

EGS5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

28 JUL 2004

EGS5のジオメトリー

- EGS5ではジオメトリーの単位を **領域 (region)** と呼ぶ。
- EGS5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。
 - **SUBROUTINE HOWFAR** : 粒子が面を通過する事の判定。
面と領域の関連づけ。
- EGS4 ジオメトリーと EGS5ジオメトリー
 - Mortran を Fortran に変更
 - マクロ を Subroutine に変更
 - HOWNEARは廃止
 - ジオメトリー関連の変数名は変更せず

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. 面や円筒の数、半径、位置などを入力

2. Combinatorial Geometry

- 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 `cgview` (ジオメトリーチェッカー付き)
- iv. 上記「定型」に比べて約2-2.5倍遅い(2003年版CGに比べて最大5倍の高速化 by 杉田氏)

3. 非定型: 自作HOWFAR

- i. 自由度大 (プログラミングの作業が必要)
- ii. CGに比べて速い

この講演のストーリーとゴール

- 多重円筒平板用HOWFARと、そのためのMAINでの入力を記述して行く。
- 多重円筒平板用入力の理解
- HOWFARを自作する時のためのHOWFARの構造の理解

USTEP, IDISC, IRNEW

- HOWFARで重要なEGS5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** の時に廃棄領域 (discard region) 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 **廃棄領域 (discard region)** なら
 - **IDISC=1** を設定し、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;

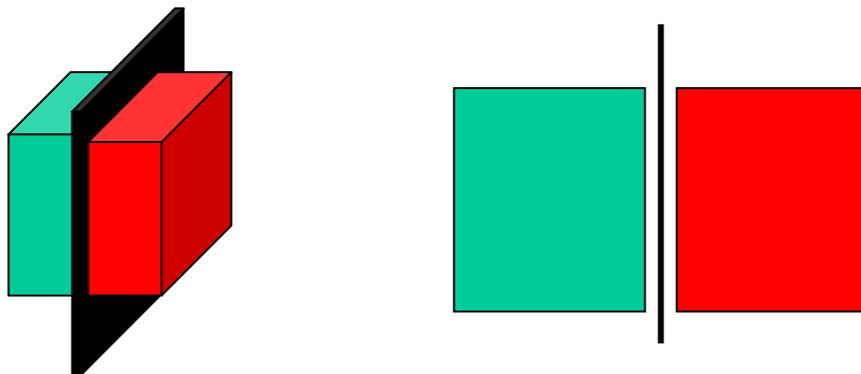
IRNEW=NEXTREG;

ここで、「;」はFortranの文区切り記号 (Mortranではない)
PPTの場所の節約のため使用している。

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。
- EGS5 システムにはいくつかのジオメトリ用サブルーチンが含まれている。
 - PLANE1, PLAN2P, PLAN2X
 - CYLNDR, CYL2
 - CONE, CONE2
 - SPHERE, SPH2
- HOWFARの機能を助ける二つのサブルーチン。
 - CHGTR : 必要な時に USTEP と IRNEW を書き換える。
 - 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

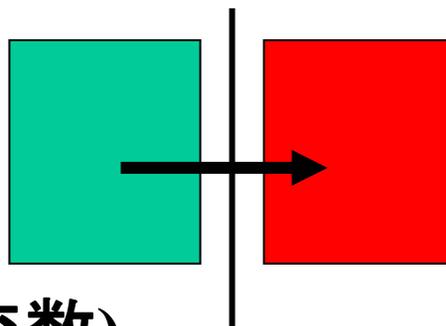


二分された空間を区別する

法線ベクトルの
出発側の空間:

ISIDE=1

($\$SPLANE1$ での変数)



法線ベクトルの
到着側の空間:

ISIDE= -1

Subroutine plane1 plane1(nplan, iside, ihit, tval)

nplan : 調べたい平面の番号

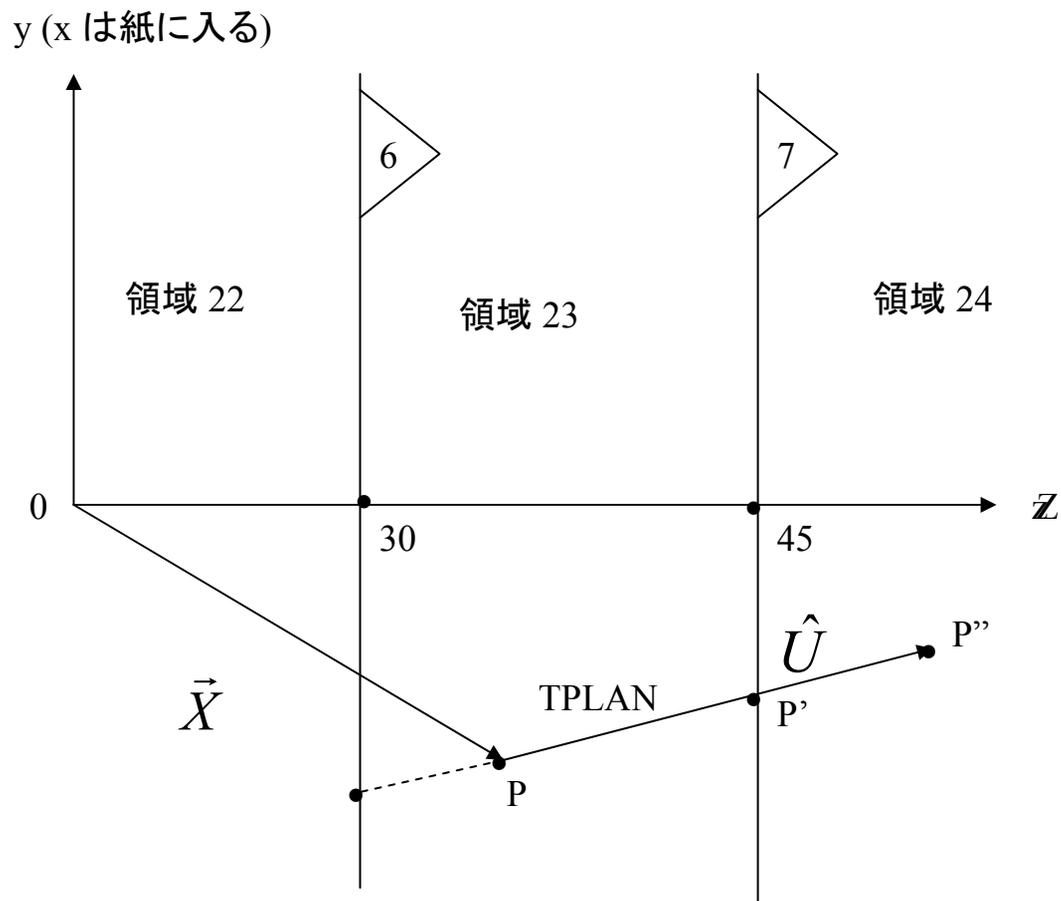
iside : 粒子が法線ベクトルの出発側にある場合は1を指定
: 粒子は法線ベクトルの到着側にある場合は-1を指定

ihit : 粒子の飛跡が平面と交差する場合、1が戻る
粒子の飛跡と平面は平行な場合、2が戻る
粒子の飛跡は平面から遠ざかる場合、0が戻る

tval : IHIT=1 の場合に、平面までの距離が戻る

- 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

include 'include/egs5_h.f' !Other includes are omitted

integer irl !Other declarations are omitted

IRL=IR(NP)

IF(IRL.NE.23)

IDISC=1 !Discard particles outside region 23

ELSE !Track particles within region 23

call PLANE1(7,1,IH IT,TPLAN) !Check upstream plane first

IF(IHIT.EQ.1) !Surface is hit --- make change if necessary

call CHGTR(TPLAN,24)

ELSEIF(IHIT.EQ.0) !Heading backwards

call PLANE1(6,-1,IHIT,TPLAN) !To get TPLAN-value (IHIT=1, must)

call CHGTR(TPLAN,22) !Make change if necessary

END IF

END IF

RETURN; END;

Subroutine CHGTR

- **Subroutine CHGTR(tvalp,irnewp)** で行うことは:
 - もし **TVALP.LE.USTEP** なら
 - **USTEP=tvalp** そして
 - **IRNEW=irnewp**
 - そうでない時には何もしない.

```
if (tvalp .le. ustep) then
```

```
    ustep = tvalp
```

```
    irnew = irnewp
```

```
end if
```

Subroutine FINVAL

- Subroutine FINVAL は粒子の飛行後の最終座標を求める.

Subroutine 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.

Subroutine PLAN2P

- ここまで見てきたHOWFARの例題は Subroutine **PLAN2P** によって、さらに単純になる

```
SUBROUTINE HOWFAR;
```

```
include 'include/egs5_epcont.f' !Other includes are omitted
```

```
integer irl !Other declarations are omitted
```

```
IRL=IR(NP)
```

```
IF(IRL.NE.23)
```

```
    IDISC=1 !Discard particles outside region 23
```

```
ELSE !Track particles within region23
```

```
    CALL PLAN2P(7,24,1,6,22,-1)
```

```
END IF
```

```
RETURN
```

```
END
```

PLAN2P の引数

Subroutine 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

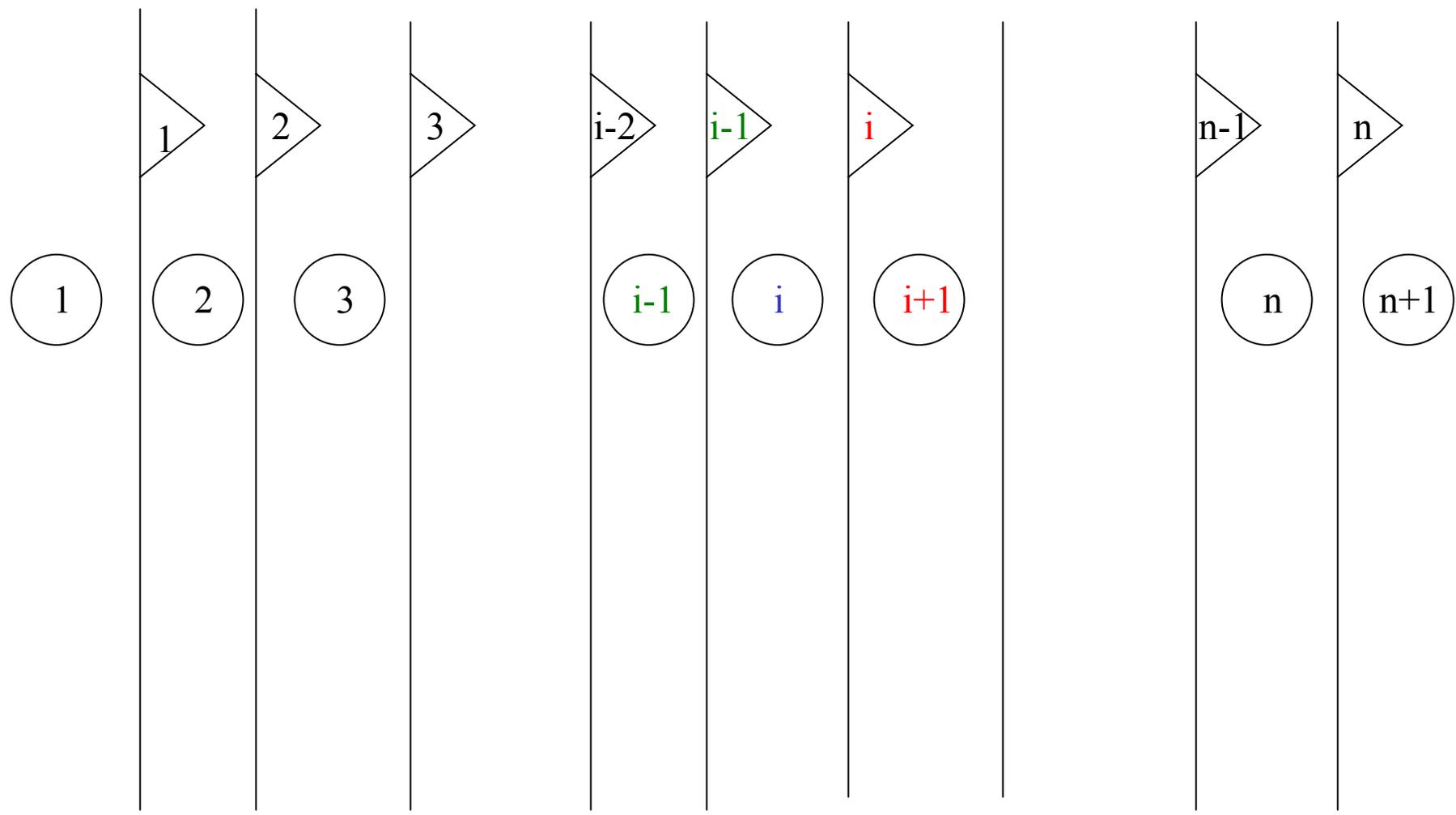
CHGTR(TPLAN,22) が続くことを意味している。

領域 I において: 前方の平面番号= i , 前方領域番号= $i+1$

$NPL1=IRL$; $NRG1=IRL+1$;

後方の平面番号= $i-1$, 後方の領域番号= $i-1$

$NPL2=IRL-1$; $NRG2=IRL-1$;



多重平板ジオメトリー

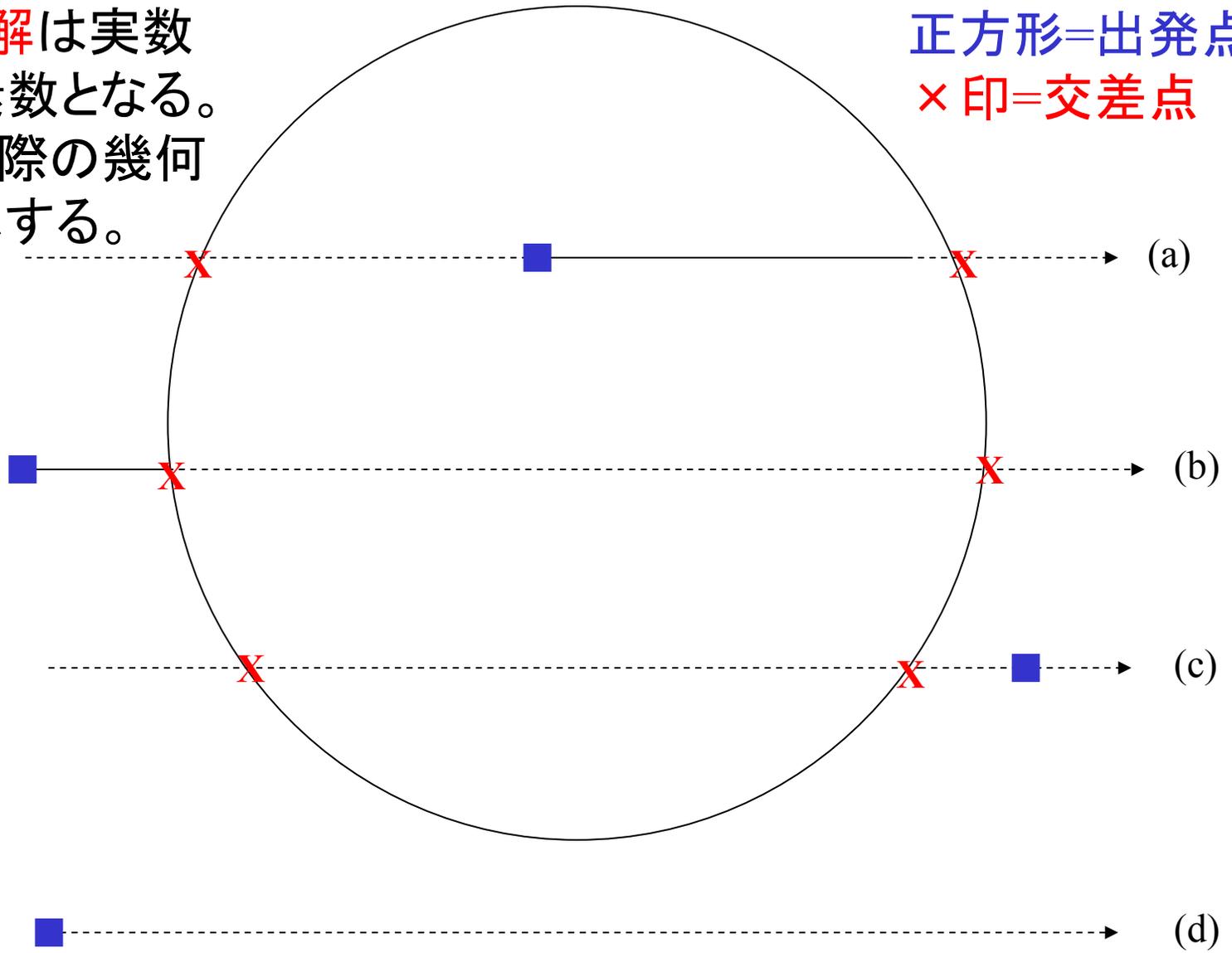
- ここまでの HOWFAR を多重平板へと簡単に拡張できる.

```
SUBROUTINE HOWFAR !Multi-slab  
include 'include/egs5_epcont.f' !See program for all include  
integer irl ! See program for all the declarations  
IRL=IR(NP); !Create a local variable  
IF(IRL.EQ.1.OR.IRL.EQ.NREG)  
  IDISC=1; !Upstream/downstream region  
ELSE  
  CALL PLAN2P(IRL,IRL+1,IRL-1,IRL-1,-1)  
  END IF  
RETURN  
END
```

•ベクトルと円筒表面
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時のパターン

正方形=出発点,
×印=交差点



- Subroutine CYLNDRはこれらのすべてのパターンに対応できる

Subroutine CYLNDR(ICYL,ISIDE,IHIT,TCYL)

ICYL : 調べる円筒のID 番号

ISIDE : 粒子が円筒の内部にある場合に、1と指定せよ
粒子が円筒の外部にある場合に、0と指定せよ

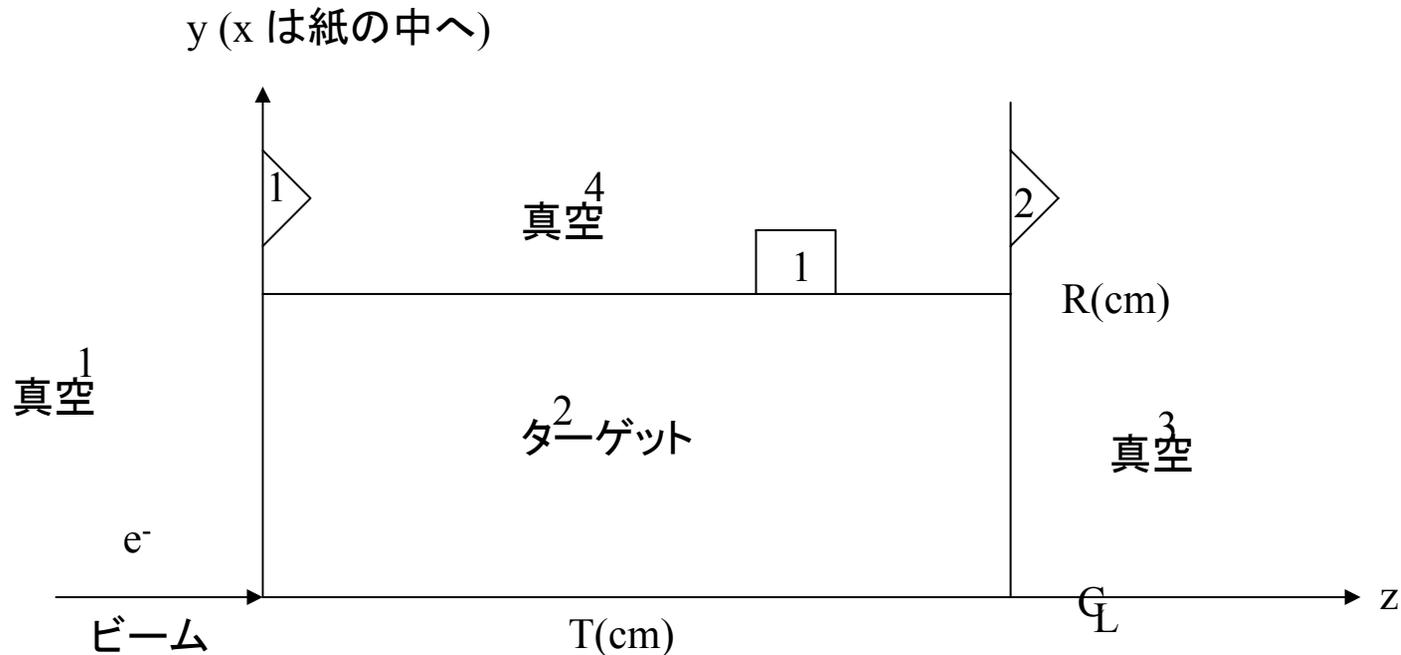
IHIT : 粒子が表面と交差する場合に、1が戻ってくる
粒子が表面と交差しない場合に、0が戻ってくる

TCYL : IHIT=1 の場合に、表面までの距離が戻ってくる

円錐表面および球表面アルゴリズムも円柱と基本的に同じ。
CONE とSPHERE はSUBROUTINE HOWFAR 内で
CYLNDR と同じように使える。

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

include 'user_auxcommons/cyldta' ! See program for all include files

integer irl ! See program for all declarations

IRL=IR(NP) !Create local variable

IF(IRL.NE.2)

IDISC=1 !Discard particles outside the target

ELSE !Track particle within the target

CALL CYLNDR(1,1,IHI,TCYL) !Check the cylinder surface

IF(IHIT.EQ.1)

CALL CHGTR(TCYL,4) ! Change if necessary

CALL PLAN2P(2,3,1,1,1,-1) !Check the down (and up) stream planes

END IF

RETURN

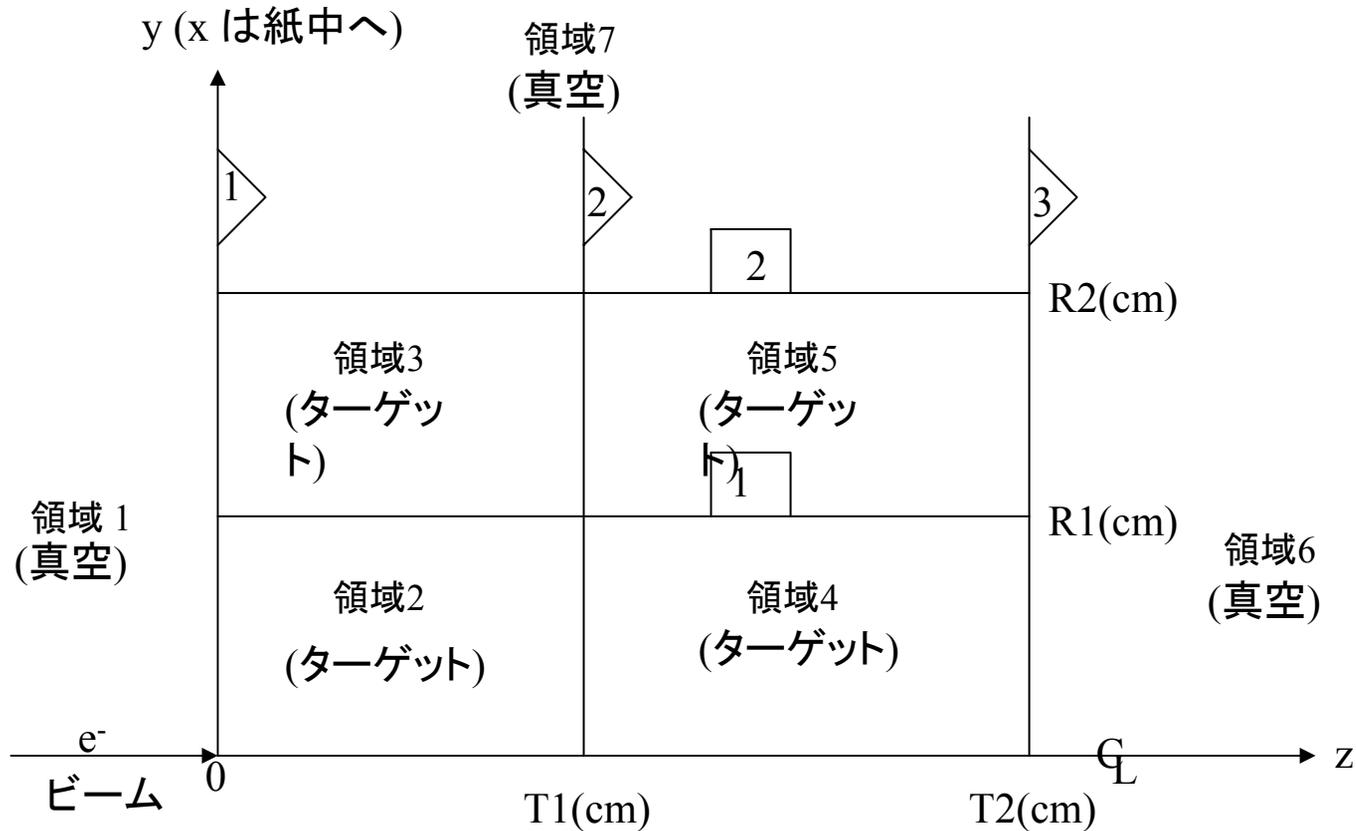
END

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.
- COMMON/CYLDTA/ を経由してこれらの量が HOWFAR に伝わる.
- 円筒の数の上限(**MXCYLS**)は **user_auxcommons/aux_h.f** で定義されており、随時変更可能.

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:



Subroutine CYL2: 2つの円筒の間の粒子を扱う

(このSubroutineは平行な2平面を扱うPLAN2Pに対応する)

Subroutine 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

include 'user_auxcommons/cyldta' ! See program for all include

integer irl !See program for all declarations

IRL=IR(NP) !Create local variable

IF(IRL.LE.1.OR.IRL.GE.IRZ+2)

IDISC=1

RETURN

END IF

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

END IF

```
IF(NSLAB.GT.1)
  NRG2=IRL-NCYL
ELSE
  NRG2=1
END IF
CALL PLAN2P(NPL1,NRG1,1,NPL2,NRG2,-1)
IF(NANNU.LT.NCYL)
  NRG2=IRL+1
ELSE
  NRG2=IRZ+3
END IF
IF(NANNU.GT.1)
  NRG1=IRL-1
  NCL2=NANNU
  NCL1=NANNU-1
  CALL CYL2(NCL1,NRG1,NCL2,NRG2)
  RETURN
END IF
CALL CYLNDR(1,1,IHIT,TCYL)
IF(IHIT.EQ.1) CALL CHGTR(TCYL,NRG2)
RETURN
END
```

- 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:
 - ucrz_nai3.f
 - NaI (Tl) 検出器のレスポンスを計算する.

円筒平板体系の入力のまとめ

- common/PLADTA/pcoord(3,MXPLNS),
pnorm(3,MXPLNS)
- common/CYLDTA/cyrad2(MXCYLS),cyr
ad(MXCYLS)
- common/GEORZ/ncyl,nplan,irz

NPLAN:# of planes

NCYL:# of cylinders

NREG=(NPLAN-1)*NCYL+3

IRZ=NREG-3

改訂記録

- 7JAN2003 ISIDEの説明を追加
- 29JUL2003 ジオメトリ構造の選択を追加
- 24JUL2004 egs5用に改訂