How to Write Geometry of EGS5

Y. Namito and H.Hirayama
KEK, High Energy Accelerator Research Organization

28 JUL 2004

Geometry in EGS5

- An EGS5 User Code requires:
  - **SUBROUTINE AUSGAB** for scoring results
  - **SUBROUTINE HOWFAR** to provide information to EGS5 about the nature of the geometry
  - In EGS5, the unit of geometry is called **“region”**. Material is assigned to each region.
- EGS4 geometry versus EGS5 geometry
  - Mortran is changed into fortran.
  - Macros are changed into subroutines.
  - No HOWNEAR is needed.
  - Geometry related variable names are not changed.

Selection of geometry structure

1. Regular（Multi cylinder, Multi slab, Voxel）
   i. HOWFAR is already written.
   ii. Input: Number and location of plane, cylinder.
2. Combinatorial Geometry
   i. HOWFAR is already written.
   ii. Input: Size and location of box, cylinder, sphere etc and combination of them.
   iii. Geometry display system cgview (w/geometry checker)
   iv. About 2-2.5 times slower than "Regular" above. (Up to 5 times faster than 2003 version CG by Mr. Sugita.)
3. Self-written HOWFAR
   i. Lot of freedom (Effort is needed for coding.)
   ii. Faster than CG

Story and goal of this talk

- Explanation of HOWFAR for multi cylinder and multi slab, input for it in MAIN.
- Understanding of input for this HOWFAR (Most busily used howfar)
- Understanding of structure of HOWFAR which is necessary when one write it.

USTEP, IDISC, IRNEW

- Three EGS5 variables that play important roles in HOWFAR.
- Available in **COMON/EPCONT/**.
- **USTEP:** Distance to the next position. It was set to the interaction point in the case of photons before calling HOWFAR. It was set to a current region (IR(NP)) before calling HOWFAR.
- **IDISC:** Flag to indicate a ***discard region*** when it is **1**.
- **IRNEW:** The region number when a particle moves a distance

Functions of HOWFAR

- If a current region is a **discard region**
  - Set **IDISC=1** and return
- Calculate a straight-line distance to the boundary (DIST).
  - If **DIST < USTEP,**
  - Shrunk **USTEP** to **DIST**
  - and set **IRNEW** to the region number that a particle will enter (**NEXTREG**).

USTEP=DIST;

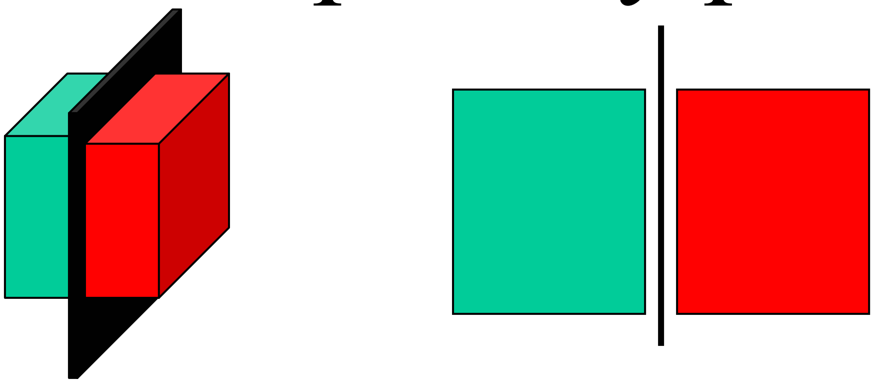
IRNEW=NEXTREG;

; is Fortran statement separator (NOT Mortran)
Used just here to save space in PPT

How to calculate the distance to the boundary

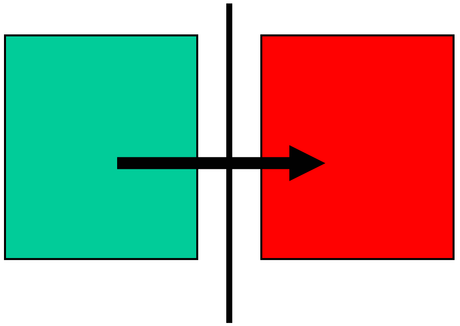
- User can use his/her own way to calculate the distance to the boundary.
- EGS5 system provides several subroutines.
  - **PLANE1, PLAN2P, PLAN2X**
  - **CYLNDR, CYL2**
  - **CONE, CONE2**
  - **SPHERE, SPH2**
- Two other geometry subroutines are available to help the function of HOWFAR.
  - **CHGTR** for changing USTEP and IRNEW if needed.
  - **FINVAL** for getting new coordinate at the and of any transport

Separate space by plane



Name the two spaces

Space where normal vector starts:
ISIDE=1
(Variable in PLANE1)



Space where normal vector reaches:
ISIDE= -1

Subroutine PLANE1(NPLAN,ISIDE,IHIT,TVAL)

NPLAN: ID number of plane to be checked

ISIDE : Specify 1 if region is between origin and outer normal

Specify -1 if region is not between origin and outer normal

IHIT : 1 is returned if particle trajectory will hit plane

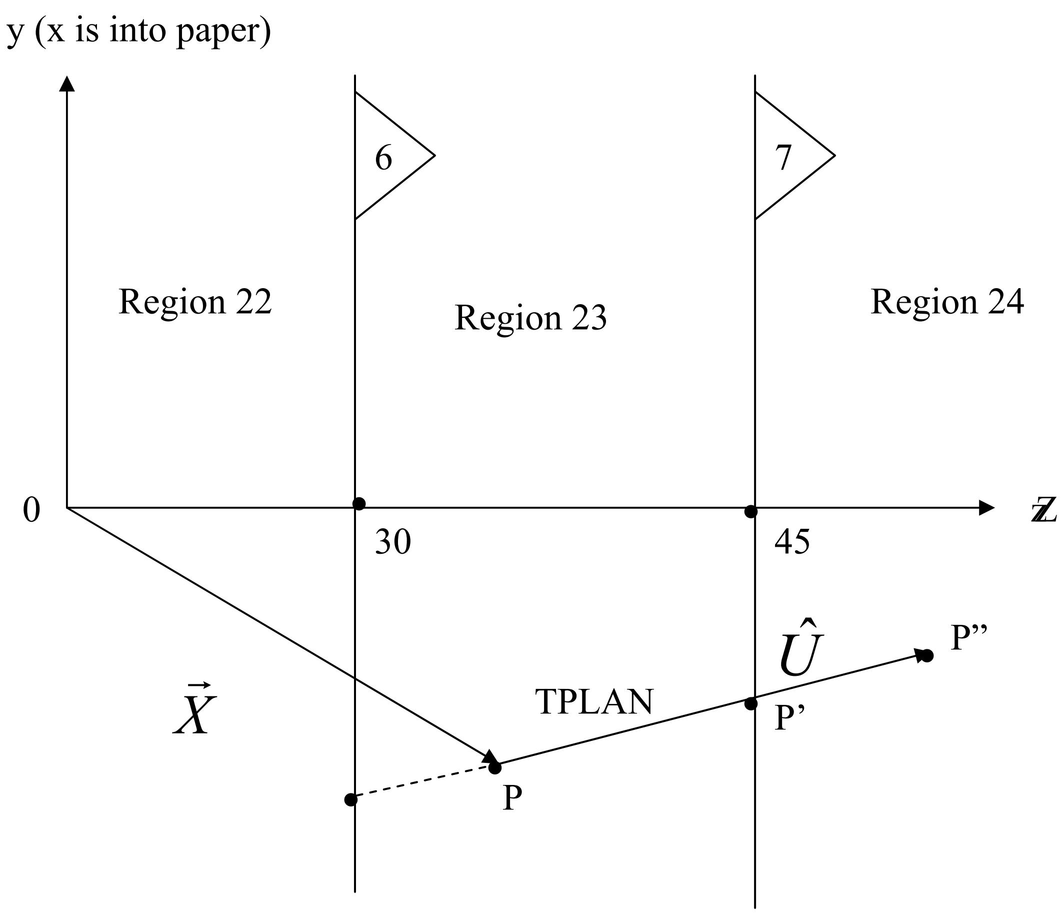
2 is returned if particle trajectory is parallel to plane

0 is returned if particle trajectory is away from plane

TVAL : Distance to plane is returned if IHIT=1

- The plane is defined by its normal vector **(PNORM(I,J),I=1,3)** and coordinates at the intersection point of the normal vector and the plane **(PCOORD(I,J),I=1,3)**.
- Both variables are available in **COMMON/PLADTA/**.

Consider two parallel planes separating three regions:



The regions are identified by number 22, 23 and 24 and the planes by number 6 and 7.

Definition of planes

- Assuming that planes 6 and 7 are located at z=30 cm and z=45cm, respectively.
- Planes 6 and 7 are defined as:

```
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;
```

Particles are initially started in region 23, and discarded when they leave this region.

```
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 region23
 call PLANE1(7,1,IHIT,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

- The subroutine **CHGTR(tvalp,irnewp)** does the following:
  - If **tvalp.le.ustep** then
    - **ustep=tvalp** and
    - **irnew=irnewp**
  - Otherwise, nothing is done.

```
if (tvalp .le. ustep) then
  ustep = tvalp
  irnew = irnewp
end if
```

Subroutine FINVAL

Subroutine FINVAL is useful for determining the final coordinates of a particle.

Subroutine FINVAL (DIST,XCOORD,YCOORD,ZCOORD)

DIST : the distance traveled.

XCOORD: X-coordinate after travel.

YCOORD: Y-coordinate after travel.

ZCOORD: Z-coordinate after travel.

Subroutine PLAN2P

- The HOWFAR example that we have been following can be simplified even further with the aide of **Subroutine PLAN2P**

```
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
```

Arguments of PLAN2P

Subroutine 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)

- The first group of numbers (NPL1,NRG1,ISIDE) (7,24,1) corresponding check the downstream plane is equivalent to

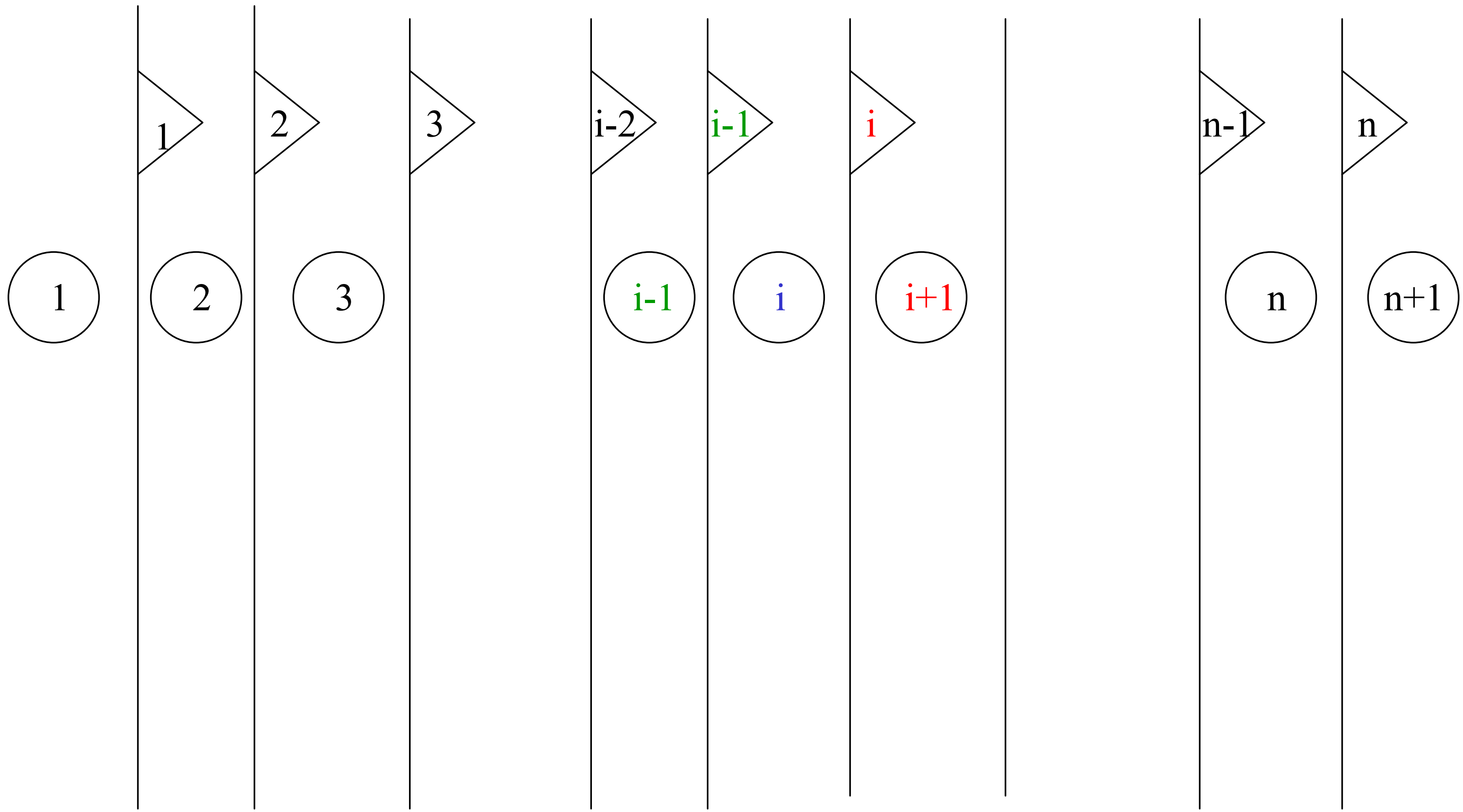
  PLANE1(7,1,IHIT,TPLAN) followed by **CHGTR(TPLAN,24).**

- The second group of numbers (NPL2,NRG2,ISIDE) (6,22,-1) corresponding check the downstream plane is equivalent to

  PLANE1(6,-1,IHIT,TPLAN) followed by **CHGTR(TPLAN,22).**

At region i: forward plane No=i, forward region No=i+1 **NPL1=IRL; NRG1=IRL+1;**

backward plane No=i-1, backward region No=i-1 **NPL2=IRL-1; NRG2=IRL-1;**



Multi-slab Geometry

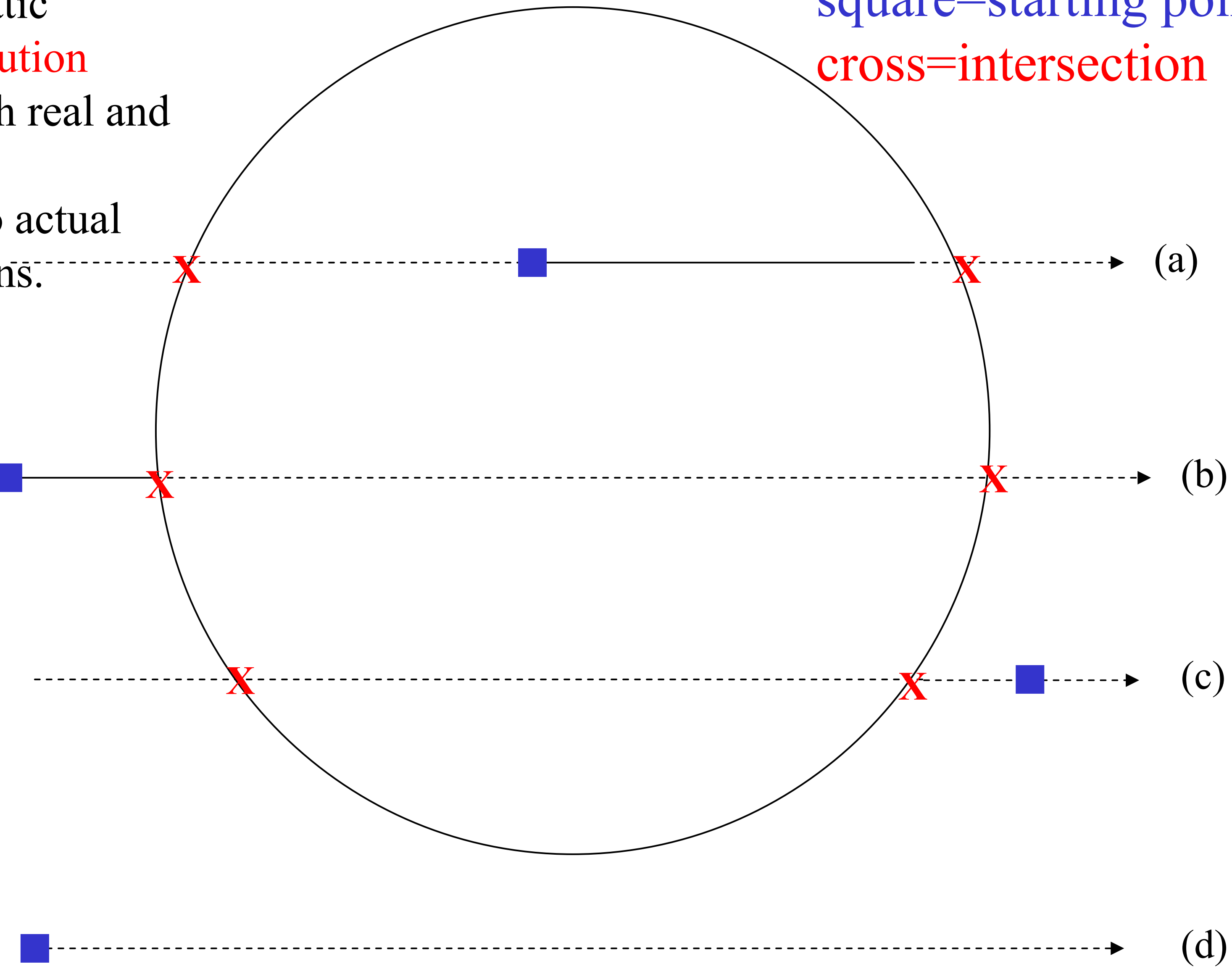
- It is simple to extend the previous HOWFAR for many slabs.

```
SUBROUTINE HOWFAR   !Multi-slab
include 'include/egs5_epcont.f'   ! See program for other include
integer irl   !See program for other 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
```

Possible trajectories intersection a cylinder

The intersection of a vector with a cylindrical surface leads to a quadratic equation, the solution of which are both real and imaginary and corresponding to actual physical situations.

square=starting point, cross=intersection



- Subroutine CYLNDR was designed to take all these possibilities into account.
- A cylinder which has the Z-axis as symmetry axis is;

Subroutine CYLINDR(ICYL,ISIDE,IHIT,TCYL)

ICYL : ID number of cylinder to be checked

ISIDE : Specify as 1 when particle is inside cylinder

Specify as 0 when particle is outside cylinder

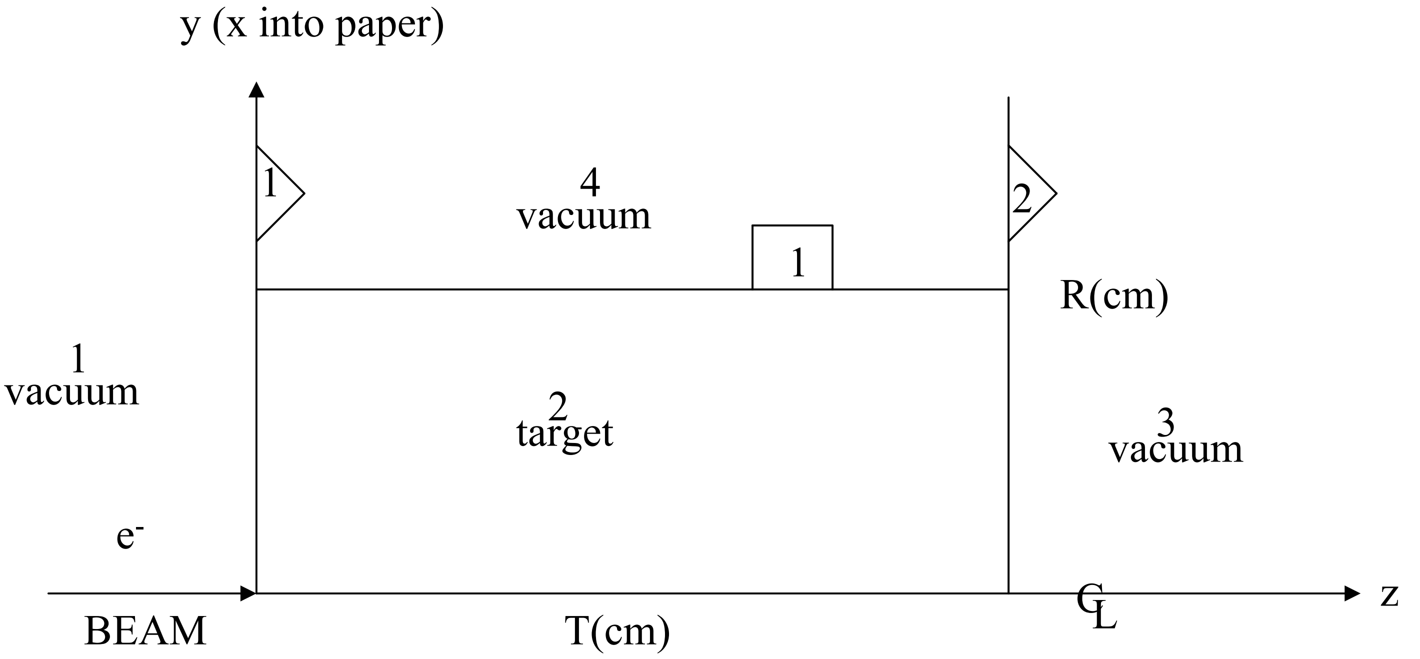
IHIT : 1 is returned if particle intersects surface

0 is returned if means particle misses surface

TCYL : Distance to surface is returned if IHIT=1

The conic surface algorithms basically are all the same.
CONE and SPHERE may be used in SUBROUTINE HOWFAR in the same manner as CYLNDR.

Example of cylindrical target



- The cylinder of rotation about z-axis is defined by box 1.
- There are four regions of interest – the target (region 2) and three vacuum regions upstream, downstream and surrounding the target.

•The following HOWFAR will work for this geometry.

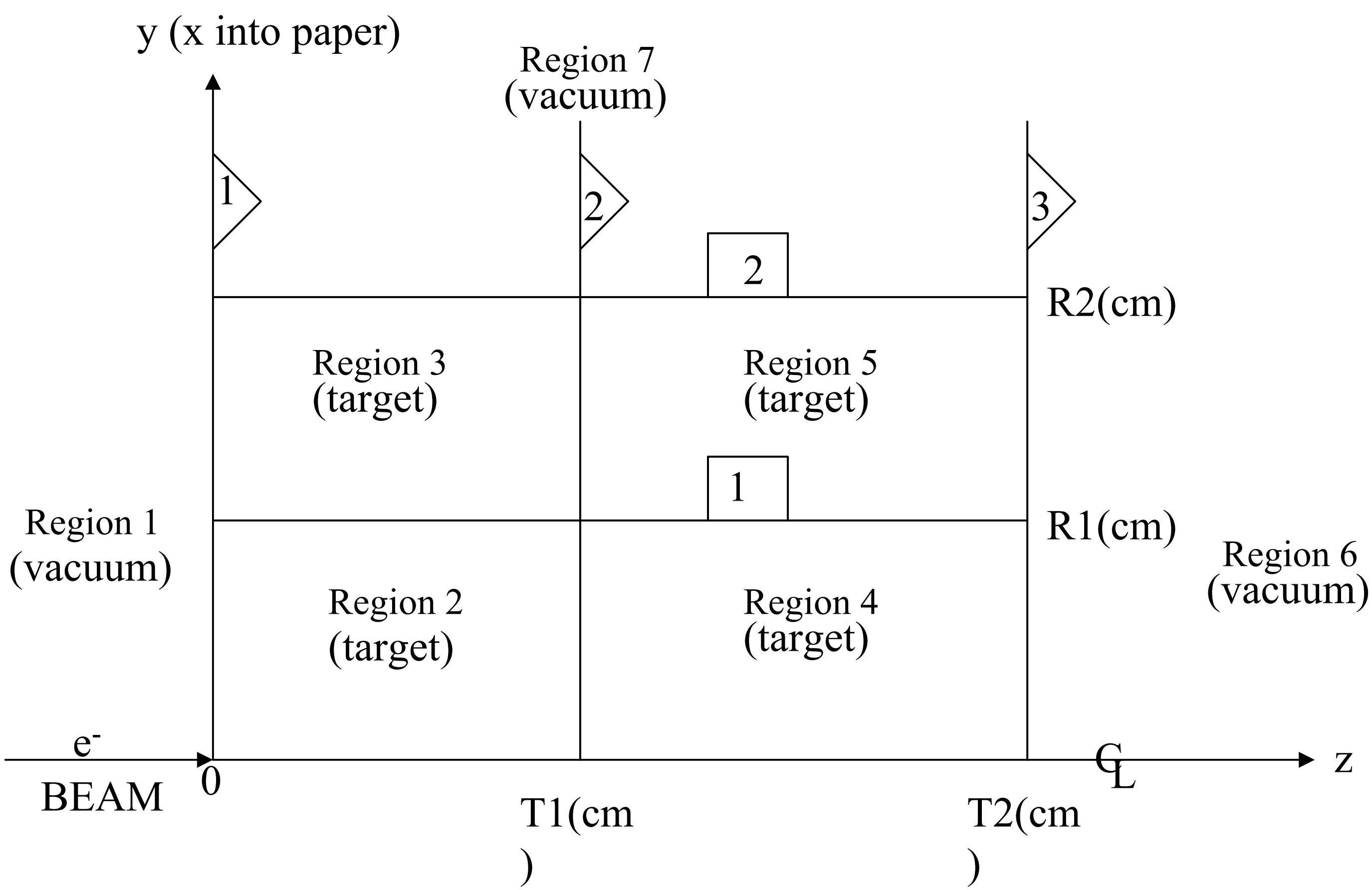
```
SUBROUTINE HOWFAR;
include 'user_auxcommons/cyldta.f' ! See program for all include files
integer irl  !See program for all declarations
IRL=IR(NP)  !Create local variable
IF(IRL.NE.2)
 IDISC=1  ! Discard particles outside the target
ELSE  !Track particle  within the target
 call CYLNDR(1,1,IHI,TCYL) !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 downstream (and upstream) planes
   END IF
RETURN
END
```

Definition of cylinder radius

- The radius of the cylinder (**CYRAD**) and its square (**CYRAD2**) must be defined in MAIN.
- These quantities are passed to HOWFAR via COMMON/CYLDTA/.
- Maximum number of cylinder (**MXCYLS**) is defined in **user_auxcommons/ aux_h.f** and can be re-defined.

Multi Cylinders and Slabs Example

- Consider a case that include 2 cylinders and 3 slabs:



Subroutine CYL2 : Treats particles between 2 cylinders

This subroutine corresponding to PLAN2P for the two parallel planes.

Subroutine 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.

```
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
```

- This HOWFAR can be used for a geometry having any number of cylinders and slabs.
- Sample user code of multi cylinders and slabs:
  - ucrz_nai.f
  - This user code also simulates NaI(Tl) detector response.

Summary of input for multi-cylinder multi-slab geometry

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

NPLAN:# of planes
NCYL:# of cylinders
NREG=(NPLAN-1)*NCYL+3;
IRZ=NREG-3;