

EGS4のジオメトリの書き方

波戸芳仁 平山英夫
高エネルギー加速器研究機構

29 JUL 2003 改訂

EGS4のジオメトリー

- EGS4ではジオメトリーの単位を“**region (領域)**”と呼ぶ。
- EGS4のユーザーコードでのジオメトリー記述:
 - **MAIN**: 面の定義、領域への物質の割り当て。
領域毎に物質が異なってもいいし、領域をまたがって同じ物質を指定してもいい。
 - **SUBROUTINE HOWFAR** : 粒子が面を通過する事の判定。面と領域の関連づけ。

ジオメトリー構造の選択

1. 定型 (多重円筒平板、多重平板)

- i. 専用HOWFAR有り
- ii. 面の数、位置などを入力

2. 非定形: CG-PRESTA, MARS

- i. 専用HOWFAR有り
- ii. 直方体、円柱、円筒、球などの大きさ、位置、組み合わせなどを入力
- iii. ジオメトリー表示システム
- iv. ~ 10倍遅い

3. 非定形: 自作HOWFAR

- i. 自由度大
- ii. CGに比べて速い

USTEP, IDISC, IRNEW

- HOWFARで重要な働きをするEGS4の3変数
- COMMON/EPCONT/に含まれる
- **USTEP:** 次の場所までの距離. 光子の場合には、相互作用点までの距離が設定され、次にHOWFARが呼ばれる。

HOWFARを呼ぶ前には現在地の領域 (IR(NP))に対して設定される。
- **IDISC:1** の時に*discard region (廃棄領域)*を示す。
- **IRNEW:** 粒子がある距離を移動した後の領域番号

HOWFARの機能

- もし現領域が *discard region (廃棄領域)* なら
 - **IDISC=1** を設定して return する
- 境界までの直線距離 (DIST) を計算.
 - もし **DIST < USTEP** なら
 - **USTEP** を **DIST** に縮め、**IRNEW** を粒子が次に入る領域の番号 (**NEXTREG**) に設定する.

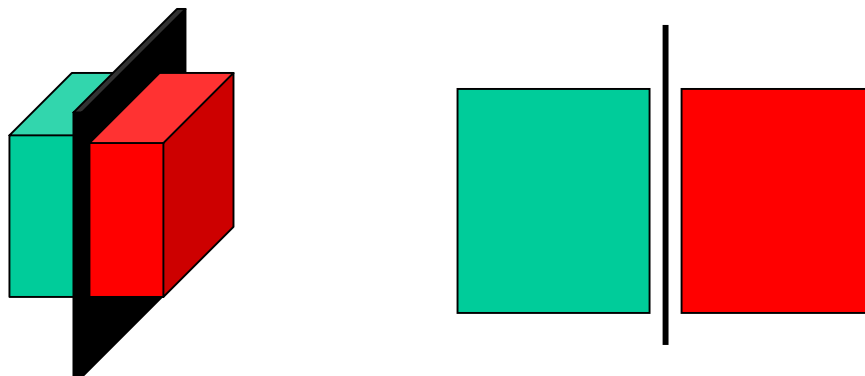
USTEP=DIST;

IRNEW=NEXTREG;

境界までの距離の計算方法は？

- ユーザーは境界までの距離を自分の方法で計算できる。
- EGS4 システムにはいくつかのジオメトリ用マクロ(またはサブルーチン)が含まれている。
 - \$PLANE1 (PLANE1), \$PLAN2P (PLAN2P), \$PLAN2X (PLAN2X)
 - \$CYLNDR (CYLNDR), \$CYL2 (CYL2)
 - \$CONE (CONE), \$CON2 (CONE2)
 - \$SPHERE (SPHERE), \$SPH2 (SPH2)
- HOWFARの機能を助けるための二つのマクロ(またはサブルーチン).
 - \$CHGTR (CHGTR) : 必要な時に USTEP と IRNEW を書き換える.
 - \$FINVAL (FINVAL) : 粒子輸送後に、新しい座標を得る。

平面で空間を二分する

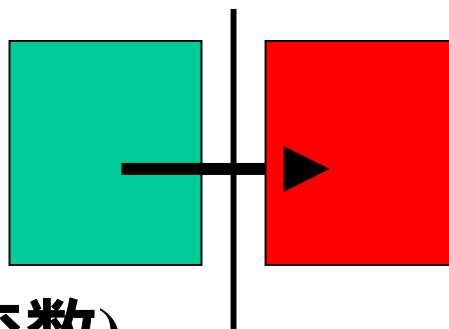


二分された空間を区別する

法線ベクトルの
出発側の空間:

ISIDE=1

(`$PLANE1`での変数)



法線ベクトルの
到着側の空間:

ISIDE= -1

\$PLANE1 マクロ

\$PLANE1(NPL,ISIDE,IHIT,TVAL);

NPL : 調べたい平面の番号

ISIDE : 1 ならば、粒子は法線ベクトルの**出発側**にある。

-1 ならば、粒子は法線ベクトルの**到着側**にある。

IHIT : 1 ならば、粒子の飛跡が平面と交差する

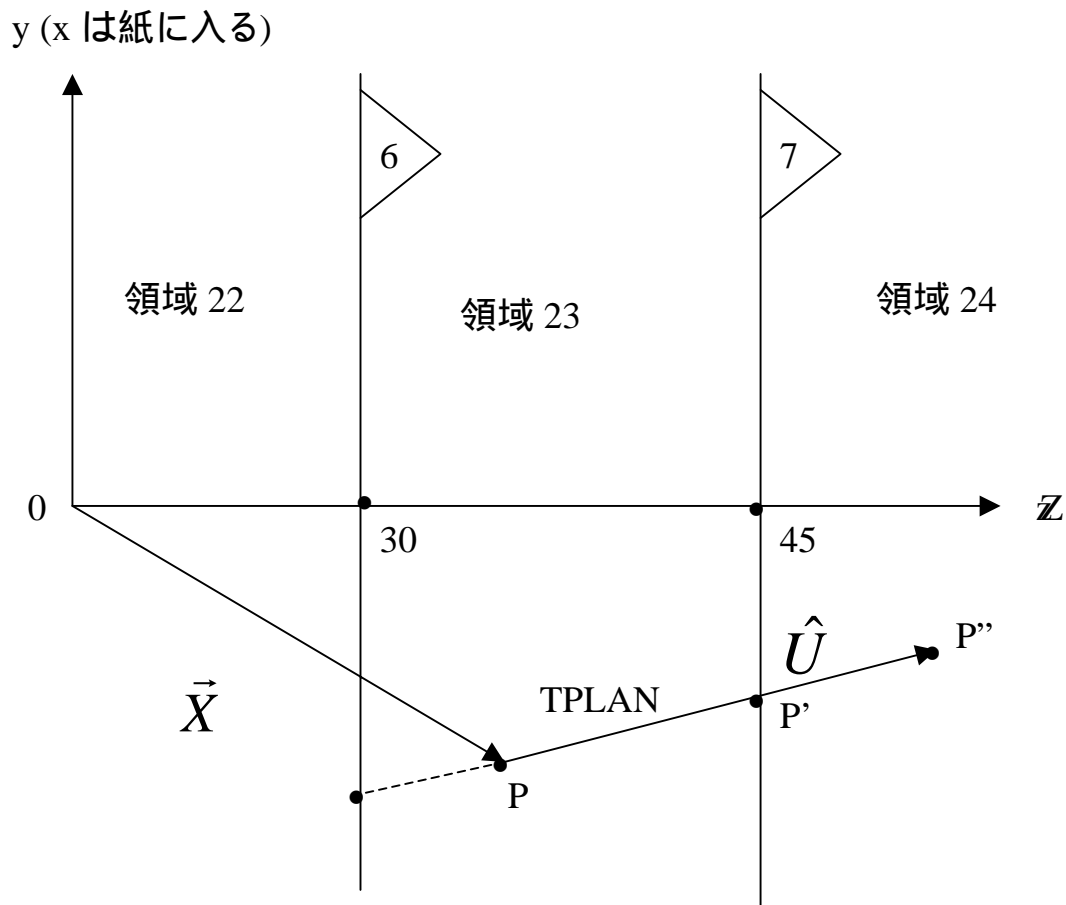
2 ならば、粒子の飛跡と平面は平行

0 ならば、粒子の飛跡は平面から遠ざかる

TVAL : IHIT=1 の場合に、平面までの距離

- 平面は垂直ベクトル (**PNORM(I,J),I=1,3**) と垂直ベクトルと平面の交点の座標 (**PCOORD(I,J),I=1,3**) で定義する。
- 両変数は **COMMON/PLADTA/** に含まれる。

2枚の平行平面で3領域を区切ってみる:



領域は22, 23 と 24 の番号で、平面は 6 と 7 の番号で区別する.

平面の定義

- 平面 6 と 7 はそれぞれ $z=30$ cm と $z=45$ cm に位置すると仮定する。ユーザーコード内でのそれらの記述は、

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;**

•この例のためのHOWFAR. 粒子は最初に領域 23 にあり、そこから出ると廃棄されるという条件を追加してある.

```
SUBROUTINE HOWFAR;  
COMIN/EOCONT,PLADTA,STACK/;  
IRL=IR(NP);  
IF(IRL.NE.23) [IDISC=1; "Discard particles outside region 23"]  
ELSE ["Track particles within region23"]  
    $PLANE1(7,1,IHIT,TPLAN); "Check upstream plane first"  
    IF(IHIT.EQ.1) ["Surface is hit --- make change if necessary"  
        $CHGTR(TPLAN,24);]  
    ELSEIF(IHIT.EQ.0) ["Heading backwards"  
        $PLANE1(6,-1,IHIT,TPLAN); ""To get TPLAN-value (IHIT=1, must)"  
        $CHGTR(TPLAN,22); "Make change if necessary"  
    ]  
]  
RETURN; END;
```

\$CHGTR

- **\$CHGTR** ルーチンで行うことは:
 - もし **TPLAN.LE.USTEP** なら
 - **USTEP=TPLAN** そして
 - **IRNEW=24** (または **22**)
 - そうでない時には何もしない.

```
REPLACE{$CHGTR(,#);} WITH
```

```
{;IF({P1}.LE.USTEP) [USTEP={P1}; IRNEW={P2};]}
```

\$PLAN2P マクロ

- ここまで見てきたHOWFARの例題は **\$PLAN2P** の助けの
よって、さらに単純になる。

```
SUBROUTINE HOWFAR;  
COMIN/EOCONT,PLADTA,STACK/;  
IRL=IR(NP);  
IF(IRL.NE.23) [IDISC=1; "Discard particles outside region 23"]  
ELSE ["Track particles within region23"  
      $PLAN2P(7,24,1,6,22,-1);  
      ]  
RETURN;  
END;
```

•\$PLAN2P のパラメータの意味:

\$PLAN2P(NPL1,NRG1,ISIDE1,NPL2,NRG2,ISIDE2);

NPL1: 最初に調べる平面の番号.

NRG1: 粒子が最初の平面を横切った後で入っていく領域.

ISIDE1: 1 または -1 (PLANE1のISIDEと同じ)

NPL2: 2番目に調べる平面の番号.

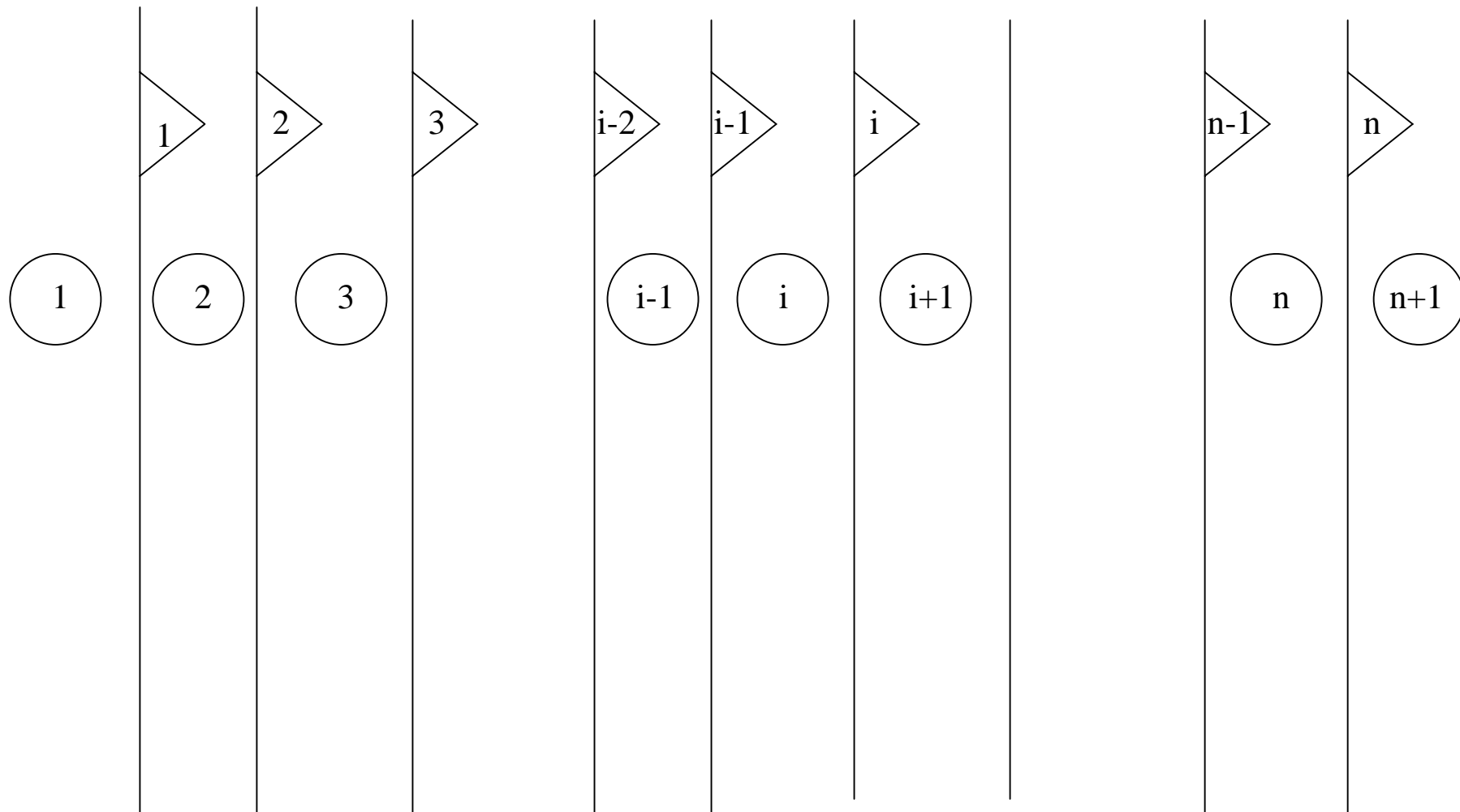
NRG2: 粒子が2番目の平面を横切った後で入っていく領域.

ISIDE2: 1 または -1 (PLANE1のISIDEと同じ)

- 最初のグループの数字 (NPL1,NRG1,ISIDE)(この例では 7,24,1) は下流の平面が **\$PLANE1(7,1,IHIT,TPLAN)** であり **\$CHGTR(TPLAN,24)** が続くことを意味している。
- 2番目のグループの数字 (NPL2,NRG2,ISIDE) (この例では 6,22,-1) は下流の平面が **\$PLANE1(6,-1,IHIT,TPLAN)** であり **\$CHGTR(TPLAN,22)** が続くことを意味している。

$NPL1=IRL;$ $NPL2=IRL-1;$

$NRG1=IRL+1;$ $NRG2=IRL-1;$



領域 I において: 前方の平面番号= i , 前方領域番号= $i+1$

後方の平面番号= $i-1$, 後方の領域番号= $i-1$

多重平板ジオメトリー

- ここまでの HOWFAR を多重平板へと簡単に拡張できる.

```
SUBROUTINE HOWFAR;
```

```
”Multi slab (NREG-1) shower calorimeter.”
```

```
COMIN/EOCONT,PLADTA,STACK/;
```

```
COMMON/PASSIT/NREG; ”To make the number of regions (NREG) available”
```

```
IRL=IR(NP); ”Create a local variable”
```

```
IF(IRL.EQ.1.OR.IRL.EQ.NREG) [IDISC=1; ”Upstream/downstream region ”]
```

```
ELSE [
```

```
    $PLAN2P(IRL,IRL+1,IRL-1,IRL-1,-1);
```

```
]
```

```
RETURN;
```

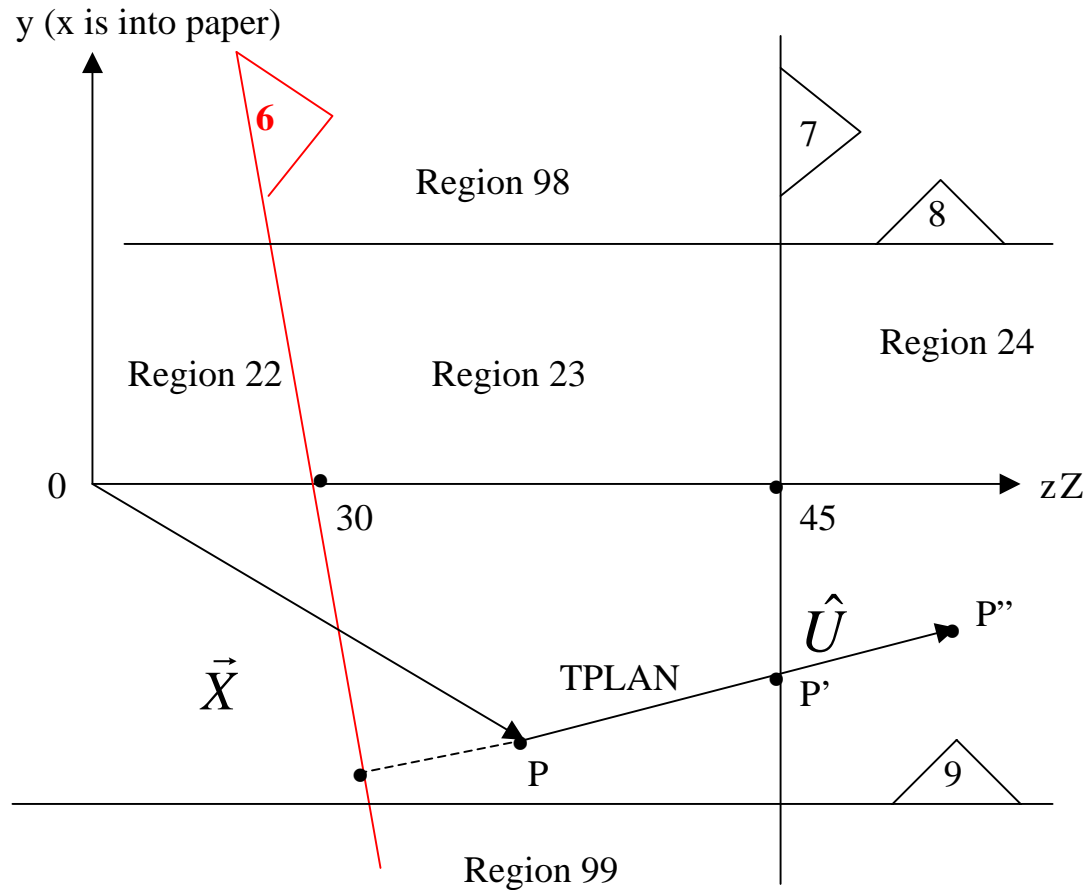
```
END;
```

多重平板ジオメトリのユーザーコードの例

- 半無限平板体系 – ucnai1.mor または ucnai1p.mor
- 有限平板体系 – ucnai2.mor または ucnai2p.mor
- 3次元デカルト体系 – ucnai4.mor または ucnai4p.mor
- これらのサンプルユーザーコードは NaI(Tl) 検出器レスポンスを計算する
- KEK Internal 99-5 の“How to Code Geometry” 参照.

\$PLAN2X マクロ

- 一組の平行平面と一組の**交差する**平面で構成される5領域を想定する。



この例のための HOWFAR。第23領域を出る粒子は廃棄するという条件を追加してある。

```
SUBROUTINE HOWFAR;
```

```
”Two crossing planes (6,7) separating three regions (22,23,24)”
```

```
”with second set of parallel planes (8,9) separating three regions”
```

```
”(98,23,99)”
```

```
COMIN/EPCONT,PLADTA,STACH/;
```

```
COMMON/PASSIT/NREG; ”To make the number of regions (NREF) available”
```

```
IRL=R(NP); ”Create local variable”
```

```
IF(IRL.NE.23) [IDISC=1; ”Discard particles outside region 23”]
```

```
ELSE [”Track particles within region 23”
```

```
    $PLAN2X(7,24,1,6,22,-1);
```

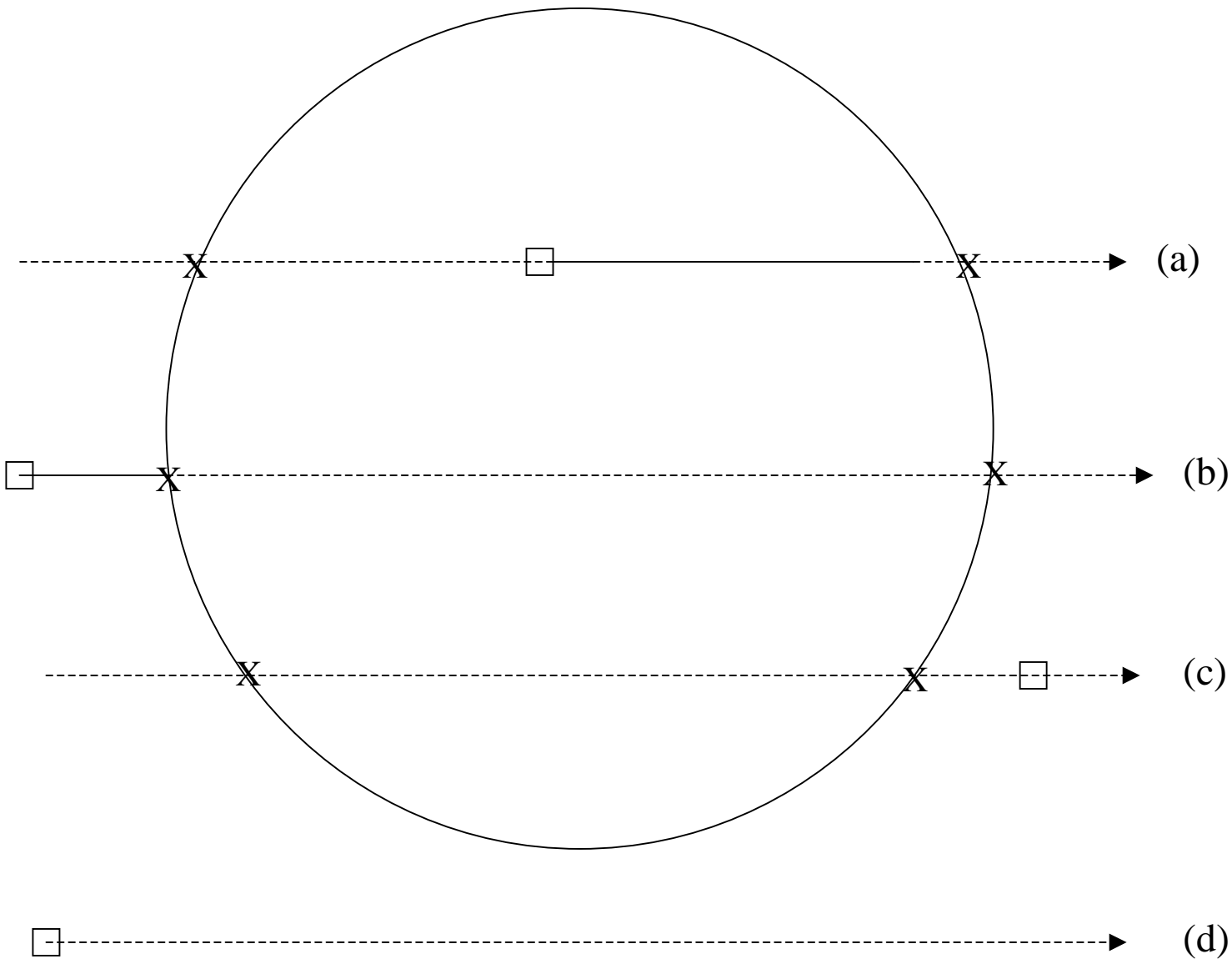
```
    $PLAN2P(8,98,1,9,99,-1);
```

```
    ]
```

```
RETURN; END;
```

\$CYLNDR, \$CONE と \$SPHERE マクロ

- コーン(円錐)表面アルゴリズムも基本的に同じ。
\$CONE と \$SPHERE は \$SUBROUTINE HOWFAR 内で
\$CYLNDR と同じように使える。
- ベクトルとコーン表面の交差は2次式で表され、その解は
実数または複素数となる。これらは実際の幾何形状と対
応する。
- 飛跡が円柱を横切る場合のパターンを次の図に示す (正
方形=出発点, ×印=交差点)。



- \$CYLNDR (CYLNDR) はこれらのすべてのパターンに対応できるように設計されている。ユーザーは粒子の現在地が円筒の **内部** か **外部** かを指定する。
- \$CYLNDR マクロ (Z-軸を中心軸とする円筒用)は、

\$CYLINDR(ICYL,ISIDE,IHIT,TCYL);

ICYL : 調べる円筒のID 番号

ISIDE : 1 ならば粒子は円筒の内部にある

0 ならば粒子は円筒の外部にある

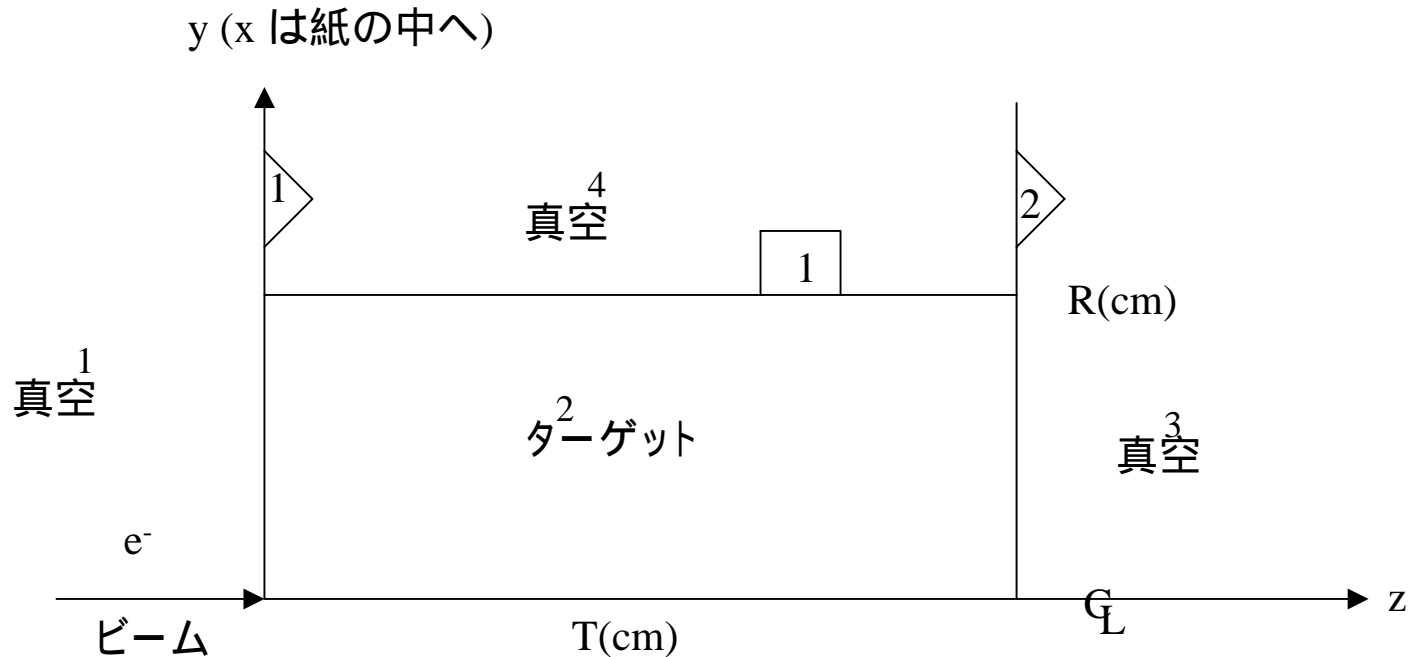
IHIT : 1 ならば粒子は表面と交差する

0 ならば粒子は表面と交差しない

TCYL : IHIT=1 の場合に、表面までの距離

円筒平板(Cylinder-Slab) の例

- 電子ビームが入射する円柱ターゲットを考える:



- Z軸まわりの円筒は箱1で定義される.
- 4領域が設定される - ターゲット (領域 2) と3カ所の真空領域: 上流、下流とターゲットまわり.

•このジオメトリのHOWFAR.

```
SUBROUTINE HOWFAR;
```

```
”Cylinder of rotation about z-axis bounded by two planes.”
```

```
COMIN/CYLDTA,PLADTA,STACH/;
```

```
IRL=R(NP); ”Create local variable”
```

```
IF(IRL.NE.2) [IDISC=1; ”Discard particles outside the target”]
```

```
ELSE [”Track particle within the target”
```

```
    $CYLNDR(1,1,IHL,TCYL); ”Check the cylinder surface”
```

```
    IF(IHIT.EQ.1) [$CHGTR(TCYL,4); ”Change if necessary”]
```

```
    $PLAN2P(2,3,1,1,1,-1);”Check the downstream plane first and”
```

```
        ”then upstream one if necessary.”
```

```
    ]
```

```
RETURN;
```

```
END;
```

円筒面の指定

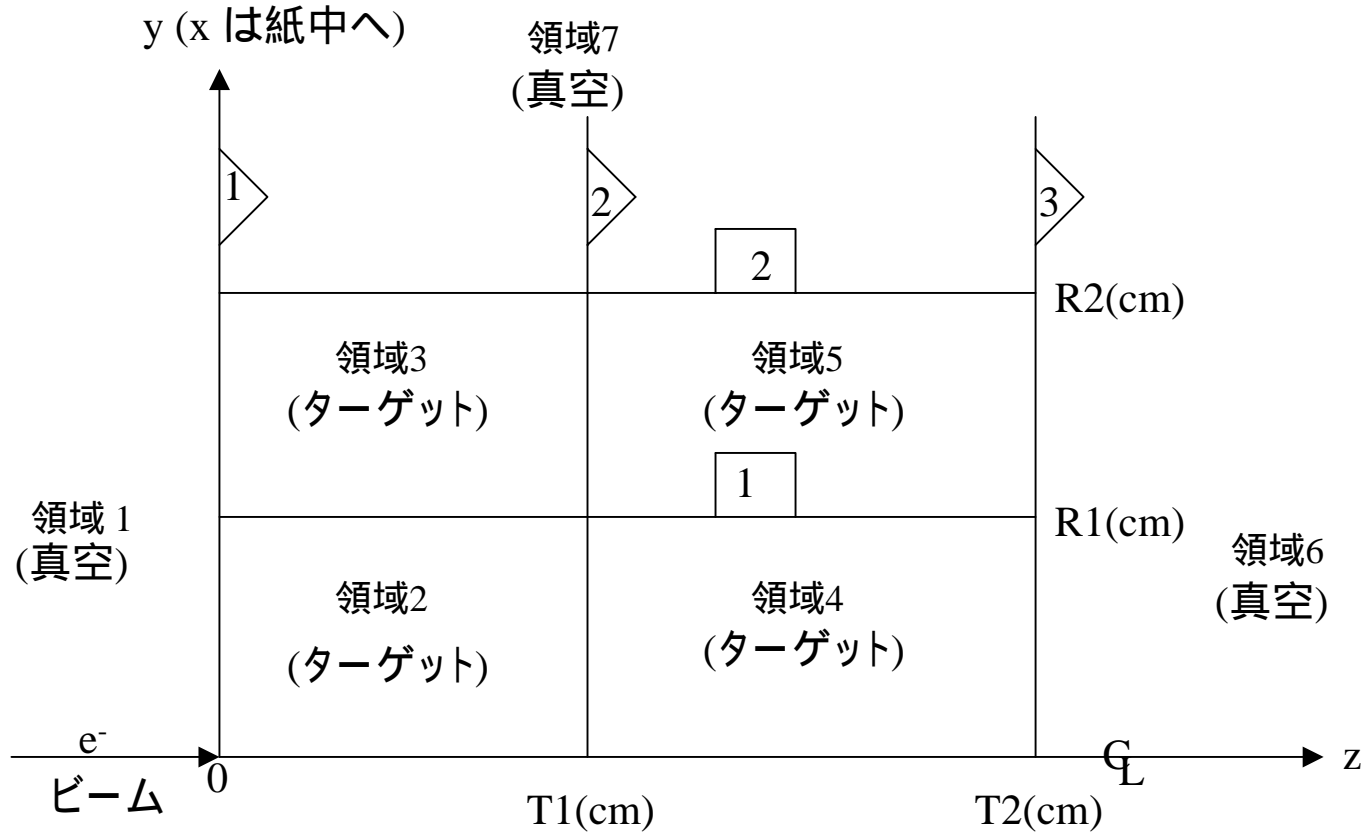
- 円筒の半径(CYRAD)とその二乗(CYRAD2)をMAINで定義しておく.
- COMIN/CYLDTA/;を經由してこれらの量がHOWFARに伝わる.
- 円筒の数の上限は EGS4MAC または KEK4MAC で定義されており、それをユーザーコード内で再定義することも可能.

```
PARAMETER $MXCYLS=75; "Max, No. of cylinders"
```

```
REPLACE {;COMIN/CYLDTA;} WITH  
{;COMMON/CYLDTA/CYRAD2($MXCYLS),CYRAD($MXCYLS);}
```

多重円筒 & 多重平板の例

- 2つの円筒と3枚の平面の例を考える:



- 平行平面のための \$PLAN2P に対応するマクロは **\$CYL2**、これは2つの円筒の間の粒子を扱う。

\$CYL2(NCY1,NRG1,NCY2,NRG2);

NCY1: 最初に調べる円筒(第1円筒)の ID 番号.

粒子を必ず第1円筒の外側におくこと.

NRG1: 粒子が第1円筒と交差した後に入っていく領域.

NCY2: 次に調べる円筒(第2円筒)のID番号.

粒子を必ず第2円筒の内側におくこと.

NRG2: 粒子が第2円筒と交差した後に入っていく領域.

```

SUBROUTINE HOWFAR;
COMIN/CYLDTA,EPCONT,PLADTA,STACH/;
COMMON/PASSIT/NREG,NPLAN,NCYL,IRZ;
IRL=IR(NP); "Create local variable"
IF(IRL.LE.1.OR.IRL.GE.IRZ+2) [IDISC=1; RETURN; ]
NSLAB=(IRL-2)/NCYL+1; "Slab number. NCYL:number of cylinder"
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

```

NPLAN:平面の数

NCYL:円筒の数

NREG=(NPLAN-1)*NCYL+3;

IRZ=NREG-3;

- この HOWFAR で任意の数の平面と円筒を持つジオメトリーが表される.
- 多円筒 & 多平面のサンプルユーザーコード:
 - ucnai3.mor と ucnai3p.mor
 - これらのユーザーコードも NaI(Tl) 検出器のレスポンスを計算する.
 - KEK Internal 99-5 の “How to Code Geometry of EGS4” 参照.

\$FINVAL macro

- \$FINVAL マクロは粒子の飛行後の最終座標を求める.

\$FINVAL(DIST,XCOORD,YCOORD,ZCOORD);

DIST : 移動距離.

XCOORD: 移動後のX-座標.

YCOORD: 移動後のY-座標.

ZCOORD: 移動後のZ-座標.

”\$FINVAL macro”

REPLACE {\$FINVAL(##,##,##);} WITH

{P2}=X(NP)+{P1}*U(NP); {P3}=Y(NP)+{P1}*V(NP);

{P4}=Z(NP)+{P1}*W(NP);}

”Note: Everywhere \$FINVAL is used one must include COMIN/STACK/”

\$SHOWNEAR マクロ

- **PRESTA** が現地地点から境界までの最短距離を必要とする.
- この距離は用いているジオメトリに依存する.
- ユーザーは**\$SHOWNEAR macro**内でこの距離を定義する.
- ucnai1p.mor, ucnai2p.mor, ucnai3p.mor と ucnai4p.mor はそれぞれのジオメトリに対応した \$SHOWNEAR マクロを持っている.
- CGでもSHOWNEARは自動的に計算される。

改訂記録

- 7JAN2003 ISIDEの説明を追加
- 29JUL2003 ジオメトリー構造の選択を追加