置とする。X-方向について考えると、 $-R_1$  から  $R_1$  の区間で一様であるので、

$$x = R_1(2\zeta_1 - 1) \quad (21)$$

により決める事ができる。y は、x が決まると一義的に

$$y = -\sqrt{(R_1^2 - x^2)} \quad (22)$$

となる。z-方向は、x-方向と同様に  $-H/2$  から  $H/2$  間の一様な分布であるので

$$Z = \frac{H}{2}(2\zeta_2 - 1) \quad (23)$$

として決める事ができる。

プログラム例

```

$RANDOMSET RN5;
XI=R1*(2.0*RN5-1.0);
YI=-SQRT(R1*R1-XI*XI);
$RANDOMSET RN6;
ZI=H/2.0*(2.0*RN6-1.0);

```

### 2.1.4 円筒体積線源

第 4 図の円筒内部に一様に分布している線源を考える。x-, y-座標は、面線源で  $R_0=0$  の場合と同じ様に

$$r = R_1\sqrt{\zeta_1} \quad (24)$$

$$\phi = \pi(2\zeta_2 - 1) \quad (25)$$

$$x = r \cos \phi \quad (26)$$

$$y = r \sin \phi \quad (27)$$

により決定する事ができる。また、z-座標は、先の例と同様、

$$Z = \frac{H}{2}(2\zeta_3 - 1) \quad (28)$$

により決定出来る。

### プログラム例

```

$RANDOMSET RN1;
R12=R1*R1;
RR=SQRT(RN1*R12);
$RANDOMSET RN2;
PHAI=PI*(2.0*RN2-1.0);
" COMIN/UPHIOT/ must be included to use PI"
XI=RR*COS(PHAI);
YI=RR*SIN(PHAI);
$RANDOMSET RN3;
ZI=H/2.0*(2.0*RN3-1.0);

```

## 2.2 方向余弦の決定

### 2.2.1 点等方線源

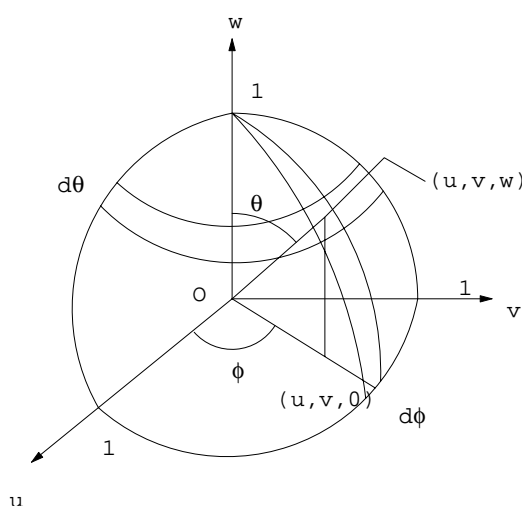


図 5: Determination of directional cosine.

第 5 図に示すように単位球をにおいて z-軸となす角  $\theta$  が  $\theta$  と  $\theta + d\theta$  の間にくる確率を考えると Z-方向の方向余弦  $w$  の確率密度関数は、

$$f(w) = -2\pi \sin \theta d\theta / 4\pi = \frac{1}{2} dw; \int_{-1}^1 f(w) dw = 1 \quad (29)$$

となるので、 $w$  は

$$\zeta_1 = \int_{-1}^w f(w) dw = \int_{-1}^w \frac{1}{2} dw = \frac{1}{2}(w + 1) \quad (30)$$

から、 $w = 2\zeta_1 - 1$  により決定する事が出来る。

$\phi$  については、先の面線源の時と同じ方法で、

$$\phi = \pi(2\zeta_2 - 1) \quad (31)$$

により決定する事ができる。x-, y-方向の方向余弦  $u, v$  は、

$$\sin \theta = \sqrt{1 - w^2} \quad (32)$$

$$u = \sin \theta \cos \phi \quad (33)$$

$$v = \sin \theta \sin \phi \quad (34)$$

## プログラム例

```

$RANDOMSET RN7;
WI=2.0*RN7-1.0;
$RANDOMSET RN8;
PHAI=PI*(2.0*RN8-1.0);
" COMIN/UPHIOT/ must be included to use PI"
SINTH=SQRT(1.0-WI*WI);
UI=COS(PHAI)*SINTH;
VI=SIN(PHAI)*SINTH;

```

rejection 法を用いる事により以下の様にして決定する事もできる。

1.  $x=-1, x=1, y=-1, y=1, z=-1, z=1$  で囲まれた立法体を考える。
2. この立方体内部で一様にサンプリングを行い点  $(x_1, y_1, z_1)$  を決定する。

$$x_1 = 2\zeta_1 - 1 \quad (35)$$

$$y_1 = 2\zeta_2 - 1 \quad (36)$$

$$z_1 = 2\zeta_3 - 1 \quad (37)$$

3. 決定した点が半径 1 の球の外にある時、すなわち

$$R = \sqrt{x_1^2 + y_1^2 + z_1^2} \geq 1 \quad (38)$$

の場合には、ステップ 1 に戻りサンプリングをやり直す。

4. 球内の時は、

$$u = x_1/R; \quad v = y_1/R; \quad w = z_1/R \quad (39)$$

を方向余弦とする。

## プログラム例

```

:SAMPLING:$RANDOMSET RN9;
$RANDOMSET RN10;
$RANDOMSET RN11;
XO=2.0*RN9-1.0;
YO=2.0*RN10-1.0;
ZO=2.0*RN11-1.0;
RO=SQRT(XO*XO+YO*YO+ZO*ZO);
IF(RO.GT.1.0) [GO TO :SAMPLING:;]
UI=XO/RO; UI=YO/RO; WI=ZO/RO;

```

検出器前面から離れた位置に点等方線源がある場合を考える。形状は、ucnai3.mor, ucnai3p.mor の様な円筒/平板形状とし、ガンマ線で途中の空気の影響は無視できるとする。線源から、検出器を見込む立体角外に放出されるガンマ線は検出器には入らないので、ウエイトで補正する事とする。検出器の前面を  $Z=0$  とし、 $-Z_0$  の位置に線源があるとする。 $\theta$  が、 $\theta_0$  より、大きい時は検出器には入射しないので、線源で発生するガンマ線の内、検出器に達する割合は、

$$WT = \int_{\theta_0}^0 (-2\pi \sin \theta d\theta) / 4\pi = (1 - \cos \theta_0) / 2 \quad (40)$$

である。この値をウエイトとして用いる。 $\theta$  が、0 から  $\theta_0$  の間にある場合には、Z-方向の方向余弦の確率密度関数は、

$$f(w)dw = -2\pi \sin \theta d\theta / (2\pi(1 - \cos \theta_0)) = dw / (1 - \cos \theta_0) \quad (41)$$

となるので、 $w$  は、

$$\zeta_1 = \int_{\cos \theta_0}^w f(w)dw = \int_{\cos \theta_0}^w \frac{1}{1 - \cos \theta_0} dw = \frac{w - \cos \theta_0}{1 - \cos \theta_0} \quad (42)$$

から、 $w = 2\zeta_1(1 - \cos \theta_0) + \cos \theta_0$  により決定する事が出来る。  
検出器前面での入射粒子の座標は、

$$WI = w \quad (43)$$

$$UI = \cos \phi \sqrt{1 - w^2} \quad (44)$$

$$VI = \sin \phi \sqrt{1 - w^2} \quad (45)$$

$$XI = Z0 \times UI \quad (46)$$

$$YI = Z0 \times VI \quad (47)$$

$$ZI = 0.0 \quad (48)$$

$\phi$  は、乱数を使って求める事になるが、形状が Z-軸に対して対称である場合には上記に様に  $UI, VI$  を決める必要はなく、 $UI = \sqrt{1 - w^2}, VI = 0.0$  として差し支えない。

粒子の入射するリージョンは、入射点の半径から判断する。先に求めたウエイトは、WTI に設定する。検出器中での吸収エネルギーを求める場合には、AUSGAB ではウエイトを加味せず CALL SHOWER の後で検出効率や波高分布を求める時に加味し、スペクトルの場合は、AUSGAB でのスコア時に加味する。



## プログラム例

```

"*****"
"***** STEP 6. DETERMINATION OF INCIDENT PARTICLE PROPERTIES *****"
"*****"

ZI=0.0; "STARTING COORDINATES (CM)"
ZO=-10.0;
TT=SQRT(ZO*ZO+CYRAD2(3));
COSMI=ZO/TT; "Minimum value of WI to enter detector"
WTI=(1-COSMI)/2.0;
.....

"*****"
"***** STEP 7. SHOWER-CALL---NEXT *****"
"*****"

DO NOFBAT=1,NBATCH [ "BATCH-LOOP"

DO I=1,NCASPB ["START OF SHOWER CALL LOOP OF EACH BATCH"

$RANDOMSET WIO;
WI=WIO*(1.0-COSMI)+COSMI;
$RANDOMSET PHAIO;
PHAI=PI*(2.0*PHAIO-1.0);
" COMIN/UPHIOT/ must be included to use PI"
SINTH=SQRT(1.0-WI*WI);
UI=COS(PHAI)*SINTH;
VI=SIN(PHAI)*SINTH;
XI=ABS(ZO)*UI;
YI=ABS(ZO)*UI;
RR2=XI*XI+YI*YI;
IF(RR2.LE.CYRAD2(1)) [IRI=2;]
ELSEIF(RR2.LE.CYRAD2(2)) [IRI=3;]
ELSE [IRI=4;]

IF(NCOUNT.LE.NWRITE.AND.ILINES.LE.NLINES) [
  OUTPUT EI,XI,YI,ZI,UI,VI,WI,
  IQI,IRI,IDINC; (7G15.7,3I5);
  ILINES=ILINES+1;]

CALL SHOWER(IQI,EI,XI,YI,ZI,UI,VI,WI,IRI,WTI);

```

### 2.2.2 コサイン分布

粒子の角度分布が  $\cos\theta$  に比例するコサイン分布の場合には次の様にする。方向余弦  $w$  が正の場合のみが意味があるので、分布関数は、

$$f(w) = kw dw; \int_0^1 kw dw = \frac{1}{2}k \quad (49)$$

$k$  は、比例常数で、 $\int_0^1 f(w)dw = 1$  なので、 $k=2$  となる。従って、 $w$  は、

$$\zeta_1 = \int_0^w 2w dw = w^2 \quad (50)$$

より、 $w = \sqrt{\zeta_1}$  となる。 $\phi$  については、点等方線源の場合と同じである。

## プログラム例

```

$RANDOMSET RN12;
WI=SQRT(RN12);
$RANDOMSET RN13;
PHAI=PI*(2.0*RN13-1.0);
" COMIN/UPHIOT/ must be included to use PI"
SINTH=SQRT(1.0-WI*WI);
UI=COS(PHAI)*SINTH;
VI=SIN(PHAI)*SINTH;

```

## 2.3 エネルギーの決定

### 2.3.1 内挿法による決定

線源のエネルギーの決定は、線源のエネルギー分布から決定するが、一般的には直接法により求める事は難しい。最も一般的なのは、対象となるエネルギー領域をエネルギーグループに分け、理論または実測データから求めたエネルギーグループの確率密度関数を入力データとして用いる方法である。

上限エネルギーを  $E_n$ 、各エネルギーグループの下限エネルギーを  $E_1 < E_2 < \dots < E_n$  とし、対応する確率密度関数を  $F(E_1), F(E_2), \dots, F(E_n)$  とした時具体的な方法は、

1.  $F(E_i) < \zeta < F(E_{i+1})$  となる  $i$  を求める。
2.  $E_i$  と  $E_{i+1}$  の区間で以下の直線内挿によりエネルギーを決定する。

$$E = E_i + \frac{(\zeta - F(E_i)) \times (E_{i+1} - E_i)}{F(E_{i+1}) - F(E_i)} \quad (51)$$

である。

#### プログラム例

```

$RANDOMSET RN14;
DO IE=2,NEMAX [
IF(RN14.LE.PDF(IE)) EXIT;
]
EI=ES(IE-1)+(RN14-PDF(IE-1))*(ES(IE)-ES(IE-1))/(PDF(IE)-PDF(IE-1));

```

### 2.3.2 Rejection 法による決定

エネルギースペクトルが式の形で与えられている場合には rejection 法を用いる事ができる。エネルギースペクトルが  $p(E)$  で、対象とするエネルギー領域  $E_L$  から  $E_U$  間でのエネルギースペクトルの最大値を  $p^*$  とすると、

1.  $E = E_L + \zeta_1(E_U - E_L)$
2.  $\zeta_2 < p(E)/p^*$  の場合は、 $E$  を決定すべきエネルギーとし、そうでない場合はステップ 1 に戻ってやりなおす。

方法で決定する事ができる。

#### プログラム例

```

:SAMPLING:$RANDOMSET RN15;
EI=EMIN+RN15*(EMAX-EMIN);
PDF=SPEC(EI)/SPEMAX;
"SPEC is the FUNCTION to calculate spectrum"
$RANDOMSET RN16;
IF(RN16.GT.PDF) [GO TO :SAMPLING:;]

```

### 2.3.3 ウェイトを用いる方法

先の方法はどれも確率分布関数の形を反映しているので、頻度の少ないエネルギー領域は当然の事ながらサンプリングされる割合が小さい。しかし、問題によっては、頻度の少ない領域の粒子が結果に大きく寄与する場合がある。このような場合にはウェイトを用いると効率的な計算ができる場合がある。

具体的には

1. 対象となるエネルギー領域でエネルギーを一様にサンプリングする。

$$E = E_L + \zeta(E_U - E_L) \quad (52)$$

2. 線源粒子のウェイトを  $f(E)$  とする。

プログラム例

```
$RANDOMSET RN17;  
EI=EMIN+RN17*(EMAX-EMIN);  
WTI=PDF(EI);  
"PDF is the probability distribution function of spectrum"
```

# Appendix: Lists of ucna3p.mor

```

!INDENT M3;
!INDENT F2;
"*****"
"***** STANFORD LINEAR ACCELERATOR CENTER *"
"***   U C N A I 3 P   *****"
"***** EGS4 USER CODE -- 27 JUL 1998/0930 *"
"*****"
"
" PROGRAMMER:   HIDEO HIRAYAMA
"              NATIONAL LABORATORY FOR HIGH ENERGY PHYSICS (KEK)
"              OHO-MACHI, TSUKUBA-GUN, IBARAKI,
"              JAPAN
"
"*****"
"
" PROGRAM:      UCNAI3
"
"              EGS4 USER CODE TO CALCULATE NAI RESPONSE
"              Cylinder-slab geometry with PRESTA.
"              Add Ranmar random generator option.
"
"*****"
"
"              F E A T U R E S
"
" - USES ENERGY CONSERVATION PROGRAM CALLED ECNSV1
" - USES 'COUNTER' ROUTINE CALLED NTALLY
"
"*****"
"
" THE FOLLOWING 'STEPS' REFER TO THE STEPS OUTLINED
" IN THE EGS3 USER MANUAL (SLAC-210).
" VARIOUS EGS USER NOTES (EUN'S) HAVE BEEN CREATED
" TO SUPPLEMENT SLAC-210 FOR THE CORRECTIONS, CHANGES
" AND ADDITIONS THAT ARE IN EGS4.
"
"*****"
"***** STEP 1.  USER-OVER-RIDE-OF-EGS-MACROS *****"
"*****"
%C80
!NEWCONDITIONAL;
"-----"
"Select random number generater: 0=RAN6 1=RANMAR
"RANMAR is a Lagged-Fibonacci Method pseudo random number generator"
"devised by George Marsaglia and Arif Zaman.
"-----"
REPLACE {$RNGEN} WITH {0}

"STEP 1.  USER-OVER-RIDE-OF-EGS-MACROS"

REPLACE {;COMIN/RANDOM/;} WITH {
  {SETR B=$RNGEN}
  [IF] {COPY B}=0 [
    ;COMMON/RANDOM/URNDM(97),IXX,IXXST;
  ]
  [IF] {COPY B}=1 [
    ;COMMON/RANDOM/URNDM(97),CRNDM,CDRNDM,CMRNDM,IXX,JXX;
  ]
}

REPLACE {$COMMON-RANDOM-DECLARATION-IN-BLOCK-DATA;} WITH {
  {SETR B=$RNGEN}
  [IF] {COPY B}=0 [
    ;COMMON/RANDOM/URNDM(97),IXX,IXXST;
  ]
  [IF] {COPY B}=1 [
    ;COMMON/RANDOM/URNDM(97),CRNDM,CDRNDM,CMRNDM,IXX,JXX;
  ]
}

```

```

}

REPLACE {$RANDOMSET#;} WITH {
  {SETR B=$RNGEN}
  [IF] {COPY B}=0 [
    IXX=IXX*663608941;{P1}=IXX*0.23283064E-09;IF(IXX.LT.0){P1}={P1}+1.0;
    IF(IXX.EQ.IXXST) [OUTPUT;(' WARNING !'/
    ' Same random number will be produced.'/
    ' It is better to use RANMAR random number generator.')]
  ]
  [IF] {COPY B}=1 [
    {P1}=URNDM(IXX)-URNDM(JXX); IF({P1}.LT.0.) {P1}={P1}+1.;
    URNDM(IXX) = {P1};
    IXX=IXX-1; IF(IXX.EQ.0) IXX=97;
    JXX=JXX-1; IF(JXX.EQ.0) JXX=97;
    CRNDM=CRNDM-CDRNDM; IF(CRNDM.LT.0.) CRNDM=CRNDM+CMRNDM;
    {P1}={P1}-CRNDM; IF({P1}.LT.0.) {P1}={P1}+1.;
  ]
}

```

"This should be called somewhere near the beginning of the main routine"  
"before any random numbers are asked for";

```

REPLACE {$RNG-INITIALIZATION;} WITH {;
  {SETR B=$RNGEN}
  [IF] {COPY B}=0 [;]
  [IF] {COPY B}=1 [ IXX=0; JXX=0; CALL RMARIN;
    DO II=1,20005[
      $RANDOMSET XRANM;
      IF(II.GT.20000) OUTPUT (MOD(INT(XRANM*16.**JJ),16),JJ=1,7);
      (8X,7I3); ]
  ]
}

```

```

"-----"
"          PLACE COMPILER DEPENDENT SUBROUTINE CALL HERE          "
"-----"
" Select Fortran Compiler.                                         "
" Lahey Fortran = 1                                               "
" Microsoft Fortran = 2                                           "
" Other PC = 3 (default)                                         "
" Sun UNIX Workstation = 4                                        "
" HI-UX Workstation =5                                           "
" Other UNIX Workstation =6                                       "
"-----"

```

```

REPLACE {$COMPILER} WITH {3}
"-----"

```

```

"-----"
"Macro to select the timer to be used (compiler dependent).      "
"-----"
REPLACE {$TIMERSET;} WITH {
  {SETR B=$COMPILER}
  [IF] {COPY B}=5 [CALL CLOCK(IDUMMY,3);]
  [ELSE]           [;]
}

```

```

REPLACE {$TIME-NOW#;} WITH {
  {SETR B=$COMPILER}
  [IF] {COPY B}=1 [CALL TIMER({P1});]
  [IF] {COPY B}=2 [CALL GETTIM(IHR,IMIN,ISEC,I100);
    {P1}=(IHR*3600+IMIN*60+ISEC)*100+I100;]
  [IF] {COPY B}=3 [{P1}=0.0;]
  [IF] {COPY B}=4 [TT=ETIME(TARRAY);
    {P1}=TARRAY(1)*100.0;]
  [IF] {COPY B}=5 [CALL CLOCK(TT,5);
    {P1}=TT*100.0;]
  [IF] {COPY B}=6 [{P1}=0.0;]
}

```

```

"-----"
"Macro to define DIMENSION related to the timer to be used      "
"          (compiler dependent).                                  "
"-----"

```

```

REPLACE {$TIME-DIM;} WITH {
  {SETR B=$COMPILER}
  [IF] {COPY B}=4 [REAL TARRAY(2);]
  [ELSE]          [;]
}

"-----"
"Macro to define OPEN STATEMENT for Cross-section data.      "
"                                                                "
"                                                                "
"-----"
REPLACE {$OPEN;} WITH {
  {SETR B=$COMPILER}
  [IF] {COPY B}<4 [OPEN(12,FILE='mortjob.xse',status='old');
                  OPEN(6,FILE='mortjob.out',status='new');
                  OPEN(8,FILE='mortjob.dum',status='new');]
  [ELSE]          [OPEN(12,FILE='mortjob.xsec',status='old');
                  OPEN(6,FILE='mortjob.output',status='new');
                  OPEN(8,FILE='mortjob.dummy',status='new');]
}

REPLACE{$$CALL-HOWNEAR(#);} WITH
  {$CALL-HOWNEAR-FOR-SLAC-CYLINDRICAL-GEOMETRY({P1});}
"          ****          "
"THIS IS THE MACRO THAT SHOULD RETURN THE CLOSEST PERPENDICULAR "
"DISTANCE TO ANY SURFACE WHICH FORMS A BOUNDARY FOR THE CURRENT "
"REGION. IN THIS APPLICATION IT IS REPLACED BY THE MACRO FOLLOWING "
"WHICH IS SPECIALIZED FOR THE SLAC          "          ****          "
"          "          "          "          "          "          "
"CYLINDRICAL-PLANE GEOMETRY.          "
"IT IS THE USER'S RESPONSIBILITY TO PROVIDE THIS MACRO FOR HIS OWN "
"GEOMETRY.          "
"+++++          "
; "BUFFER FLUSH"
REPLACE{$$CALL-HOWNEAR-FOR-SLAC-CYLINDRICAL-GEOMETRY(#);} WITH
"          *****          "
  {;ZL=Z(NP);RL=SQRT(X(NP)**2+Y(NP)**2);
  $GET-JR-JZ(IRL);
  ZLEFT=ZL-PCOORD(3,JZ);ZRIGHT=PCOORD(3,JZ+1)-ZL;ROUT=CYRAD(JR)-RL;
  {P1}=MIN(ZLEFT,ZRIGHT,ROUT);
  IF(JR.NE.1) [RIN=RL-CYRAD(JR-1);{P1}=MIN({P1},RIN);]}
"THIS ROUTINE IS INTENDED TO BE USED TO CALCULATE THE MINIMUM "
"PERPENDICLAR TO THE NEAREST BOUNDING SURFACE. THIS VERSION IS "
"SPECIALLY DESIGNED FOR THE SLAC CYLINDRICAL GEOMETRY PACKAGE. "
"A DIFFERENT VERSION IS NEEDED FOR OTHER          "
"****          "
"GEOMETRY PACKAGES.          "
; "BUFFER FLUSH"
"SLAC MACRO TO GET CYLINDER (JR) AND SLAB (JZ) ZONES FROM REGION "
"NUMBER"          "
"*****          "
REPLACE {$GET-JR-JZ(#);} WITH
"          =====          "
  {;JZ=({P1}-2)/NCYLP+1;JR={P1}-1-NCYLP*(JZ-1);}
; "BUFFER FLUSH"
"GEOMETRICAL INFORMATION"
"SLAC DEFINITION OF /GEOM/ AND RE-DEFINITIONS FOR /CYLDTA/ AND "
"/PLADTA/          "
"****          "
REPLACE {;COMIN/GEOM/;} WITH {;COMIN/CYLDTA,PLADTA/;}
"          =====          "

REPLACE {;COMIN/CYLDTA/;} WITH
"          =====          "
  {;COMMON/CYLDTA/CYRAD($MXCYLS),CYRAD2($MXCYLS),NCYLP;}
"NOTE: CYRAD MUST BE PROVIDED IN ADDITION TO CYRAD2 (USUALLY IN "
"MAIN)"          "

REPLACE {;COMIN/PLADTA/;} WITH
"          =====          "
  {;COMMON/PLADTA/PCOORD(3,$MXPLNS),PNORM(3,$MXPLNS);}

```

```

"COMMON to define variables to score at AUSGAB"
"DEPE:deposited energy inside the detector"
"DELTAE:energy bin width in MeV"
"SPG:Gamma spectrum, SPE:Electron spectrum, SPP:Positron spectrum"

REPLACE {;COMIN/TOTALS/;} WITH
{;COMMON/TOTALS/DEPE,DELTAE,SPG($NDET,$NEBIN),SPE($NDET,$NEBIN),
SPP($NDET,$NEBIN);}

"COMMON of print-out parameter"

REPLACE {;COMIN/LINES/;} WITH
{;COMMON/LINES/NLINES,NWRITE,NCOUNT,ILINES;}

"COMMON of geometry related parameter"

REPLACE {;COMIN/PASSIT/;} WITH
{;COMMON/PASSIT/NREG,NPLAN,NCYL,IRZ;}

PARAMETER $MXPLNS=5; "NUMBER OF PLANE"
PARAMETER $MXCYLS=3; "NUMBER OF CYLINDER"
PARAMETER $NCASES=5000; "MAXIMUM NUMBER OF CASES"
PARAMETER $NBATCH=50; "Number of batch"
PARAMETER $NEBIN=50; "Number of energy bin"
PARAMETER $NDET=1; "Number of detector"

"*****"
"***** ADDITIONAL (NON-EGS) MACROS *****"
"*****"

" N O N E "

"*****"
"***** DECLARATIONS *****"
"*****"

;COMIN/DEBUG,BOUNDS,BREMPR,EDGE,ELEGIN,ETALY1,GEOM,LINES,MEDIA,MISC,
NTALY1,PASSIT,RANDOM,STACK,THRESH,TOTALS,UPHIOT,USEFUL,USER/;
DIMENSION PH($NEBIN),PHPB($NEBIN,$NBATCH);
DIMENSION SPGPB($NDET,$NEBIN,$NBATCH),SPEPB($NDET,$NEBIN,$NBATCH),
SPPPB($NDET,$NEBIN,$NBATCH);
DIMENSION PEPFB($NBATCH),TEFPB($NBATCH);
REAL*8 TOTKE,AVAILE,DEPE;
"NEEDED FOR ENERGY CONSERVATION TABULATION"

$TYPE MEDARR(24,3);
DATA MEDARR/$S'NAI-IAPRIM',14*' ',
$S'AL-IAPRIM',15*' ',
$S'QUARTZ-IAPRIM',11*' ' /;

"*****"
"***** START OF EXECUTABLE CODE *****"
"*****"

$OPEN;

"*****"
"***** STEP 2. PRE-HATCH-CALL-INITIALIZATION COMES NEXT *****"
"*****"
NMED=3; "NUMBER OF MEDIA"

DO J=1,NMED [
DO I=1,24 [MEDIA(I,J)=MEDARR(I,J);]]

NPLAN=$MXPLNS; "NUMBER OF PLANES"
NCYL=$MXCYLS; "NUMBER OF CYLINDER"
NCYLP=NCYL; "*PRESTA*"

NREG=(NPLAN-1)*NCYL+3; "NUMBER OF REGIONS (INCLUDING OUTSIDE VACUUM"
" REGION) "
IRZ=NREG-3;

```

```

"SET MEDIUM INDEX FOR EACH REGION"

/MED(1),MED(NREG-1),MED(NREG)/=0;  "VACUUM REGIONS"

/MED(2),MED(3),MED(4),MED(7),MED(10),MED(13)/=2;
"
                                Al region"
MED(8)=1;  "NaI(Tl) detector region"
/MED(11),MED(12)/=3;  "Quartz region"

/ECUT(2),ECUT(3),ECUT(4),ECUT(7),ECUT(8),ECUT(10)/=0.561;
/ECUT(11),ECUT(12)/=0.561;

/MED(5),MED(6),MED(9)/=0;  "Vacuum region inside case"

IEDGFL(8)=53;  "53:Atomic number of I"
"
                                0:K-X ray of I is not produced"

"*****"
"***** STEP 3.  HATCH-CALL COMES NEXT *****"
"*****"

CALL HATCH;

"OUTPUT VARIOUS QUANTITIES ASSOCIATED WITH THE MEDIA"

OUTPUT;  ('1QUANTITIES ASSOCIATED WITH EACH MEDIA:',//);

DO J=1,NMED [
OUTPUT (MEDIA(I,J),I=1,24);  (/ ,1X,24A1);
OUTPUT RHO(J),RLC(J);  (5X,' RHO=',G15.7,' G/CM**3      RLC=',
G15.7,' CM');
OUTPUT AE(J),UE(J);  (5X,' AE=',G15.7,' MEV      UE=',G15.7,' MEV');
OUTPUT AP(J),UP(J);  (5X,' AP=',G15.7,' MEV      UP=',G15.7,' MEV');
]

OUTPUT;(/' INFORMATION OF MEDIUM AND CUT-OFFFOR EACH REGION'//);
DO I=1,NREG [
IF(MED(I).EQ.0) [OUTPUT I,ECUT(I),PCUT(I);
(' MEDIUM(',I3,')=VACUUM',18X,'ECUT=',G10.5,' MEV, PCUT=',G10.5,' MEV');
]
ELSE [OUTPUT I,(MEDIA(II,MED(I)),II=1,24),ECUT(I),PCUT(I);
(' MEDIUM(',I3,')=',24A1,'ECUT=',G10.5,' MEV, PCUT=',G10.5,' MEV');]
]

DO I=1,NREG [IF(IEDGFL(I).NE.0) [CALL EDGSET(NREG); EXIT;]]

"*****"
"***** STEP 4.  HOWFAR-INITIALIZATION COMES NEXT *****"
"*****"

"DEFINE VARIOUS THICKNESSES/DISTANCES"

TCOV=0.1;  "Thickness of Al case in cm"
TGAP=0.5;  "Gap between case and detector in cm"
TDE=7.62;  "Thickness of detector in cm"
TQUARTZ=0.5;"Thickness of quartz window in cm"

"DEFINITION OF PLANES"

"SET ALL COORDINATES AND NORMALS TO ZERO TO BEGIN WITH"
DO J=1,NPLAN [
PCOORD(1,J)=0.0;  PCOORD(2,J)=0.0;  PCOORD(3,J)=0.0;
PNORM(1,J)=0.0;  PNORM(2,J)=0.0;  PNORM(3,J)=1.0;
]

"NOW PUT IN THE EXCEPTIONS"

PCOORD(3,2)=PCOORD(3,1)+TCOV;
PCOORD(3,3)=PCOORD(3,2)+TGAP;
PCOORD(3,4)=PCOORD(3,3)+TDE;
PCOORD(3,5)=PCOORD(3,4)+TQUARTZ;

OUTPUT;  ('1PCOORD AND PNORM VALUES FOR EACH J-PLANE (I=1,3):',//);

```



```

DO J=1,NPLAN [
OUTPUT J,(PCOORD(I,J),I=1,3),(PNORM(I,J),I=1,3);
(I5,6G15.7);]

"DEFINE THE CYLINDER RADII"

RDET=3.81; "Radius of detector in cm"
RGAP=0.5; "Gap between detector and case in cm"
RTCOV=0.1; "Cover thickness in cm"

CYRAD(1)=RDET;
CYRAD(2)=CYRAD(1)+RGAP;
CYRAD(3)=CYRAD(2)+RTCOV;

"PRINT OUT THE CYLINDER RADII"
OUTPUT ;(///,' CYLINDER RADII',/);
DO I=1,NCYL [
CYRAD2(I)=CYRAD(I)*CYRAD(I);
OUTPUT I,CYRAD(I);(' I=',I5,5X,' CYRAD(I)=',G15.5,'CM');
]
"*****"
"***** STEP 5.  INITIALIZATION FOR AUSGAB COMES NEXT *****"
"*****"

CALL ECNSV1(0,NREG,TOTKE);"  INITIALIZE ESUM ARRAY FOR ENERGY"
"          CONSERVATION CALCULATION."
"          NREG=NUMBER OF REGIONS"
"          TOTKE=TOTAL KE (DUMMY VARIABLE HERE)"
"          (MUST BE REAL*8)"

CALL NTALLY(0,NREG);

NCOUNT=0; "PARTICLE HISTORY COUNTER"
ILINES=0; "INITIALIZE LINE-OUTPUT COUNTER"
DEPE=0.DO; "ZERO THE ENERGY DEPOSITION AT SCINTILATOR"
/PEF,TEF/=0.0; "Zero the efficiency"

DO J=1,$NEBIN [PH(J)=0.0;] "Zero the pulse-heght"
DO ND=1,$NDET [
DO J=1,$NEBIN [
/SPG(ND,J),SPE(ND,J),SPP(ND,J)/=0.0; "Zero the spectrum"
]]

"*****"
"***** STEP 6.  DETERMINATION OF INCIDENT PARTICLE PROPERTIES *****"
"*****"

IQI=0; "INCIDENT PARTICLE"

EI=1.33 +ABS(IQI)*PRM; "TOTAL ENERGY OF PARTICLE (MEV) "

AVAILE=EI + IQI*PRM; "AVAILABLE K.E. (MEV) (MUST BE REAL*8)"
EKIN=AVAILE;
ECUTMN=ECUT(4); EKO=EKIN; "*PRESTA*"
$PRESTA-INPUTS; "INPUT THE *PRESTA* VARIABLES"

DELTAE=0.05; "Energy bin of response"

XI=0.0; YI=0.0; ZI=0.0; "STARTING COORDINATES (CM)"
UI=0.0; VI=0.0; WI=1.0; "INCIDENT DIRECTION COSINES"
IRI=2; "ENTRANCE REGION DEFINITION"
WTI=1.0; "WEIGHT FACTOR OF UNITY"

IDINC=-1; "AN IDENTIFIER (LIKE IARG) TO MARK INCIDENT PARTICLES"

IXXST=17847465;
IXX=IXXST; "INITIALIZED RANDOM NUMBER WITH STARTING SEED"

$RNG-INITIALIZATION;

NWRITE=10; "NUMBER OF INCIDENT CASES TO PRINT OUT"

NCASES=$NCASES; "MAXIMUM NUMBER OF INCIDENT CASES TO RUN"

```

```

NBATCH=$NBATCH; "NUMBER OF BATCH"
NCASPB=NCASES/NBATCH; "NUMBER OF CASES PER BATCH"
NOFBAT=0; "NUMBER OF BATCH FINISHED"

NLINES=15; "NUMBER OF LINES TO PRINT OUT"

"*****"
"***** STEP 7. SHOWER-CALL---NEXT *****"
"*****"

DO NOFBAT=1,NBATCH [ "BATCH-LOOP"

DO I=1,NCASPB ["START OF SHOWER CALL LOOP OF EACH BATCH"

IF(NCOUNT.LE.NWRITE.AND.ILINES.LE.NLINES) [
  OUTPUT EI,XI,YI,ZI,UI,VI,WI,
  IQI,IRI,IDINC; (7G15.7,3I5);
  ILINES=ILINES+1;]

CALL SHOWER(IQI,EI,XI,YI,ZI,UI,VI,WI,IRI,WTI);

"If some energy is deposited inside detector add pulse-height"
"and efficiency"

IF(DEPE.GT.0.DO) [
  IE=DEPE/DELTAE+1;
  IF(IE.LE.$NEBIN) [PH(IE)=PH(IE)+WTI;]
  IF(DEPE.GE.EI*0.999) [PEF=PEF+WTI;]
  TEF=TEF+WTI;]
DEPE=0.DO;

NCOUNT=NCOUNT + 1;
IXXEND=IXX; "LAST RANDOM NUMBER USED"

] "End of SHOWER CALL loop for each BATCH"

"Calcurate average value for this BATCH"

DO IE=1,$NEBIN [
  PHPB(IE,NOFBAT)=PH(IE)/NCASPB;
  PH(IE)=0.0;
]
PEFPB(NOFBAT)=PEF/NCASPB;
TEFPB(NOFBAT)=TEF/NCASPB;
/PEF,TEF/=0.0;

DO ND=1,$NDET [
DO IE=1,$NEBIN [
  SPGPB(ND,IE,NOFBAT)=SPG(ND,IE)/NCASPB; "Gamma spectrum for this BATCH"
  SPEPB(ND,IE,NOFBAT)=SPE(ND,IE)/NCASPB; "Electron spectrum for this BATCH"
  SPPPB(ND,IE,NOFBAT)=SPP(ND,IE)/NCASPB; "Positron spectrum for this BATCH"
  /SPG(ND,IE),SPE(ND,IE),SPP(ND,IE)/=0.0;
]]

] "End of BATCH-loop"

TOTKE=NCOUNT*AVAILE; "TOTAL (AVAILABLE) K.E."

"*****"
"***** STEP 8. OUTPUT OF RESULTS *****"
"*****"

OUTPUT NCOUNT,IXXST,IXXEND,AVAILE,TOTKE;
('1',I10,' CASES COMPLETED',
//,' IXXST=',I12,/, ' IXXEND=',I12,/, ' AVAILABLE K.E.=',
G15.5,' MEV',/, ' TOTKE=',E15.5,' MEV',/);

IF(IPLC.EQ.0.AND.IBCA.EQ.0) [
OUTPUT;(/' PRESTA algorithm is used'/);]
ELSE [
OUTPUT;(/' Default EGS4 calculation'/);]

```

```

OUTPUT TDE,RDET,TCOV,RTCOV,TGAP,RGAP;
(// Detector length=',G15.5,' cm'/ ' Detector radius=',G15.5,' cm'/
' Al cover thickness=',G10.2,' cm'/ ' Al cover side thickness=',
G10.2,' cm'/ Front gap =',G10.2,' cm'/ Side gap =',G10.2,' cm'/);

OUTPUT EI;(' Results for ',G15.5,'MeV photon'/);

"Calculate average and its deviation"

/AVPE,DESCI2/=0.0;
DO J=1,NBATCH [
AVPE=AVPE+PEFPB(J)/NOFBAT;
DESCI2=DESCI2+PEFPB(J)*PEFPB(J)/NBATCH;
]
SIGPE=SQRT((DESCI2-AVPE*AVPE)/(NBATCH-1));
AVPE=AVPE*100.0;
SIGPE=SIGPE*100.0;
OUTPUT AVPE,SIGPE;(' Peak efficiency =',G15.5,'+-',G15.5,' %');

/AVTE,DESCI2/=0.0;
DO J=1,NBATCH [
AVTE=AVTE+TEFPB(J)/NOFBAT;
DESCI2=DESCI2+TEFPB(J)*TEFPB(J)/NBATCH;
]
SIGTE=SQRT((DESCI2-AVTE*AVTE)/(NBATCH-1));
AVTE=AVTE*100.0;
SIGTE=SIGTE*100.0;
OUTPUT AVTE,SIGTE;(' Total efficiency =',G15.5,'+-',G15.5,' %');

OUTPUT ;(// Pulse height distribution ');
DO IE=1,$NEBIN [
ELOW=DELTAE*(IE-1);
EUP=DELTAE*IE;
IF(ELOW.GT.EKIN) [EXIT;]

/AVPH,DESCI2/=0.0;
DO J=1,NBATCH [
AVPH=AVPH+PHPB(IE,J)/NBATCH;
DESCI2=DESCI2+PHPB(IE,J)*PHPB(IE,J)/NBATCH;
]
SIGPH=SQRT((DESCI2-AVPH*AVPH)/(NBATCH-1));
AVPH=AVPH/DELTAE;
SIGPH=SIGPH/DELTAE;
OUTPUT EUP,AVPH,SIGPH;
(' E (upper-edge --',G10.4,' MeV )=',G15.5,'+-',G15.5,
' counts/MeV/incident');
]
"Particle spectrum. Incident particle spectrum to detector system"
"in this example."

OUTPUT;(// Particle spectrum crossing the detector plane'/
30X,'particles/MeV/source photon'/
' Upper energy',11X,' Gamma',18X,' Electron',14X,' Positron');

DO ND=1,$NDET [ "Spectrum at each detector plane"
DO IE=1,$NEBIN [
ELOW=DELTAE*(IE-1);
EUP=DELTAE*IE;
IF(ELOW.GT.EKIN) [EXIT;]

"Gamma spectrum per MeV per source"
/AVSPG,DESCI2/=0.0;
DO J=1,NBATCH [
AVSPG=AVSPG+SPGPB(ND,IE,J)/NBATCH;
DESCI2=DESCI2+SPGPB(ND,IE,J)*SPGPB(ND,IE,J)/NBATCH;
]
SIGSPH=SQRT((DESCI2-AVSPG*AVSPG)/(NBATCH-1));
AVSPG=AVSPG/DELTAE;
SIGSPH=SIGSPH/DELTAE;

"Electron spectrum per MeV per source"
/AVSPE,DESCI2/=0.0;

```

```

DO J=1,NBATCH [
AVSPE=AVSPE+SPEPB (ND, IE, J)/NBATCH;
DESCI2=DESCI2+SPEPB (ND, IE, J)*SPEPB (ND, IE, J)/NBATCH;
]
SIGSEL=SQRT ((DESCI2-AVSPE*AVSPE)/(NBATCH-1));
AVSPE=AVSPE/DELTA E;
SIGSEL=SIGSEL/DELTA E;

"Positron spectrum per MeV per source"
/AVSPP,DESCI2/=0.0;
DO J=1,NBATCH [
AVSPP=AVSPP+SPPP (ND, IE, J)/NBATCH;
DESCI2=DESCI2+SPPP (ND, IE, J)*SPPP (ND, IE, J)/NBATCH;
]
SIGSPO=SQRT ((DESCI2-AVSPP*AVSPP)/(NBATCH-1));
AVSPP=AVSPP/DELTA E;
SIGSPO=SIGSPO/DELTA E;

OUTPUT EUP,AVSPG,SIGSPH,AVSPE,SIGSEL,AVSPP,SIGSPO;
(G10.5,' MeV--',3(G12.5,'+-',G12.5));
]
] "end of detector loop"

"NEXT, CALL THE SUBROUTINE ECNSV1 TO WRITE-OUT THE ENERGY DEPOSITION"
"TOTALS---TO CHECK ENERGY CONSERVATION FOR ONE THING"

CALL ECNSV1(1,NREG,TOTKE);

CALL NTALLY(1,NREG);

STOP;
END; "END OF MAIN PROGRAM"

%E
"*****"
" STANFORD LINEAR ACCELERATOR CENTER"
SUBROUTINE AUSGAB(IARG);
" EGS4 SUBPROGRAM - 8 MAY 1983/1730"
"*****"
;COMIN/DEBUG,EPCONT,ETALY1,LINES,MISC,NTALY1,PASSIT,STACK,TOTALS,USEFUL/;
REAL*8 DPWT,DEPE;
IRL=IR(NP); "SET LOCAL VARIABLE"
DPWT=WT(NP);

"KEEP TRACK OF THE ENERGY DEPOSITION---FOR CONSERVATION PURPOSES"
ESUM(IQ(NP)+2,IRL,IARG+1)=ESUM(IQ(NP)+2,IRL,IARG+1)+EDEP*DPWT;
NSUM(IQ(NP)+2,IRL,IARG+1)=NSUM(IQ(NP)+2,IRL,IARG+1) + 1;

IF(MED(IRL).EQ.1) ["particle is inside the detector"
DEPE=DEPE+EDEP; "Add energy deposition"
IF(IRL.NE.IROLD.AND.IARG.EQ.0) [ "particle enters into detector"
IF(IQ(NP).EQ.0) ["photon"
IE=E(NP)/DELTA E+1;
IF(IE.LE.$NEBIN) [SPG(1,IE)=SPG(1,IE)+DPWT;]]
ELSEIF(IQ(NP).EQ.-1) ["Electron"
IE=(E(NP)-RM)/DELTA E+1;
IF(IE.LE.$NEBIN) [SPE(1,IE)=SPE(1,IE)+DPWT;]]
ELSE ["Positron"
IE=(E(NP)-RM)/DELTA E+1;
IF(IE.LE.$NEBIN) [SPP(1,IE)=SPP(1,IE)+DPWT;]]
] "end of entering to detector"
] "end of inside detector"

IF(NCOUNT.LE.NWRITE.AND.ILINES.LE.NLINES) [
OUTPUT E(NP),X(NP),Y(NP),Z(NP),U(NP),V(NP),W(NP),
IQ(NP),IRL,IARG; (7G15.7,3I5);
ILINES=ILINES+1;]

RETURN;
END; "END OF SUBROUTINE AUSGAB"

```

```

%E
"*****"
"                                STANFORD LINEAR ACCELERATOR CENTER"
SUBROUTINE HOWFAR;
"                                EGS4 SUBPROGRAM - 8 MAY 1983/1730"
"*****"
;COMIN/DEBUG,EPCONT,GEOM,PASSIT,STACK,THRESH/;

IRL=IR(NP); "SET LOCAL VARIABLE"

IF(IRL.LE.1.OR.IRL.GE.IRZ+2) [IDISC=1; RETURN;]

NSLAB=(IRL-2)/NCYL + 1 ; "SLAB NUMBER"
NANNU=IRL-1-NCYL*(NSLAB-1); "ANNULUS NUMBER"
NPL1=NSLAB+1; NPL2=NSLAB;
IF(NSLAB.LT.NPLAN-1) [NRG1=IRL+NCYL;]
ELSE [NRG1=IRZ+2;]
IF(NSLAB.GT.1) [NRG2=IRL-NCYL;]
ELSE [NRG2=1;]

$PLAN2P(NPL1,NRG1,1,NPL2,NRG2,-1);

IF(NANNU.LT.NCYL) [NRG2=IRL+1;]
ELSE [NRG2=IRZ+3;]
IF(NANNU.GT.1) [NRG1=IRL-1; NCL2=NANNU;
NCL1=NANNU-1;
$CYL2(NCL1,NRG1,NCL2,NRG2); RETURN;]

$CYLNR(1,1,IHIT,TCYL);
IF(IHIT.EQ.1) [
$CHGTR(TCYL,NRG2);]

RETURN;
END; "END OF SUBROUTINE HOWFAR"

```

```

%E
"*****"
"                                STANFORD LINEAR ACCELERATOR CENTER"
SUBROUTINE ECNSV1(NTREE,NREG,TOTKE);
"                                EGS4 SUBPROGRAM - 8 MAY 1983/1730"
"*****"
" SUBROUTINE FOR KEEPING TRACK OF ENERGY CONSERVATION---TO BE "
" USED WITH EGS USER CODES. WHEN NTREE=0, THE PROGRAM IS "
" ENTERED IN ORDER TO INITIALIZE THE ESUM ARRAY TO ZERO. "
" OTHERWISE, IT IS ENTERED FOR TOTALING AND OUTPUTTING THE "
" RESULTS. THE ESUM ARRAY IS NEEDED IN SUBROUTINE AUSGAB, "
" WHERE EDEP (ENERGY DEPOSITION) IS ADDED TO THE ELEMENT OF "
" THE ARRAY CORRESPONDING TO THE VALUE OF IQ, IR, AND IARG. "
"*****"
COMIN/DEBUG,ETALY1/; "INSERT IN ALL SUBPROGRAMS THAT USE ESUM"
REAL*8 ROWSUM(4,$MXREG),COLSUM(4,5),SUMSUM(4),GSUM,TOTKE;

" CHECK WHETHER NREG IS GE $MXREG. IF IT IS, STOP AND OUTPUT. "
IF(NREG.GT.$MXREG) [
MDUMMY=$MXREG;
OUTPUT NREG,MDUMMY;
(///,' ***** NOTE: STOPPED IN SUBROUTINE ECNSV1 BECAUSE NREG= ',
I5,' IS LARGER THAN $MXREG= ',I5,' *****');
STOP;]

IF(NTREE.EQ.0) [ "INITIALIZE ESUM TO ZERO AND RETURN"
DO I=1,4 [DO J=1,NREG [DO K=1,5 [ESUM(I,J,K)=0.DO;]]]
RETURN;]

" REACH THIS POINT WHEN FINAL TALLY IS TO BE MADE."

" FIRST, INITIALIZE SUMS"

GSUM=0.DO;

DO IQ=1,4 [
SUMSUM(IQ)=0.DO;
DO IR=1,NREG [ROWSUM(IQ,IR)=0.DO;]

```

```

DO ICODE=1,5 [COLSUM(IQ,ICODE)=0.DO;]
" END OF IQ-LOOP"]

" SUM IQ=1,2,3 INTO IQ=4 OF ESUM FOR ALL IR- AND ICODE-VALUES"

DO IR=1,NREG [
DO ICODE=1,5 [
DO IQ=1,3 [
ESUM(4,IR,ICODE)=ESUM(4,IR,ICODE) + ESUM(IQ,IR,ICODE);
]]]

" NORMALIZE DATA TO TOTKE"

DO IQ=1,4 [
DO IR=1,NREG [
DO ICODE=1,5 [
ESUM(IQ,IR,ICODE)=ESUM(IQ,IR,ICODE)/TOTKE;
]]]

" SUM-UP COLUMNS AND ROWS"

DO IQ=1,4 [
DO IR=1,NREG [
DO ICODE=1,5 [
ROWSUM(IQ,IR)=ROWSUM(IQ,IR)+ESUM(IQ,IR,ICODE);
]]]

DO ICODE=1,5 [
DO IR=1,NREG [
COLSUM(IQ,ICODE)=COLSUM(IQ,ICODE)+ESUM(IQ,IR,ICODE);
]]]
" END OF IQ-LOOP"]

" NOW GET TOTAL FOR IQ AND GRAND TOTAL"

DO IQ=1,4 [
DO IR=1,NREG [
DO ICODE=1,5 [
SUMSUM(IQ)=SUMSUM(IQ)+ESUM(IQ,IR,ICODE);
IF(IQ.LE.3) [GSUM=GSUM+ESUM(IQ,IR,ICODE);]
]]]

" NOW WRITE-OUT THE RESULTS OF THE ENERGY DEPOSITION SUMMARY"

DO IQ=1,4 [
IF(IQ.LE.3) [
IQNOW=IQ-2;
OUTPUT IQNOW;
('1ENERGY DEPOSITION SUMMARY FOR PARTICLES WITH IQ=',I2,///<,
55X,'IARG',/,19X,'0',15X,'1',13X,'2',14X,'3',14X,'4',16X,'ROW SUM',
/,3X,'REGION',/);
]
ELSE [
OUTPUT; ('1ENERGY DEPOSITION SUMMARY FOR ALL PARTICLES:',///<,
55X,'IARG',/,19X,'0',15X,'1',13X,'2',14X,'3',14X,'4',16X,'ROW SUM',
/,3X,'REGION',/);
]
]

DO IR=1,NREG [
OUTPUT IR, (ESUM(IQ,IR,ICODE), ICODE=1,5), ROWSUM(IQ,IR);
(I7,5X,5G15.7,5X,G15.7);
" END OF IR-LOOP"]

OUTPUT (COLSUM(IQ,ICODE), ICODE=1,5), SUMSUM(IQ);
(/,3X,'COL SUM',2X,5G15.7,5X,G15.7);

" END OF IQ-LOOP"]

OUTPUT GSUM; (/////, ' TOTAL FRACTION=',G15.7,
' NOTE: THIS NUMBER SHOULD BE VERY CLOSE TO UNITY');

```

```

RETURN;
END; "END OF SUBROUTINE ECNSV1"

%E
"*****"
" STANFORD LINEAR ACCELERATOR CENTER"
SUBROUTINE NTALLY(NTREE,NREG);
" EGS4 SUBPROGRAM - 8 MAY 1983/1730"
"*****"
" THIS PROGRAM KEEPS 'TALLY' OF THE NUMBER OF TIMES AUSGAB IS "
" ENTERED FOR VARIOUS REASONS, ETC. IT GIVES THE USER A 'ROUGH' "
" IDEA OF WHETHER CERTAIN EVENTS ARE RARE OR NOT. "
" CAUTION *** DO NOT USE THESE NUMBERS IN ANY STATISTICAL SENSE] "
"*****"
COMIN/DEBUG,NTALY1/;
INTEGER ROWSUM(4,$MXREG),COLSUM(4,5),SUMSUM(4),GSUM;

" CHECK WHETHER NREG IS GE $MXREG. IF IT IS, STOP AND OUTPUT."
IF(NREG.GT.$MXREG) [
MDUMMY=$MXREG;
OUTPUT NREG,MDUMMY;
(///,' ***** NOTE: STOPPED IN SUBROUTINE NTALLY BECAUSE NREG= ',
I5,' IS LARGER THAN $MXREG= ',I5,' *****');
STOP;]

IF(NTREE.EQ.0) [ "INITIALIZE NSUM TO ZERO AND RETURN"
DO I=1,4 [DO J=1,NREG [DO K=1,5 [NSUM(I,J,K)=0;]]]
RETURN;]

" REACH THIS POINT WHEN FINAL TALLY IS TO BE MADE."

" FIRST, INITIALIZE SUMS"

GSUM=0;

DO IQ=1,4 [
SUMSUM(IQ)=0;
DO IR=1,NREG [ROWSUM(IQ,IR)=0;]
DO ICODE=1,5 [COLSUM(IQ,ICODE)=0;]
" END OF IQ-LOOP"]

" SUM IQ=1,2,3 INTO IQ=4 OF NSUM FOR ALL IR- AND ICODE-VALUES"

DO IR=1,NREG [
DO ICODE=1,5 [
DO IQ=1,3 [
NSUM(4,IR,ICODE)=NSUM(4,IR,ICODE) + NSUM(IQ,IR,ICODE);
]]]

" SUM-UP COLUMNS AND ROWS"

DO IQ=1,4 [
DO IR=1,NREG [
DO ICODE=1,5 [
ROWSUM(IQ,IR)=ROWSUM(IQ,IR)+NSUM(IQ,IR,ICODE);
]]]

DO ICODE=1,5 [
DO IR=1,NREG [
COLSUM(IQ,ICODE)=COLSUM(IQ,ICODE)+NSUM(IQ,IR,ICODE);
]]]
" END OF IQ-LOOP"]

" NOW GET TOTAL FOR IQ AND GRAND TOTAL"

DO IQ=1,4 [
DO IR=1,NREG [
DO ICODE=1,5 [
SUMSUM(IQ)=SUMSUM(IQ)+NSUM(IQ,IR,ICODE);
IF(IQ.LE.3) [GSUM=GSUM+NSUM(IQ,IR,ICODE);]
]]]

```

```

" NOW WRITE-OUT THE RESULTS"

DO IQ=1,4 [
IF(IQ.LE.3) [
IQNOW=IQ-2;
OUTPUT IQNOW;
('1SUMMARY OF EVENT COUNT FOR PARTICLES WITH IQ=',I2,///<,
55X,'IARG',/,19X,'0',15X,'1',13X,'2',14X,'3',14X,'4',16X,'ROW SUM',
/,3X,'REGION',/);
]
ELSE [
OUTPUT; ('1SUMMARY OF EVENT COUNT FOR ALL PARTICLES:',///<,
55X,'IARG',/,19X,'0',15X,'1',13X,'2',14X,'3',14X,'4',16X,'ROW SUM',
/,3X,'REGION',/);
]

DO IR=1,NREG [
OUTPUT IR,(NSUM(IQ,IR,ICODE),ICODE=1,5),ROWSUM(IQ,IR);
(I7,5X,5I15,5X,I15);
" END OF IR-LOOP"]

OUTPUT (COLSUM(IQ,ICODE),ICODE=1,5),SUMSUM(IQ);
(/,3X,'COL SUM',2X,5I15,5X,I15);

" END OF IQ-LOOP"]

OUTPUT GSUM; (/////, ' TOTAL NUMBER OF EVENTS=',I15);

RETURN;
END; "END OF SUBROUTINE NTALLY"

%E
%C80
"-----"
" Start of PRSTAAUX MORTRAN - Auxiliary codes required by PRESTA. "
"----- Taken from NRCCAUXP MORTRAN. "
" 24 August 1989 WRN "
"-----"
" "
"*****"
" * * "
" * FIXTMX * "
" * * "
" ***** "
SUBROUTINE FIXTMX(ESTEP,MEDIUM);
"
" THIS ROUTINE CHANGES THE STEP SIZE ALGORITHM USED IN EGS SO THAT "
" THE STEP SIZE ARRAYS FOR TMXS CORRESPOND TO AN ARBITRARY,BUT "
" FIXED FRACTIONAL ENERGY LOSS ESTEPE. "
" IT IS ONLY NECESSARY FOR LOW ENERGY ELECTRON PROBLEMS SINCE "
" TYPICALLY THE 200*TEFFO RESTRICTION ON TMXS IS MORE STRINGENT "
" FOR ELECTRONS WITH ENERGIES ABOVE A FEW MEV "
"
" NOTE THAT THE $TMXS-OVER-RIDE MACRO MAY STILL BE IN FORCE IN EGS. "
"
" THE ROUTINE CHANGES THE VALUES ONLY FOR THE MEDIUM 'MEDIUM' "
" AND IT SHOULD PROBABLY BE USED FOR ALL MEDIA IN A PROBLEM. "
";"
" THE ROUTINE MUST BE CALLED AFTER HATCH HAS BEEN CALLED AND BEFORE "
" THE SIMULATION IS BEGUN. "
"
" THE ROUTINE IS INDEPENDENT OF WHAT UNITS ARE BEING USED, AS LONG "
" AS THEY ARE CONSISTENT( E.G. CM, RL OR G/CM**2 ) "
"
" IF CALLED WITH IOLDTM=0 (PASSED IN COMIN USER) THE TMXS ARRAYS ARE "
" ADJUSTED TO GIVE A FIXED ESTEPE AND ARE SUBJECTED TO THE TMIN AND "
" CONSTRAINTS. "
" IF CALLED WITH IOLDTM=1 THE CURRENT EGS ALGORITHM IS USED. "
" IF CALLED WITH IOLDTM=0 AND ESTEPE=0 THE CURRENT EGS ALGORITHM IS "
" USED. "
" IF CALLED WITH IOLDTM=1 AND ESTEPE=0 THEN ESTEPE=1.0 IS USED. "

```



```

"
" FOR A DETAILED DISCUSSION OF THE USE OF THIS ROUTINE, SEE "
" 'Low Energy Electron Transport with EGS' in Nuclear Instr. and "
" Methods A227 (1984)535-548. D.W.O. Rogers "
"
" FOR A DISCUSSION OF THE NEW FEATURES (VO3+) OF THIS ROUTINE, "
" ESPECIALLY WITH REGARD TO THE NEW UPPER AND LOWER LIMITS, SEE "
" 'PRESTA-the Parameter Reduced Electron-Step Transport Algorithm- "
" for Electron Monte Carlo Transport' by A.F.Bielajew & D.W.O.Rogers," "
" NRCC Internal Report PIRS-042 obtainable by contacting the above. "
"
" VO1 DEC 10,1981 DAVE ROGERS NRCC "
" VO2 DEC 1984 EGS4 VERSION "
" VO3 JAN 1986 ALEX BIELAJEW NRCC REVISED FOR PRESTA "
"*****"
;COMIN/ELECN,MEDIA,USER/;

ESTEPE=ESTEP;

IF(MEDIUM > $MXMED) ["ERROR" OUTPUT MEDIUM;
(///'0***** MEDIUM=',I4,' IN FIXTMX IS TOO LARGE');RETURN;]

IF((ESTEPE = 0.) & (IOLDTM = 1)) RETURN; "USE THE CURRENT ALGORITHM "
IF(ESTEPE = 0.) ESTEPE=1.; "NEW VERSION DEFAULTS TO TOTAL ENERGY LOSS"
IF(IOLDTM = 0) [BLCCC=BLCC(MEDIUM);XCC2=XCC(MEDIUM)**2; "NEEDED BY ROOTMX"]
"SET UP SOME VARIABLES FOR FIRST PASS THROUGH LOOP"
EI =EXP( (1.-EKEO(MEDIUM))/EKE1(MEDIUM));"ENERGY OF FIRST TABLE ENTRY"
EIL = ALOG(EI); LEIL=1;
"THIS IS EQUIVALENT TO $SETINTERVAL EIL,EKE; BUT AVOIDS ROUND OFF"
$EVALUATE EDEDX USING EDEDX(EIL);"GET THE ELECTRON STOPPING AT EI"
"NOW CALCULATE STEP REQUIRED TO CAUSE AN ESTEPE REDUCTION IN ENERGY"
IF(IOLDTM = 1) [SI=ESTEPE*EI/EDEDX;]ELSE [SI=ROOTMX(EI,ESTEPE);]
"TABULATED ENERGIES ARE IN A FIXED RATIO - CALC LOG OF THE RATIO"
ERATIO=-1./EKE1(MEDIUM);

NEKE=MEKE(MEDIUM);"NUMBER OF ELEMENTS IN STORAGE ARRAY"
DO I=1,NEKE-1[
EIP1=EXP((FLOAT(I+1)-EKEO(MEDIUM))/EKE1(MEDIUM));"ENERGY AT I+1"
EIP1L=ALOG(EIP1);LEIP1L=I+1;"DESIGNED THIS WAY=$SETINTERVAL"
$EVALUATE EDEDX USING EDEDX(EIP1L);
IF(IOLDTM = 1) [SIP1=ESTEPE*EIP1/EDEDX;]ELSE [SIP1=ROOTMX(EIP1,ESTEPE);]

"NOW SOLVE THESE EQUATIONS "
" SI = TMXS1 * EIL + TMXSO "
" SIP1 = TMXS1 * EIP1L + TMXSO "
"
TMXS1(I,MEDIUM)=(SI-SIP1)/ERATIO;TMXSO(I,MEDIUM)=SI-TMXS1(I,MEDIUM)*EIL;
"TRANSFER VALUES FOR NEXT LOOP"
EIL=EIP1L;SI=SIP1;]
"NOW PICK UP LAST TABLE ENTRY WHICH APPLIES ONLY TO LAST ENERGY"
TMXSO(NEKE,MEDIUM)=TMXSO(NEKE-1,MEDIUM);
TMXS1(NEKE,MEDIUM)=TMXS1(NEKE-1,MEDIUM);
RETURN;END;
%E

"*****"
" * * "
" * ROOTMX * "
" * * "
" ***** "
"
FUNCTION ROOTMX(EI,ESTEP);
"
" THIS ROUTINE RETURNS MAX(TMIN,MIN(TMAX,ESTEPE*EI/DEDX)) WHERE "
" TMAX IS THE MAXIMUM STEP ALLOWED BY THE MOLIERE MULTIPLE SCATTERING "
" THEORY, TMIN IS THE THE MINIMUM STEP AND ESTEPE*EI/DEDX IS THE GREATEST "
" STEP ALLOWED DUE TO CONTINUOUS ENERGY LOSS PROCESSES. "
"
" NOTE THE USE OF ITS AUXILLIARY FUNCTION FTMX APPENDED TO ROOTMX. "
" BECAUSE THE TMAX FUNCTION IS STRONGLY ENERGY DEPENDENT, IT WAS FOUND "
" NECESSARY TO INCLUDE A CORRECTION FOR ENERGY LOSS IN IT. OTHERWISE THE "

```

```

"    UPPER LIMIT COULD BE GREATLY EXCEEDED - BY AS MUCH AS 50% IN SOME CASES.  "
"    CORRECTING FOR ENERGY LOSS NECESSITATES USING A ROOT FINDING METHOD TO  "
"    OBTAIN TMAX (HENCE THE NAME ROOTMX). TMIN IS ALSO STRONGLY ENERGY  "
"    DEPENDENT BUT IT DOES NOT MATTER WITHIN THE LOGIC OF THE CODE IF THIS  "
"    QUANTITY IS AS MUCH AS 50% HIGH SINCE NO PHYSICS CONSTRAINTS WILL BE  "
"    VIOLATED.  "
"  "
"    THE ZERO-FINDING ROUTINE IS A CRUDE ONE BASED ON THE ASSUMPTION THAT  "
"    THE FUNCTION FTMX IS MONOTONIC AND THAT THE FUNCTION EVALUATED AT THE TWO  "
"    STARTING POINTS RETURNS DIFFERENT SIGNS. IF THE SIGNS ARE THE SAME THEN  "
"    EITHER THE ENERGY-LOSS STEP-SIZE IS MORE RESTRICTIVE OR THE STEP-SIZE IS  "
"    BELOW TMIN.  "
"  "
"    ALTHOUGH THIS ROUTINE COMES WITH THE PRESTA PACKAGE IT IS REALLY  "
"    INDEPENDENT OF IT AND IT IS AN IMPROVEMENT OVER THE PREVIOUS TMSX METHODS.  "
"    THE OLD TMSX ROUTINE ALLOWED BOTH THE TMAX AND TMIN BOUNDS TO BE VIOLATED.  "
"    EXCEEDING TMAX TAKES ONE OUT OF THE REGION OF VALIDITY OF THE MOLIERE  "
"    THEORY STILL ALLOWING A MULTIPLE SCATTERING SELECTION BUT OF UNPREDICTABLE  "
"    WORTH. GOING LOWER THAN TMIN CAUSES THE MULTIPLE SCATTERING TO GET  "
"    SWITCHED OFF (STARTING WITH THE LOWER ENERGIES). THIS CAN SOMETIMES LEAD  "
"    TO CALCULATIONAL ARTIFACTS. ONE WORD OF CAUTION] USING THIS ROUTINE AT  "
"    VERY LOW ELECTRON ENERGIES .LE.10 keV CAUSES NEGATIVE USTEP ERRORS IF THE  "
"    OLD EGS PATHLENGTH CORRECTION ALGORITHM (BASED ON FERMI-EYGES THEORY) IS  "
"    USED. THE OLD EGS LESSENED THIS PROBLEM BY REDUCING THE UPPER LIMIT TO  "
"    0.8 THE VALUE USED IN THIS ROUTINE. THE PRESTA PATHLENGTH CORRECTION DOES  "
"    NOT GIVE NEGATIVE USTEPS IN ANY OF THE CASES WE HAVE TESTED.  "
"  "
"          VERSION 1          ALEX BIELAJEW          JAN. 86  "
"          VERSION 1.1        ALEX BIELAJEW          OCT. 87  "
"                                Lower limit ESTEPE violation fixed  "
"  "
"*****"

```

```

;
COMIN/USEFUL,USER/;
ESTEPE=ESTEP;
TMIN=2.718282*EI*(EI+2.*RM)/(BLCCC*(EI+RM)**2); "LOWER LIMIT, eq.(2-8)"
X1=TMIN; "INITIAL LOWER STARTING POINT OF THE SEARCH"
X2=ESTEPE*EI/EDEX; "INITIAL UPPER STARTING POINT OF THE SEARCH"

"THIS IS THE FIX-UP FOR THE MINIMUM STEP-SIZE"
IF( X2 <= X1 ) [ROOTMX=X1;RETURN;]

F1=FTMX(X1,EI);F2=FTMX(X2,EI);
AF1=ABS(F1);AF2=ABS(F2);
SF1=SIGN(1.,F1);SF2=SIGN(1.,F2);

"FIRST CHECK TO SEE IF EITHER OF THE STARTING POINTS IS ALREADY GOOD ENOUGH."
IF((AF1 <= $ROOTMX_PRECISION) | (AF2 <= $ROOTMX_PRECISION))[
  IF(AF1 <= AF2) [ROOTMX=X1;]ELSE [ROOTMX=X2;]

"NOW CHECK TO SEE IF EITHER THE ENERGY LOSS IS MORE RESTRICTIVE THAN THE  "
"UPPER LIMIT TMAX (TRUE FOR HIGH ENERGIES) OR IF IT MORE RESTRICTIVE THAN  "
"THE TMIN (TRUE FOR LOW ENERGIES WITH A SMALL ENOUGH ESTEPE).  "
ELSEIF(SF1 = SF2) [ROOTMX=X2;]

"OTHERWISE A SEARCH FOR TMAX MUST BE UNDERTAKEN."
ELSE[ "ITERATE"
ITI=0; "NUMBER OF ITERATIONS COUNTER"
XL=X1; "LAST X FOUND"
:SEARCH-ROOT:LOOP[
ITI=ITI+1;
IF(ITI > 1000) [ "QUIT IF THIS HAPPENS"
OUTPUT;(' SEARCH FOR TMSX ABORTED. TOO MANY ITERATIONS');STOP;]
XT=(X1*F2-X2*F1)/(F2-F1);
IF(XT = XL) [ROOTMX=XT;EXIT:SEARCH-ROOT;:"CONVERGENCE OBTAINED"]
FT=FTMX(XT,EI);AFT=ABS(FT);
IF(AFT <= $ROOTMX_PRECISION) [ROOTMX=XT;EXIT:SEARCH-ROOT;:"CONVERGENCE OBTAINED"]
ELSE[ "RE-ITERATE"
SFT=SIGN(1.,FT);
IF(SFT = SF1) [X1=XT;F1=FT;AF1=AFT;SF1=SFT;]ELSE [X2=XT;F2=FT;AF2=AFT;SF2=SFT;]
XL=XT; "UPDATE LAST X FOUND"
]
]

```

```

] "END OF SEARCH FOR ROOT LOOP"
] "END OF ITERATE ELSE"
RETURN;END;

FUNCTION FTMX(T,EI);
"When t=tmax as defined in eq.(2-10) this function returns 0. It is used by      "
"FUNCTION ROOTMX in the search for tmax.                                     "
COMIN/USEFUL,USER/;
"Energy dependent quantities are evaluated at the energy mid-point of the step."
"See section IV of the report PIRS-042.                                     "
EK=AMAX1(0.0001,EI-0.5*EDEDX*T);E=EK+RM;BETA2=EK*(E+RM)/E**2;
A=BLCCC/BETA2;G=XCC2/(E*BETA2)**2;
FTMX=1./ALOG(A/G)-G*T;
RETURN;END;
%E

"*****"
SUBROUTINE RMARIN;
"*****"
COMIN/RANDOM/;

IF((IXX.LE.0).OR.(IXX.GT.31328)) IXX=1802; "SETS MARSAGLIA DEFAULT"

" BUG. In the following line the assignment previous to 90/09/18 "
" was to IXX. This DID NOT upset the randomness of the sequence, "
" just the initial starting point. BLIF 90/09/18. "

IF((JXX.LE.0).OR.(JXX.GT.30081)) JXX=9373; "SETS MARSAGLIA DEFAULT"

I = MOD(IXX/177,177) + 2;
J = MOD(IXX, 177) + 2;
K = MOD(JXX/169,178) + 1;
L = MOD(JXX, 169) ;

DO II=1,97[
  S=0.0;T=0.5;
  DO JJ=1,24[
    M=MOD(MOD(I*J,179)*K,179);
    I=J;J=K;K=M;L=MOD(53*L+1,169);
    IF(MOD(L*M,64).GE.32) S=S+T;
    T=0.5*T;
  ]
  URNDM(II)=S;
]

CRNDM = 362436./16777216.;
CDRNDM = 7654321./16777216.;
CMRNDM = 16777213./16777216.;

IXX = 97;
JXX = 33;

RETURN;END;

"*****"
"***** UCNAI3P END *****"
"*****"
%N

```