



Geometry – Basic

Introduction to the combinatorial geometry package in FLUKA

Introduction

- Principle of Combinatorial Geometry (CG)

- Basic objects called **bodies** (such as cylinders, spheres, parallelepipeds, etc.) are combined to form more complex objects called **regions**
- This combination is done using **Boolean operations**

Operation	Math	FLUKA
Union	\cup	
Intersection	\cap	+
Subtraction	-	-

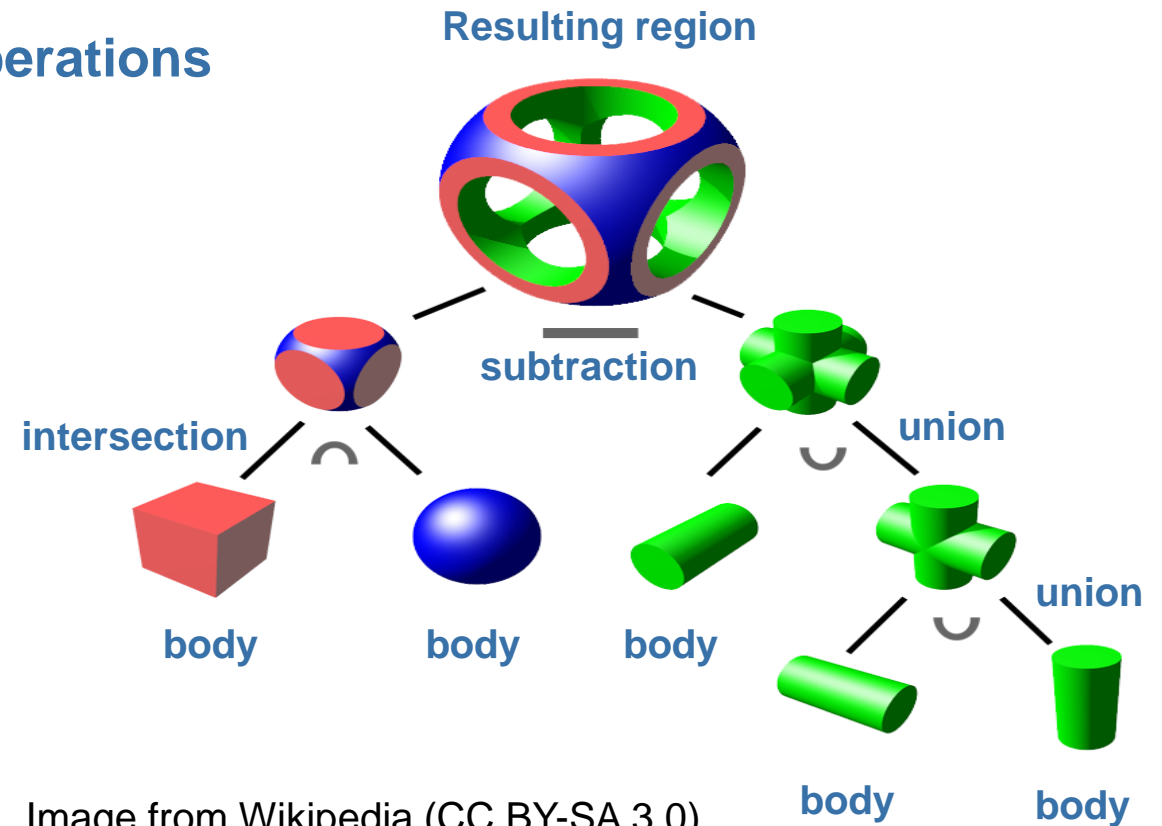
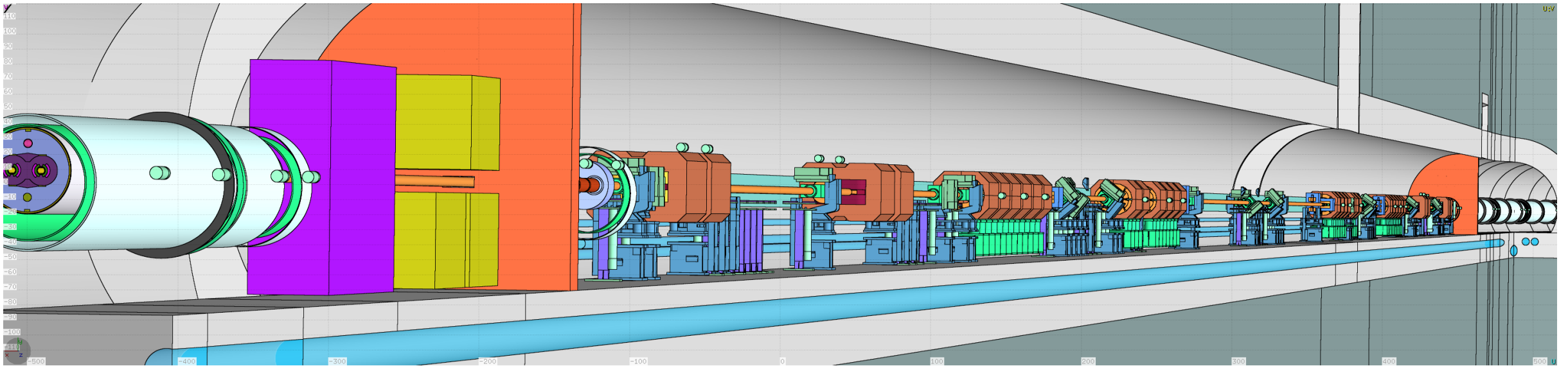


Image from Wikipedia (CC BY-SA 3.0)

Introduction

- Combinatorial Geometry in FLUKA

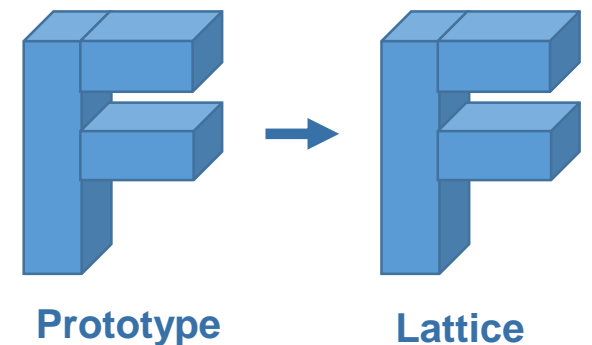
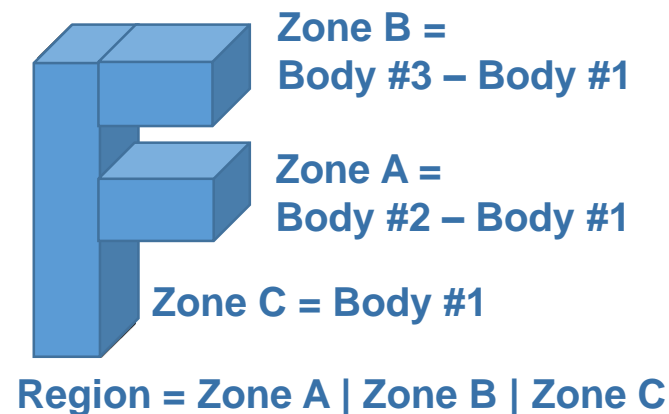
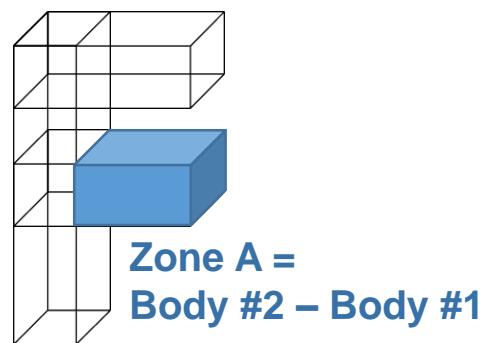
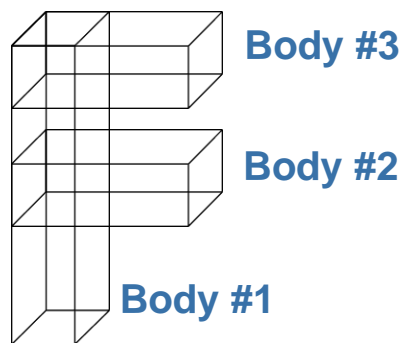
- Originally based on the geometry package for the neutron and gamma-ray transport program Morse developed at ORNL (M.B. Emmet ORNL-4972, 1975).
- Many improvements have been incorporated over time, for example **roto-translations** of bodies, **repetition of patterns**, or **voxels**



Example: LHC collimation insertion region built from the combination of bodies like **planes**, **cylinders**, **cuboids**, ...

Basic concepts and terminology

- Four concepts are fundamental in the FLUKA CG
 - **Bodies**: basic geometrical objects which are placed somewhere in the coordinate system
 - These include: **basic convex objects**, plus **infinite planes** (half-spaces), **infinite cylinders** (circular and elliptical), and **generic quadric surfaces** (surfaces described by 2nd degree equations)
 - **Zones**: sub-regions defined only via the intersection and subtraction of bodies
 - **Regions**: union of zones
 - In the simplest case a region consists of a single zone (no union operation needed in this case ...)
 - **Lattices**: duplication of existing objects (regions) confined in a container volume
 - Container can be translated and rotated, multiple duplications are possible



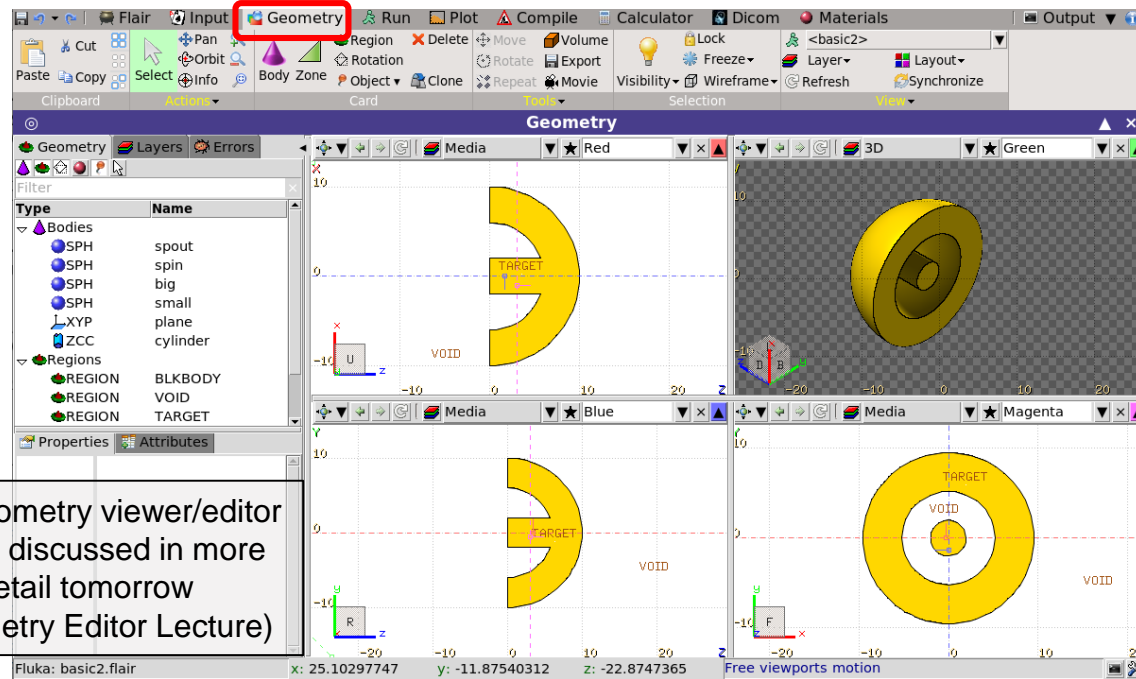
Geometry input

How to structure the geometry definition

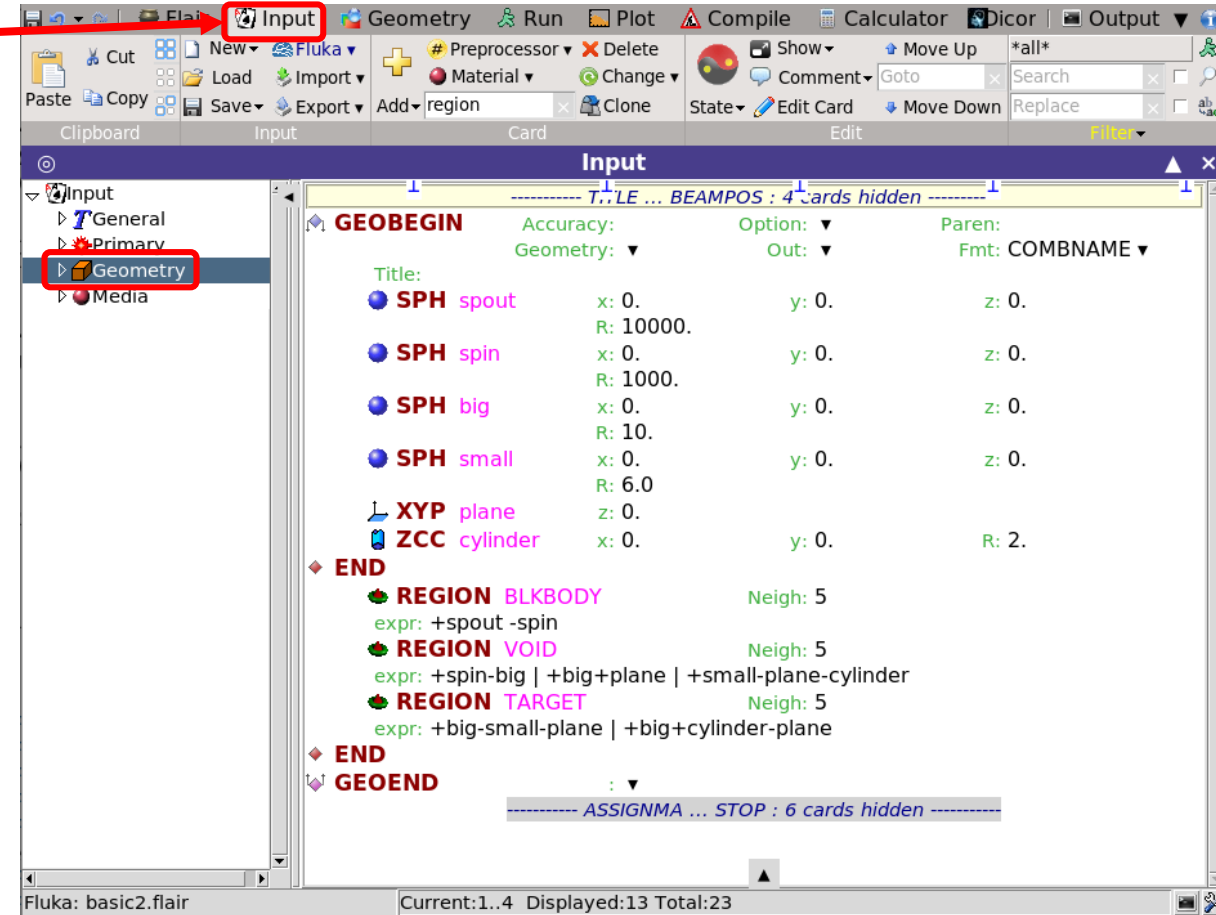
Geometry input in Flair

The geometry input (bodies, zones/regions, lattices) is defined under the “Geometry” branch of the **Input** tab

The resulting geometry can be viewed under the **Geometry** tab



The geometry viewer/editor will be discussed in more detail tomorrow (Geometry Editor Lecture)



Structure of the geometry input

The geometry definition must be enclosed by the **GEOBEGIN** and **GEOEND** cards

```

GEOBEGIN Accuracy: Option: Paren:
          Geometry: Out: Fmt: COMBNAME
Title: Gold target

SPH spout x: 0. y: 0. z: 0.
          R: 2000.
SPH spin x: 0. y: 0. z: 0.
          R: 1000.
SPH big x: 0. y: 0. z: 0.
         R: 10.
SPH small x: 0. y: 0. z: 0.
           R: 6.
XYP plane z: 0.
ZCC cylinder x: 0. y: 0. R: 2.

END
Black hole
REGION BLKBODY Neigh: 5
  expr: +spout -spin
Void around
REGION VOID Neigh: 5
  expr: +spin -big | +big +plane | +small -plane -cylinder
Target
REGION TARGET Neigh: 5
  expr: +big -small -plane | +big +cylinder -plane

END
GEOEND
  
```

Bodies are defined right after the **GEOBEGIN** card; the body definition must be terminated by an **END** card

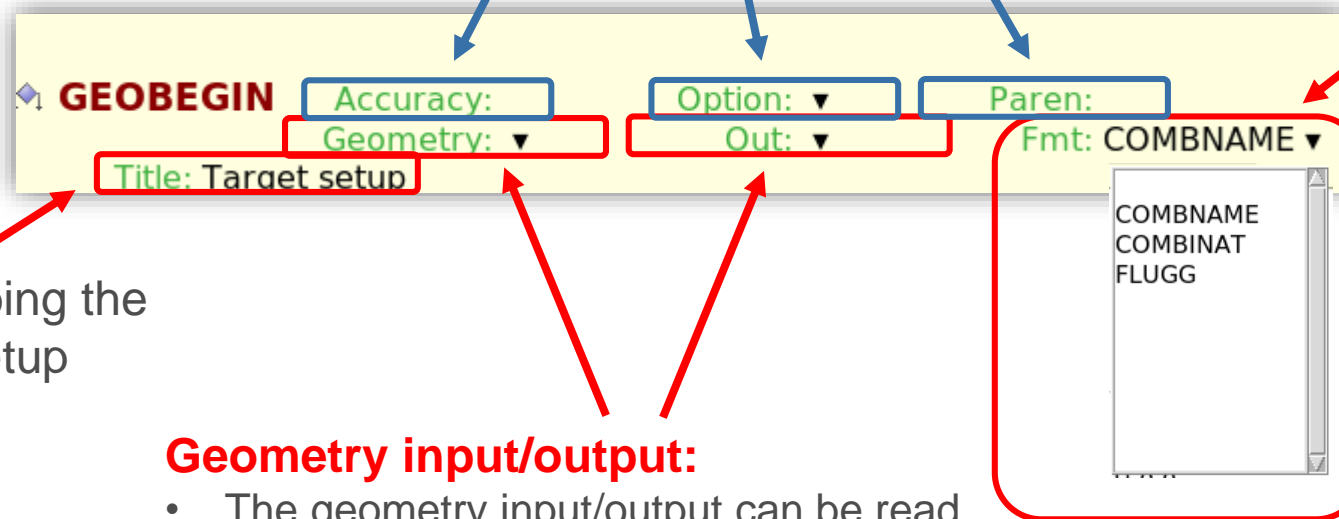
Regions (zones) are defined after the bodies; the region definition must be terminated by an **END** card

Lattices would be defined after the regions, right before the **GEOEND** card (not covered in this lecture)

Note: material definition/assignments and transformations are defined outside of GEOBEGIN→GEOEND

The GEOBEGIN card

Will be explained later in this lecture or can be found in backup



Title describing the geometry setup

Geometry input/output:

- The geometry input/output can be read from/written to another file.
- Here the logical I/O units can be chosen
- By default, output is writing to standard .out file

Geometry input format:

This option allows choosing how the geometry description is formatted.

Default is **COMBNAM**, which selects an obsolete formatting style (e.g. bodies and regions are identified via numbers) - **no longer supported by Flair!**

Choose **COMBNAM**, which is the name-based format and also features other improvements compared to **COMBNAT**.

Bodies

The basic objects of the FLUKA CG

Body definitions

- A body definition in FLUKA consists of:
 - A **3-letter code** indicating the **body type**
 - A **unique body name** (alphanumeric identifier, 8 character maximum, case sensitive)
 - A **set of geometrical quantities** defining the body, e.g. the body dimensions and the position in the coordinate system (the numbers depend on the body type, see next slides)

```
GEOBEGIN Accuracy: Option: Paren:
Geometry: Out: Fmt: COMBNAME
Title: Gold target
• SPH spout x: 0. y: 0. z: 0.
R: 2000.
• SPH spin x: 0. y: 0. Bodies
R: 1000.
• SPH big x: 0. y: 0. z: 0.
R: 10.
• SPH small x: 0. y: 0. z: 0.
R: 6.
XYP plane z: 0.
ZCC cylinder x: 0. y: 0. R: 2.
END
Black hole
• REGION BLKBODY Neigh: 5
expr: +spout -spin
Void around
• REGION VOID Neigh: 5
expr: +spin -big | +big +plane | +small -plane -cylinder
Target
• REGION TARGET Neigh: 5
expr: +big -small -plane | +big +cylinder -plane
END
GEOEND
```

Body type
(3-letter code)

Unique body
name

Geometrical
definition

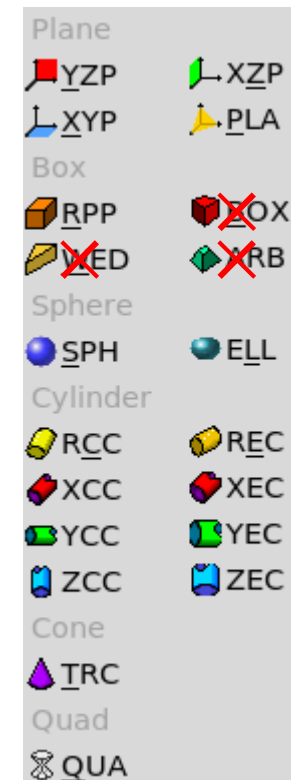
```
• RPP target Xmin: -5.0 Xmax: +5.0
Ymin: -5.0 Ymax: +5.0
Zmin: -5.0 Zmax: +5.0
```

**Note: all values
are in cm!**

List of bodies

- Following bodies are presently available in FLUKA:
 - **Planes**
 - **XYP, XZP, YZP**: Infinite half space delimited by a coordinate plane
 - **PLA**: Generic infinite half-space, delimited by a **PLA**ne
 - **Boxes**
 - **RPP**: **R**ectangular **P**arallele**P**iped
 - **Sphere and spheroid**
 - **SPH**: **SPH**ere
 - **ELL**: **ELL**ipsoid of revolution
 - **Cylinders and cones**
 - **XCC, YCC, ZCC**: Infinite **C**ircular **C**ylinder, parallel to coordinate axis
 - **RCC**: **R**ight **C**ircular **C**ylinder
 - **XEC, YEC, ZEC**: Infinite **E**lliptical **C**ylinder, parallel to coordinate axis
 - **REC**: **R**ight **E**lliptical **C**ylinder
 - **TRC**: **T**runcated **R**ight angle **C**one
 - **Other**
 - **QUA**: **QUA**dric

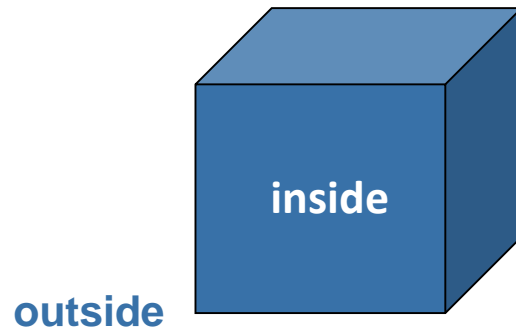
Note: BOX, WED/RAW, ARB are deprecated. Do not use them, because they can cause rounding problems.



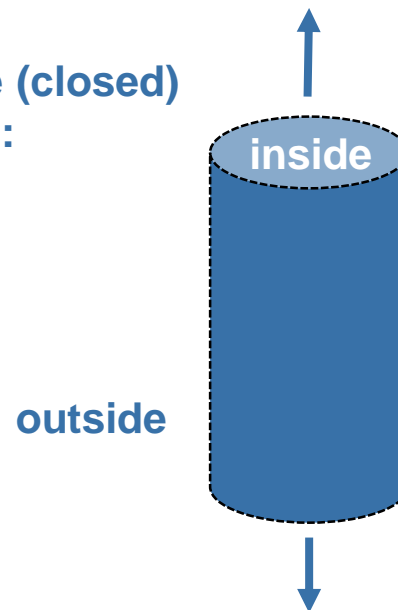
Division of space

- Each body splits the space into two domains: **inside** and **outside**
 - This concept will be later used when defining zones and regions
 - **+body** refers to the volume **inside** of the body
 - **-body** refers to the volume **outside** of the body
 - The concept of inside and outside is applied to all bodies including infinite planes

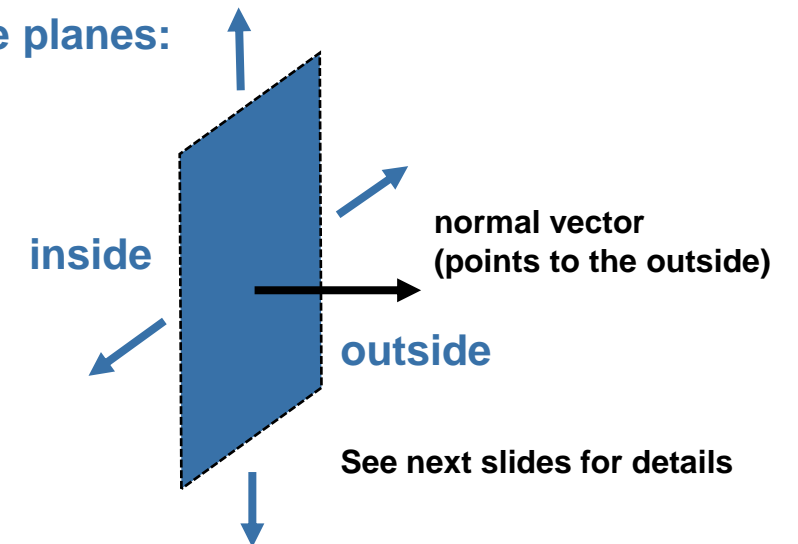
Finite bodies:



Infinite (closed) bodies:



Infinite planes:






Infinite planes parallel to coordinate axes

- Divide space in two half spaces parallel to coordinate axes

- **Three possibilities:**

- Plane \perp to the x-axis \rightarrow 3-letter code: **YZP**
- Plane \perp to the y-axis \rightarrow 3-letter code: **XZP**
- Plane \perp to the z-axis \rightarrow 3-letter code: **XYP**

In Flair:

 YZP plane1	x: 50.0
 XZP plane2	y: 100.0
 XYP plane3	z: -20.0

(note, in Flair V_x , V_y , V_z is labelled x, y, z)

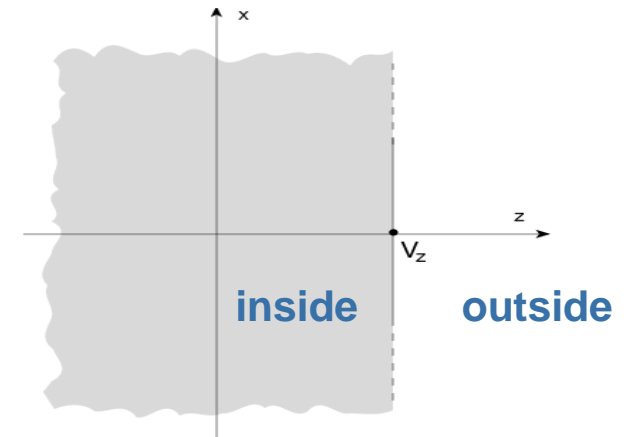
- **All defined by a single number** (coordinate of the plane on the perpendicular axis):

- x-coordinate V_x for YZP
- y-coordinate V_y for XZP
- z-coordinate V_z for XYP

- **Inside vs outside:**

- Points for which $x < V_x$ (resp. $y < V_y$, or $z < V_z$) are “inside the body”
- Points for which $x > V_x$ (resp. $y > V_y$, or $z > V_z$) are “outside the body”

Example XYP:



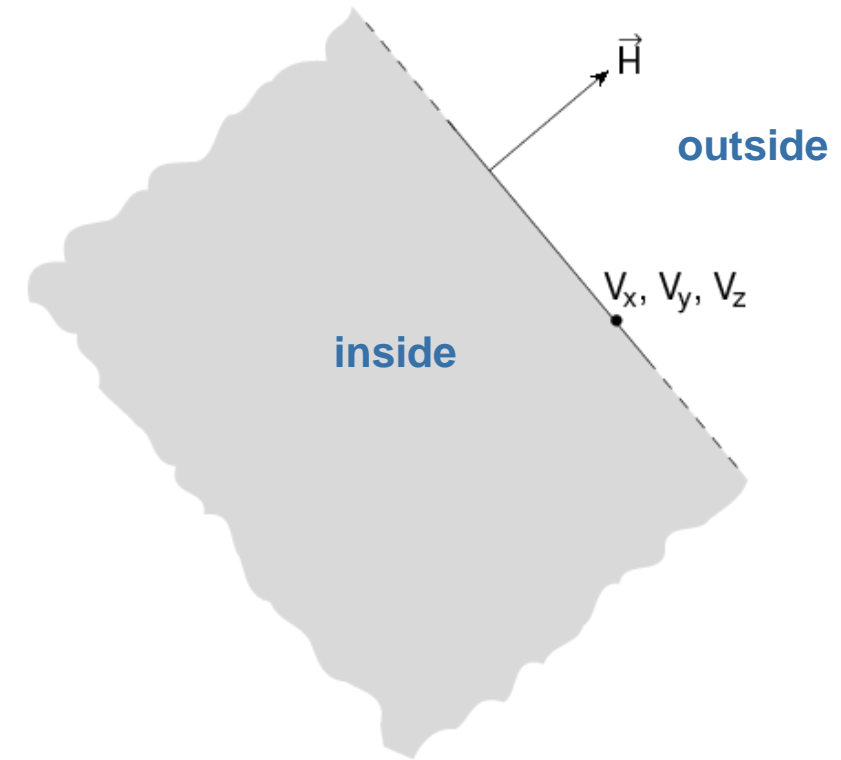
Arbitrarily orientated infinite plane

- Divides space in two arbitrarily oriented half spaces
 - 3-letter code: **PLA**
 - A PLA is defined by 6 numbers:
 - H_x, H_y, H_z (vector \perp to the plane, arbitrary length)
 - V_x, V_y, V_z (any point lying on the plane)
 - **Inside vs outside:**
 - The half-space “inside the body” is that from which the vector H_x, H_y, H_z is pointing (i.e. the vector points "outside").

In Flair:

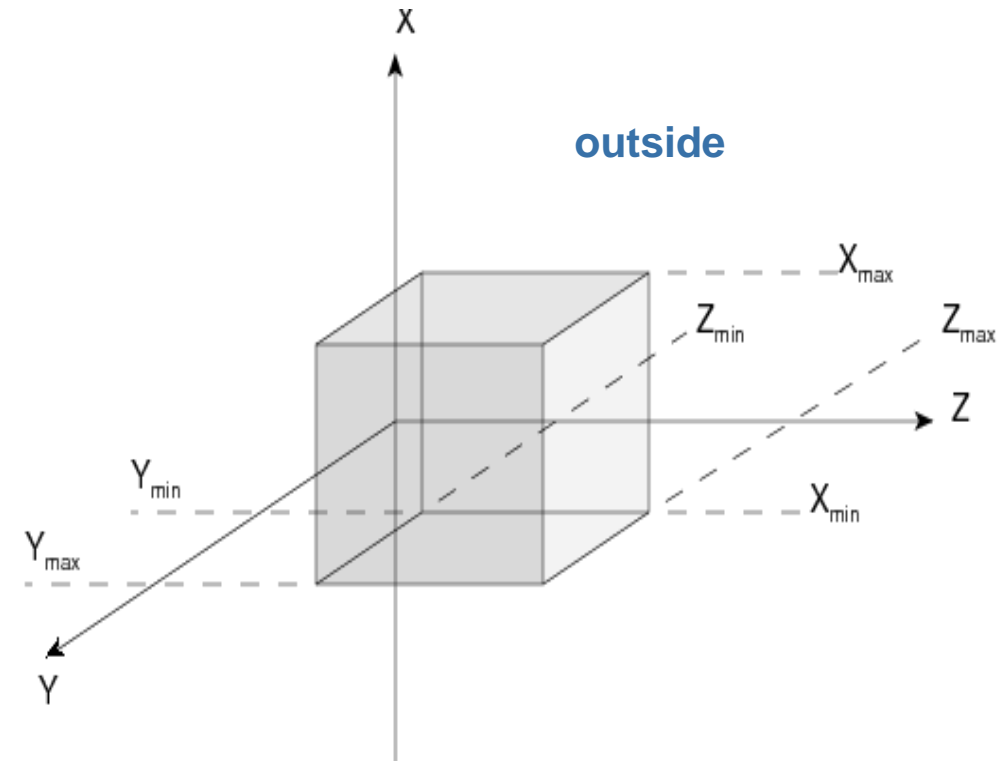
(note, in Flair H_x, H_y, H_z is labelled N_x, N_y, N_z)

▲ PLA plane4	Nx: 1.0	Ny: 5.0	Nz: 4.0
	x: 0.0	y: 3.0	z: 2.0




Rectangular parallelepiped

- Box with faces parallel to the coordinate axis
 - 3-letter code: **RPP**
 - A RPP is defined by 6 numbers:
 - X_{\min} , X_{\max} (positions of the two ZY faces of the RPP)
 - Y_{\min} , Y_{\max} (positions of the two XZ faces of the RPP)
 - Z_{\min} , Z_{\max} (positions of the two XY faces of the RPP)
 - **Inside vs outside:**
 - A point x , y , z is “inside the body” if it is contained in the RPP (i.e. $X_{\min} < x < X_{\max}$ and $Y_{\min} < y < Y_{\max}$ and $Z_{\min} < z < Z_{\max}$)



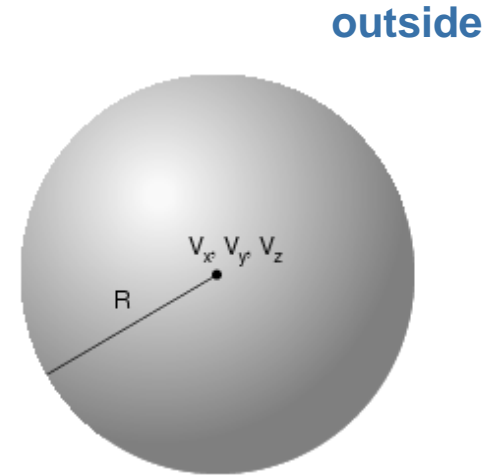
In Flair:

Do not use the deprecated body BOX

 RPP target	Xmin: -5.0	Xmax: +5.0
	Ymin: -5.0	Ymax: +5.0
	Zmin: -5.0	Zmax: +5.0

Sphere

- Sphere at arbitrary location
 - 3-letter code: **SPH**
 - A SPH is defined by 4 numbers:
 - V_x , V_y , V_z (centre of the sphere)
 - R (radius of the sphere)
 - **Inside vs outside:**
 - A point x , y , z is defined as “inside the body” if it is contained inside the sphere



In Flair:

(note, in Flair V_x , V_y , V_z is labelled x , y , z)

```
● SPH target1 x: 5.0 y: 3.0 z: 8.0
R: 4.0
```

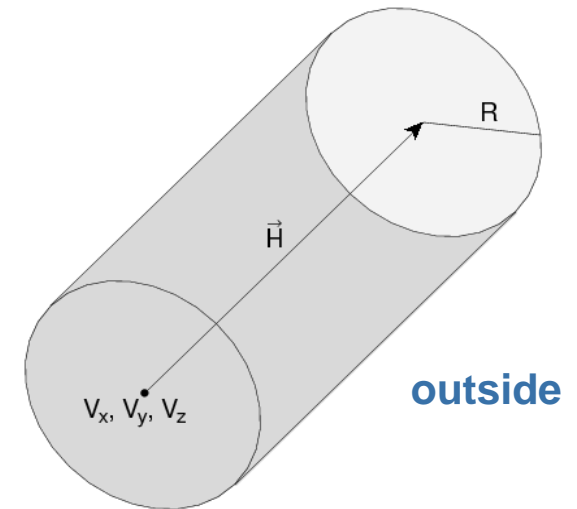

Right circular cylinder

- Finite cylinder at arbitrary position and with arbitrary orientation, limited by a cylindrical surface and two plane faces \perp to its axis
 - **3-letter code: RCC**
 - **A RCC is defined by 7 numbers:**
 - V_x, V_y, V_z (base of the cylinder, i.e., center of one cylinder face)
 - H_x, H_y, H_z (vector corresponding to the cylinder height, pointing to the center of the other face)
 - R (radius of the cylinder)
 - **Inside vs outside:**
 - A point x, y, z is defined as “inside the body” if it is contained inside the cylinder

In Flair:

(note, in Flair V_x, V_y, V_z is labelled x, y, z)

```
🔧 RCC target2  x: 5.0      y: 2.0      z: 8.0
                Hx: 3.0      Hy: 3.0      Hz: 0.0
                R: 2.0
```






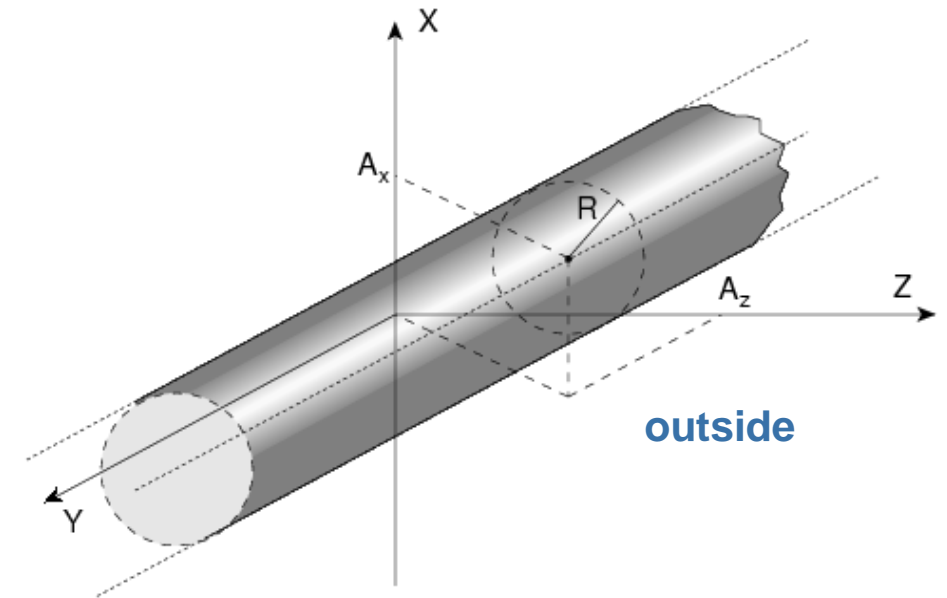
Infinite circular cylinders

- Infinite circular cylinder parallel to one of the coordinate axis, at arbitrary position
 - **Three possibilities:**
 - Cylinder parallel to the x-axis → 3-letter code: **XCC**
 - Cylinder parallel to the y-axis → 3-letter code: **YCC**
 - Cylinder parallel to the z-axis → 3-letter code: **ZCC**
 - **An XCC/YCC/ZCC is defined by 3 numbers:**
 - A_y, A_z for XCC, A_x, A_z for YCC, A_x, A_y for ZCC (coordinates of the cylinder axis)
 - R (radius of the cylinder)
 - **Inside vs outside:**
 - A point x, y, z is defined as “inside the body” if it is contained inside the cylinder

In Flair:

(note, in Flair A_x, A_y, A_z is labelled x, y, z)




 XCC cyl1	$y: 4.0$	$z: 3.0$	$R: 1.0$
 YCC cyl2	$z: 2.0$	$y: -5.0$	$R: 2.0$
 ZCC cyl3	$x: -3.0$	$y: 2.0$	$R: 5.0$

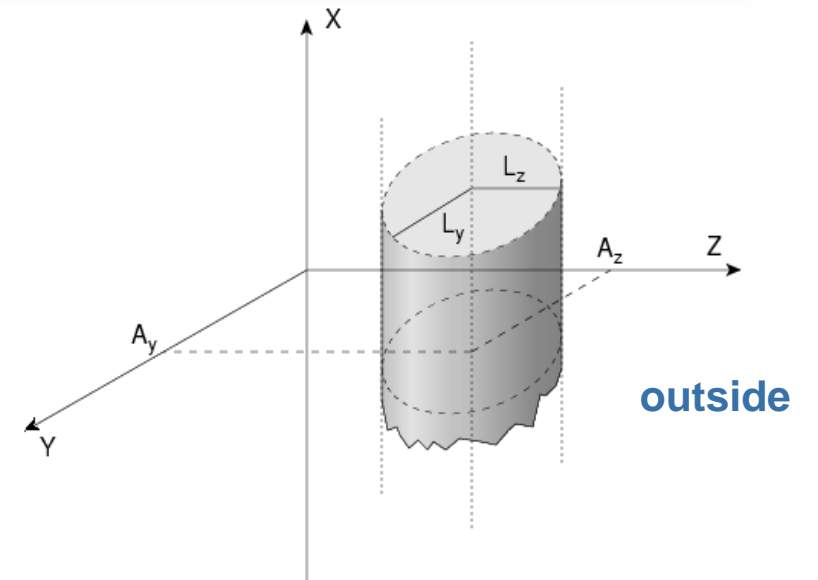


Infinite elliptical cylinders

- Infinite elliptical cylinder parallel to one of the coordinate axis, at arbitrary position
 - **Three possibilities:**
 - Cylinder parallel to the x-axis → 3-letter code: **XEC**
 - Cylinder parallel to the y-axis → 3-letter code: **YEC**
 - Cylinder parallel to the z-axis → 3-letter code: **ZEC**
 - **An XCC/YCC/ZCC is defined by 4 numbers:**
 - A_y, A_z for XCC, A_x, A_z for YCC, A_x, A_y for ZCC (coordinates of the cylinder axis)
 - L_y, L_z for XCC, L_x, L_z for YCC, L_x, L_y for ZCC (semi-axes of the cylinder)
 - **Inside vs outside:**
 - A point x, y, z is defined as “inside the body” if it is contained inside the cylinder

In Flair: (note, in Flair A_x, A_y, A_z is labelled x, y, z)

 XEC cyl4	y: 3.0 Ly: 2.0	z: 2.0 Lz: 5.0
 YEC cyl5	z: 1.0 Lz: 3.0	x: 0.0 Lx: 1.0
 ZEC cyl6	x: 4.0 Lx: 0.5	y: -2.0 Ly: 1.0



Arbitrary generic quadric

- A quadric surface defined by a 2nd degree equation $F(x,y,z) = 0$
 - 3-letter code: **QUA**
 - Each QUA is defined by 10 numbers:

$$C_{xx}, C_{yy}, C_{zz}, C_{xy}, C_{xz}, C_{yz}, C_x, C_y, C_z, C_0$$

- Corresponding to the equation:

$$C_{xx} x^2 + C_{yy} y^2 + C_{zz} z^2 + C_{xy} xy + C_{xz} xz + C_{yz} yz + C_x x + C_y y + C_z z + C_0 = 0$$

In Flair:

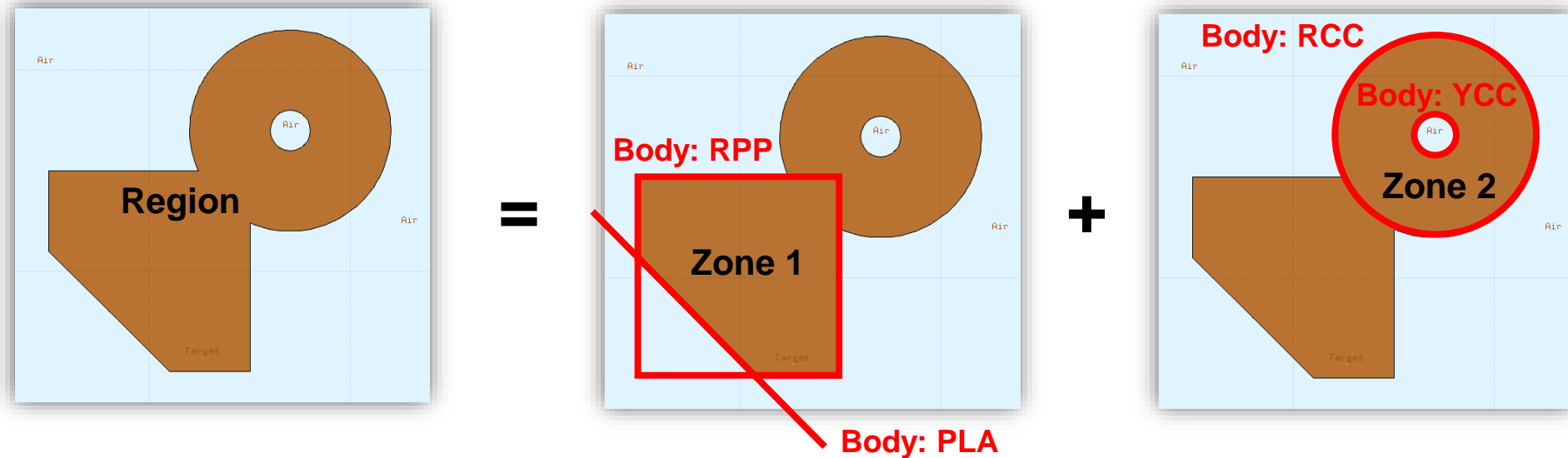
⌘ QUA target	Cxx: -5.0	Cyy: 2.0	Czz: -3.0
	Cxy: 1.0	Cxz: -4.0	Cyz: 1.0
	Cx: 6.0	Cy: 8.0	Cz: 2.0
	C: -1.0		

Regions (and zones)

Combining bodies to create your geometry model

Concept of zones and regions

- Regions are defined by combining FLUKA bodies using Boolean operations, more precisely:
 - Regions are obtained by the **union of sub-regions** (called **zones**); in the simplest case a region consists of a single zone
 - Zones are defined by **intersections** and/or **subtractions** of **bodies (Boolean zone expressions)**



Zones belonging to the same region can overlap!!

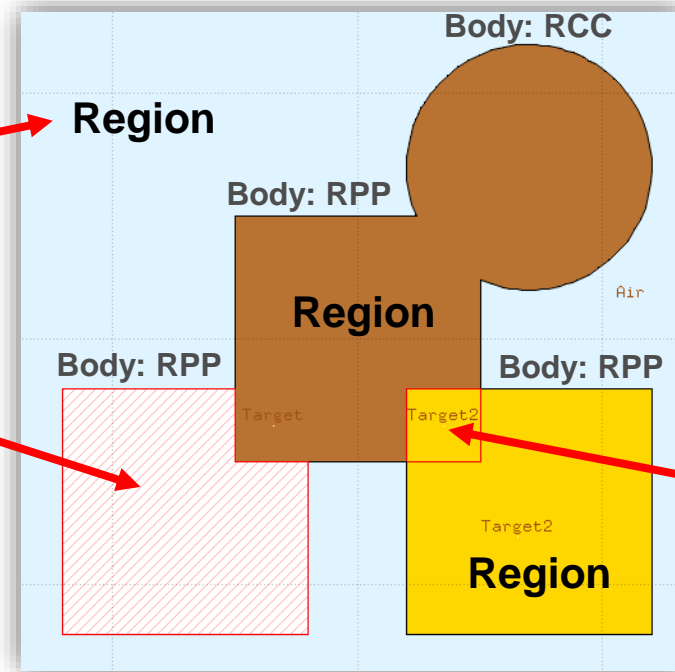
On the other hand, zones belonging to the same region must not necessarily be connected.

Basic principles ...

- ... when building geometries:
 - Zones and hence regions must be finite
 - **Each point in space must belong to one (and only one) region**
 - Note that also the region surrounding your principal geometry must be defined (e.g. air region)
 - Regions are of **homogeneous material composition** (i.e. only one material can be assigned to a region)

Note: do not forget to define the surrounding region

Error: volume does not belong to any region

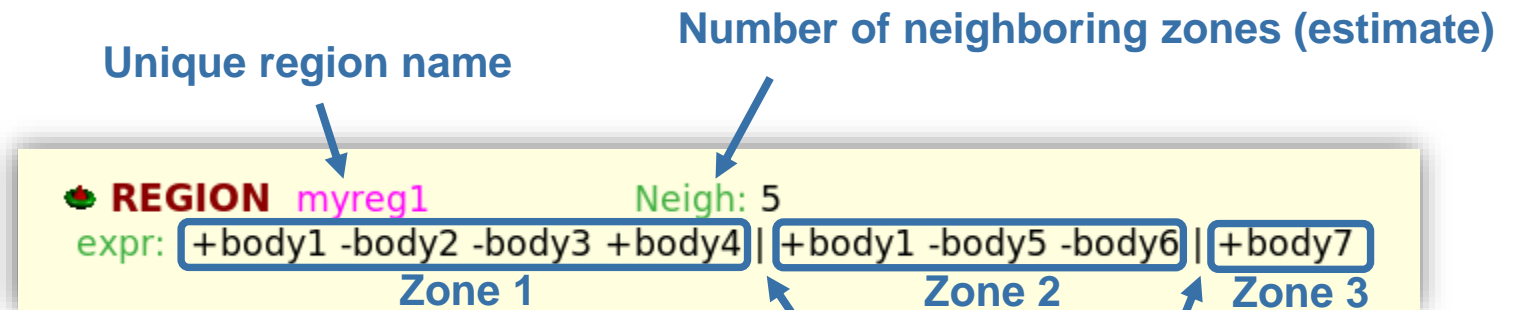


Error: volume belongs to two regions

Region definitions

- A region definition in FLUKA consists of:
 - A **unique region name** (alphanumeric identifier, 8 character maximum, case sensitive, must start with an alphabetical character)
 - An estimate of the **number of neighboring zones** (see backup)
 - A single **Boolean zone expression** or a **series of Boolean zone expressions** combined via the **union operator**

```
GEOBEGIN Accuracy: Option: Paren:
Geometry: Out: Fmt: COMBNAME
Title: Gold target
SPH spout x: 0. y: 0. z: 0.
R: 2000.
SPH spin x: 0. y: 0. z: 0.
R: 1000.
SPH big x: 0. y: 0. z: 0.
R: 10.
SPH small x: 0. y: 0. z: 0.
R: 6.
XYP plane z: 0.
ZCC cylinder x: 0. y: 0. R: 2.
END
Black hole
REGION BLKBODY Neigh: 5 Regions
expr: +spout -spin
Void around
REGION VOID Neigh: 5
expr: +spin -big | +big +plane | +small -plane -cylinder
Target
REGION TARGET Neigh: 5
expr: +big -small -plane | +big +cylinder -plane
END
GEOEND
```

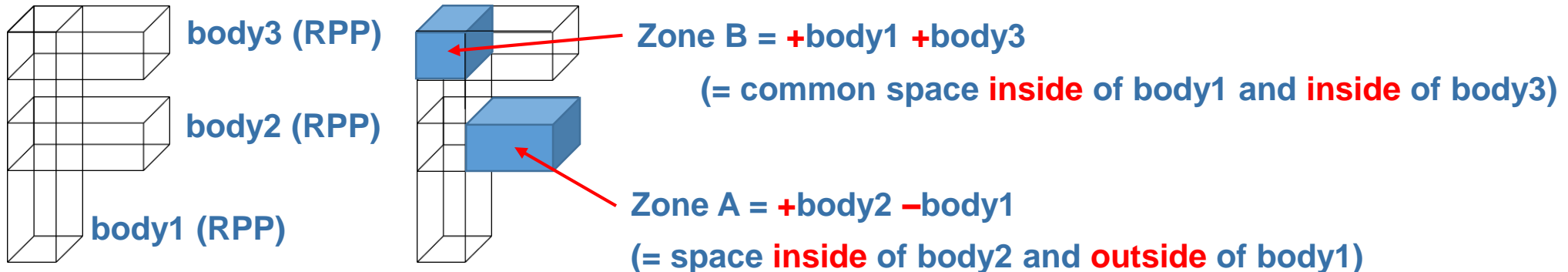


Expressions can be split over multiple lines, use as many lines as needed!

Union operators for combining zones

Boolean zone expressions

- Zones are described by a sequence of one or more bodies each being preceded by a **+** or **-** sign (remember: **+/-** means **inside/outside** a body)
 - **+body**: only the inner part of the body can belong to the zone (means that the zone being described is fully contained inside this body)
 - **-body**: only the outside of the body can belong to the zone (means that the zone being described is fully outside this body)

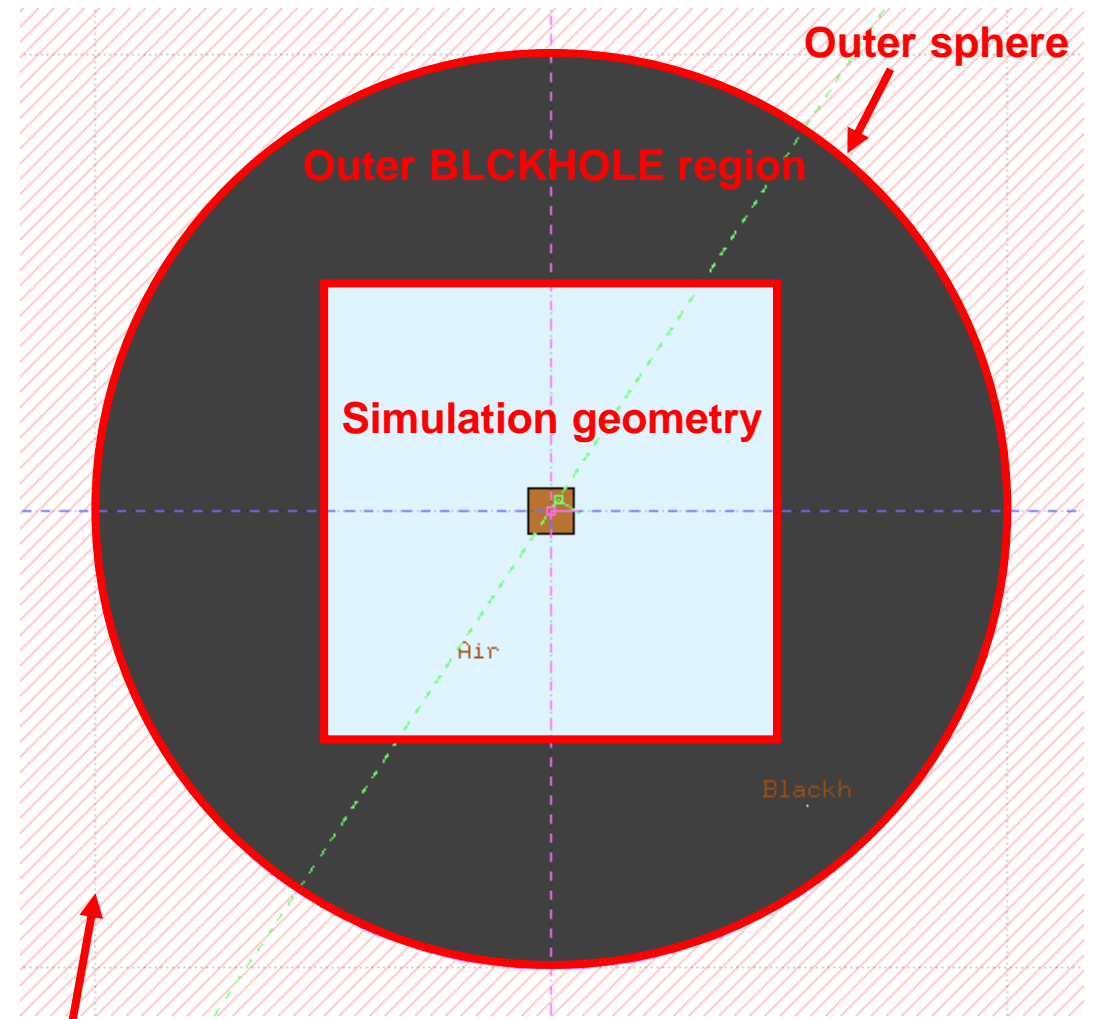


Parenthesis in region definitions

- Parentheses can be used to perform complex Boolean operations in the description of regions
- In evaluating the expressions, the highest operator precedence is given to parentheses, followed by +, - and the | operator
- Still, the extensive use of parenthesis is discouraged and can create unreadable and inefficient code
 - It is perfectly possible to implement (complex) geometries without using parenthesis
- In case you use parentheses, pay attention to the respective parameters on the GEOBEGIN card (optimization level and runtime evaluation of parenthesis)
 - See backup for details

Special region: the outer black hole confinement

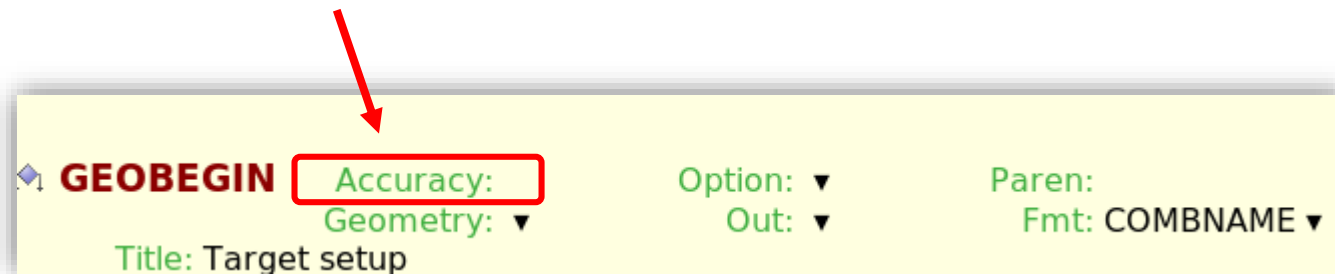
- FLUKA defines a special material called BLCKHOLE:
 - BLCKHOLE is an **all-absorbing material**
 - Particles vanish when entering a region filled with BLCKHOLE
- The entire geometry must be embedded in a region filled with BLCKHOLE
 - This avoids tracking particles to infinity
 - The outer surface of this BLCKHOLE region must be a single closed body (e.g. a sphere)



Outside of the BLCKHOLE enclosure,
the region can remain undefined!

Tracking accuracy

- The tracking accuracy parameter can be adjusted on the GEOBEGIN card
 - Defines absolute accuracy used for tracking and boundary identification (**in units of 10^{-6} cm**)
 - Should be larger than $A_r * L$, where:
 - L is the largest coordinate value in the geometry (without BLCKHOLE confinement)
 - A_r is the relative accuracy achievable in double precision ($\sim 10^{-14}$ - 10^{-15}).
 - By default, FLUKA tries to guess the parameter based on the second biggest body in your geometry (assuming the biggest one is the BLCKHOLE confinement)
 - Can be overwritten by explicitly setting the parameter on GEOBEGIN

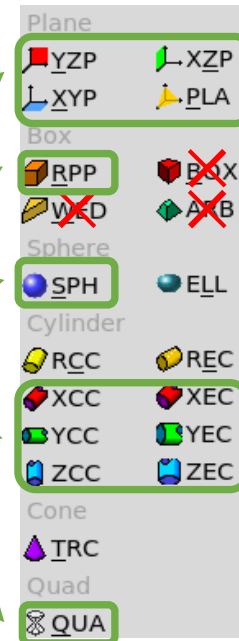


Tracking speed

- Number of bodies per zone:
 - If the number of bodies *per zone definition* becomes large this slows down the tracking
 - Rather segment in more zones than using too many bodies per zone (rule of thumb: not more than 10 bodies per zone definition)
 - Overlapping zones (for the same regions) can even be beneficial since particles may remain longer in one zone

- Recommendations on the use of bodies:

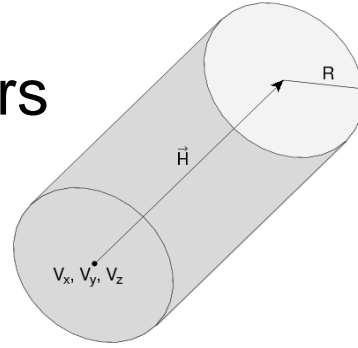
Prefer the bodies shown in green – tracking is faster since unnecessary boundary intersection calculations are avoided when the particle step is shorter than the distance to the region boundary.



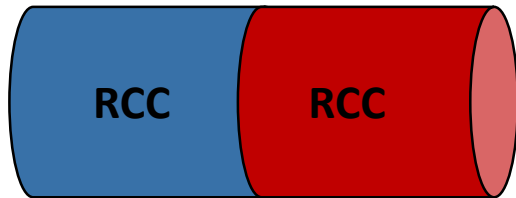
As noted before:
BOX, WED/RAW, ARB
are deprecated.
Do not use them.

Precision errors in region definitions

- Always use as many digits as possible in the definition of body parameters
 - Particularly for body heights (RCC, REC, TRC)
- Avoid touching surfaces when floating point operations are involved
 - For the above bodies, the position of the opposite face is calculated using the height vector
 - This can lead to precision errors during tracking in case two such bodies touch each other



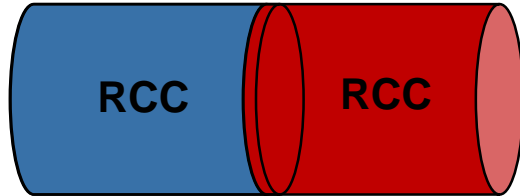
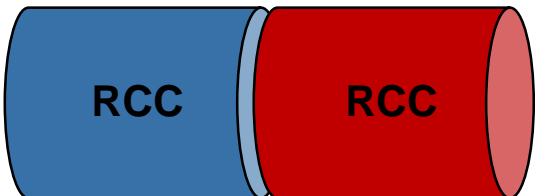
Example:
2 touching RCCs



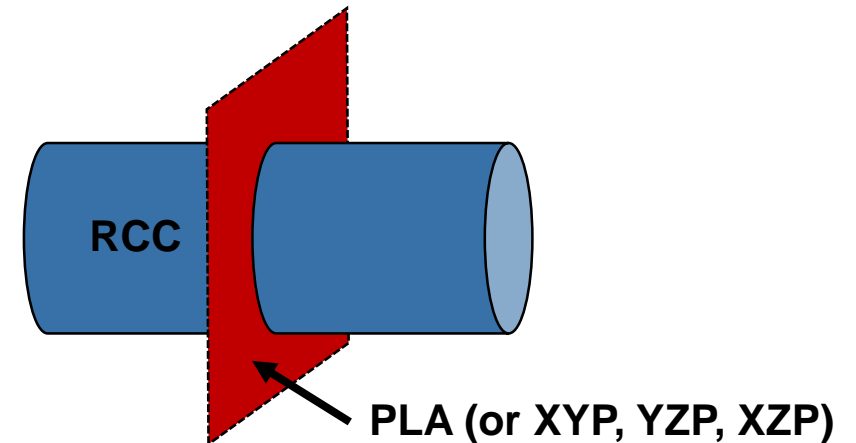
Possible precision errors:

Small gap between regions

Small overlap between regions



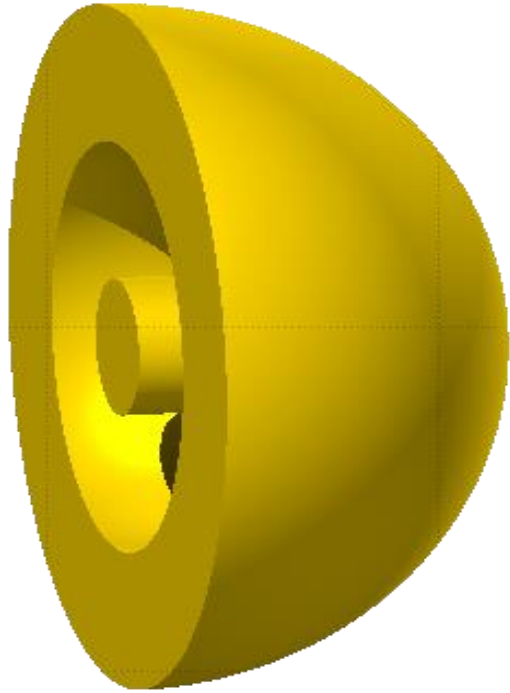
Better:
define only one
RCC and cut it
with plane



Geometry example and debugging

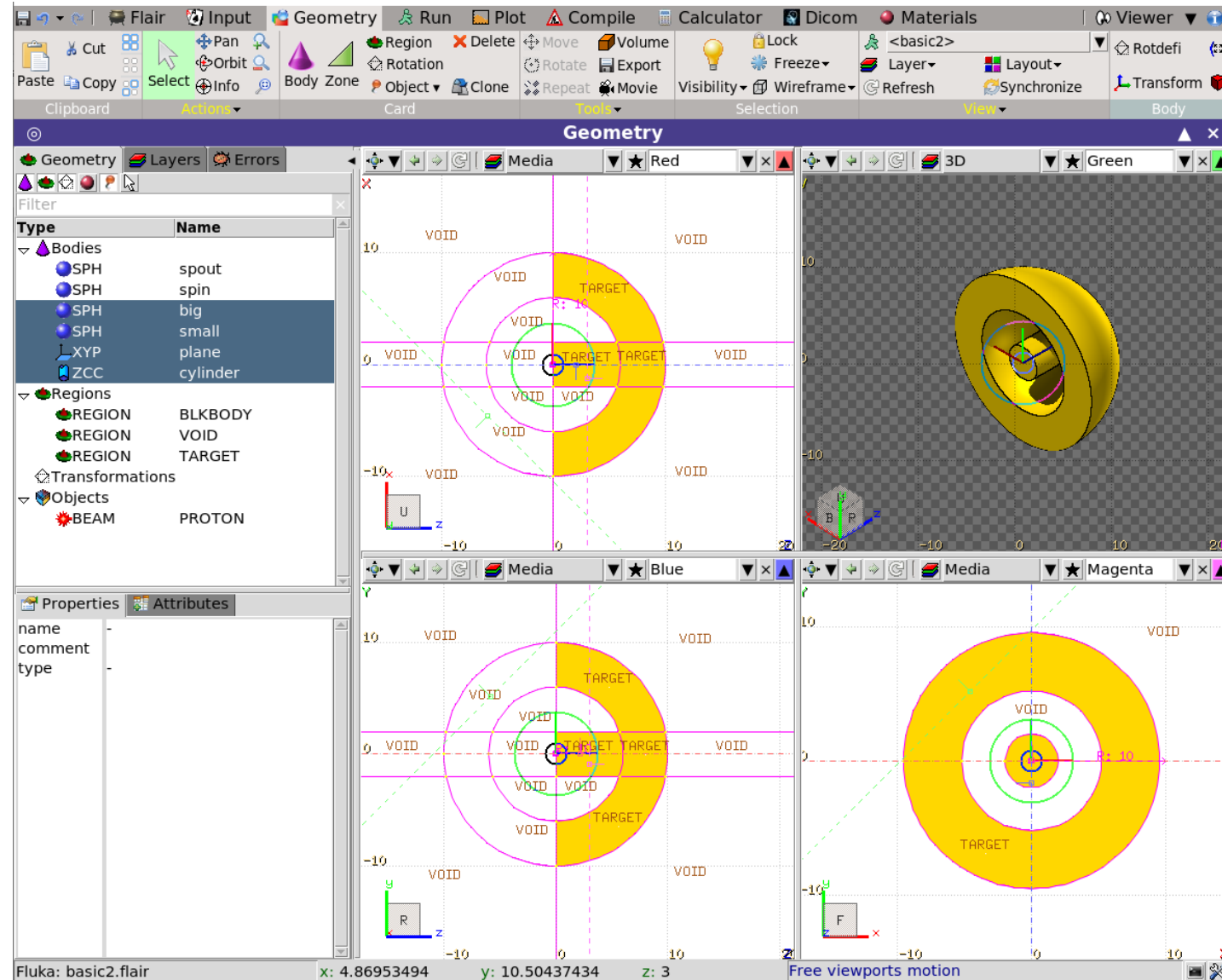
A practical example and how to detect geometry errors

Target made of gold – how to build it?



- Steps:

- Create enclosing BLCKHOLE region
- Create target region
- Create void region around the target



Step 1: surrounding BLACKHOLE region

```

GEOBEGIN  Accuracy:  Option:  Paren:
Geometry:  Fmt: COMBNAME
Title: Gold target
SPH spout  x: 0.  y: 0.  z: 0.
R: 2000.
SPH spin  x: 0.  y: 0.  z: 0.
R: 1000.
SPH big  x: 0.  y: 0.  z: 0.
R: 10.
SPH small  x: 0.  y: 0.  z: 0.
R: 6.
XYP plane  z: 0.
ZCC cylinder  x: 0.  y: 0.  R: 2.
END BLACKBODY region (1 zone)
Black hole
REGION BLKBODY  Neigh: 5
expr: +spout -spin
void around
REGION VOID  Neigh: 5
expr: +spin -big | +big +plane | +small -plane -cylinder
Target
REGION TARGET  Neigh: 5
expr: +big -small -plane | +big +cylinder -plane
END
GEOEND
    
```

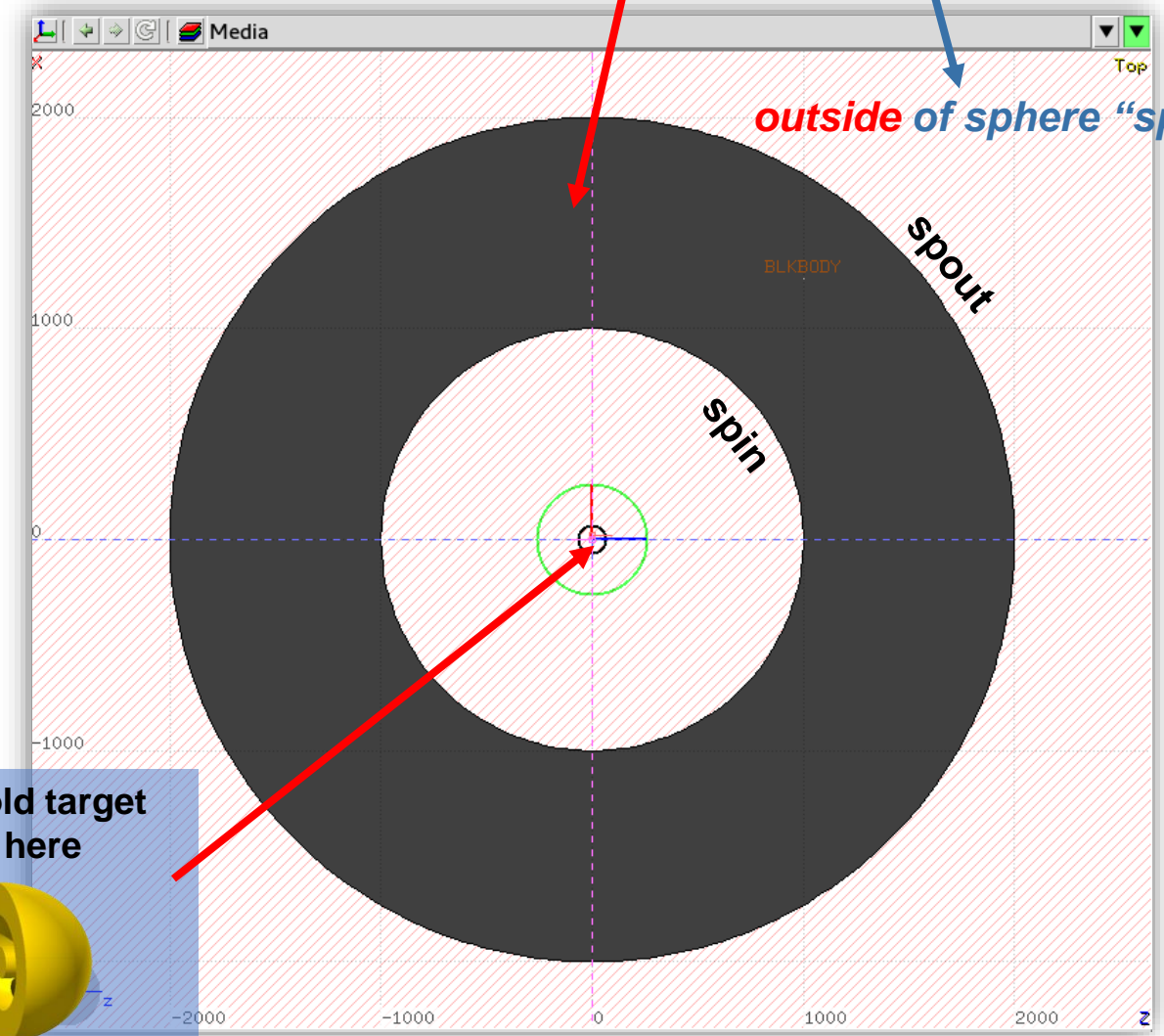
bodies

BLACKBODY region (1 zone)

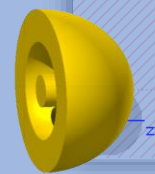
inside of sphere "spout"

zone = +spout -spin

outside of sphere "spin"



Our gold target will be here



Step 2: Gold target – zone 1

```

GEOBEGIN  Accuracy:      Option:  ▾  Paren:
          Geometry:  ▾  Out:  ▾  Fmt: COMBNAME ▾
Title: Gold target
● SPH spout  x: 0.      y: 0.      z: 0.
              R: 2000.
● SPH spin  x: 0.      y: 0.      z: 0.
              R: 1000.
● SPH big   x: 0.      y: 0.      z: 0.
              R: 10.
● SPH small x: 0.      y: 0.      z: 0.
              R: 6.
XYP plane  z: 0.
ZCC cylinder x: 0.      y: 0.      R: 2.
END
Black hole
REGION BLKBODY  Neigh: 5
expr: +spout -spin
TARGET region (2 zones)
Target
REGION TARGET  Neigh: 5
expr: +big -small -plane | +big +cylinder -plane
END
GEOEND
  
```

bodies

TARGET region (2 zones)

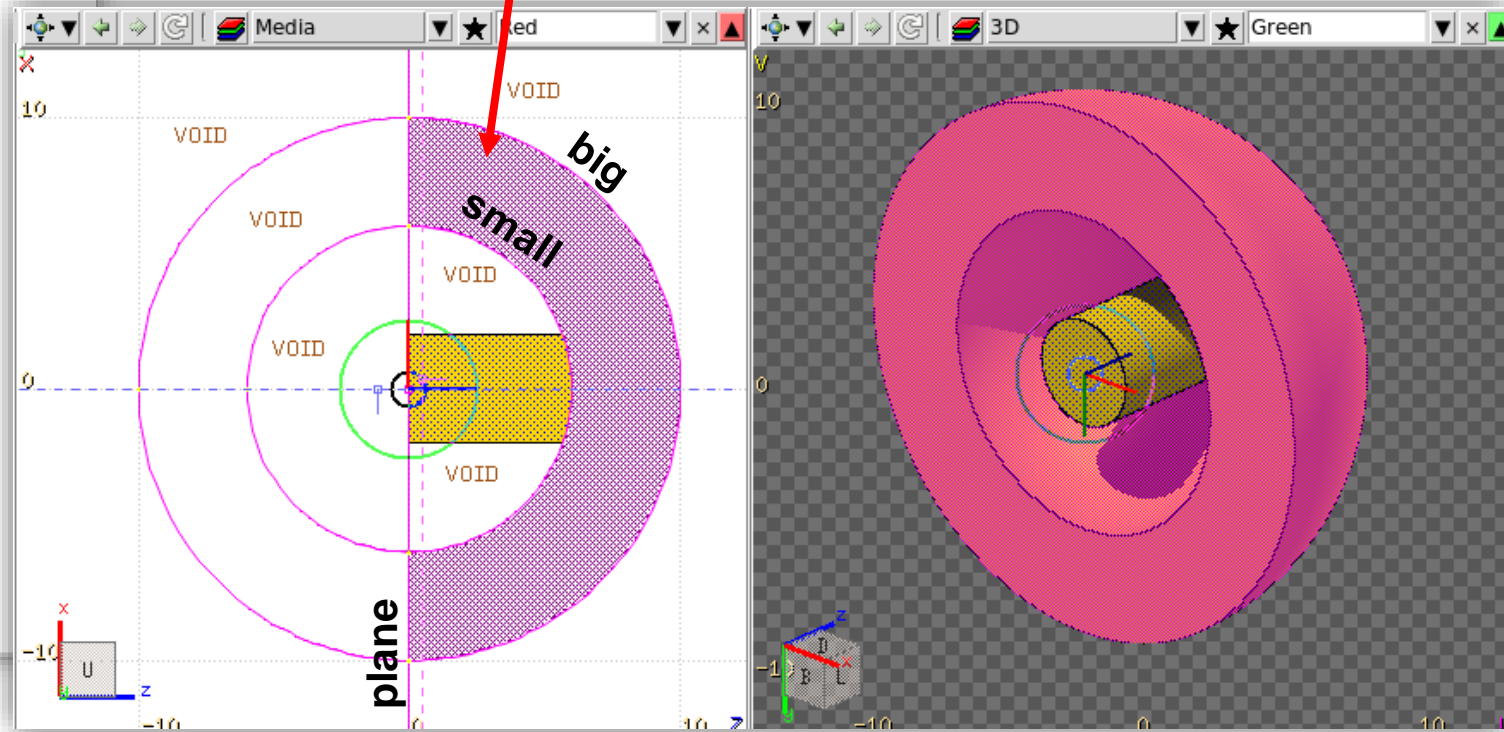
zone 1

inside of sphere "big"

outside of plane "plane"

$$\text{zone} = +\text{big} - \text{small} - \text{plane}$$

outside of sphere "small"



Step 2 (cont.): Gold target – zone 2

GEOBEGIN Accuracy: Option: ▾ Paren: Fmt: COMBNAME ▾
 Geometry: ▾ Out: ▾

Title: Gold target

- SPH spout x: 0. y: 0. z: 0.
R: 2000.
- SPH spin x: 0. y: 0. z: 0.
R: 1000.
- SPH big x: 0. y: 0. z: 0.
R: 10.
- SPH small x: 0. y: 0. z: 0.
R: 6.
- XYP plane z: 0.
- ZCC cylinder x: 0. y: 0. R: 2.

bodies

◆ END

Black hole

● REGION BLKBODY Neigh: 5
 expr: +spout -spin

TARGET region (2 zones)

Target

● REGION TARGET Neigh: 5
 expr: +big -small -plane | **+big +cylinder -plane**

◆ END

zone 2

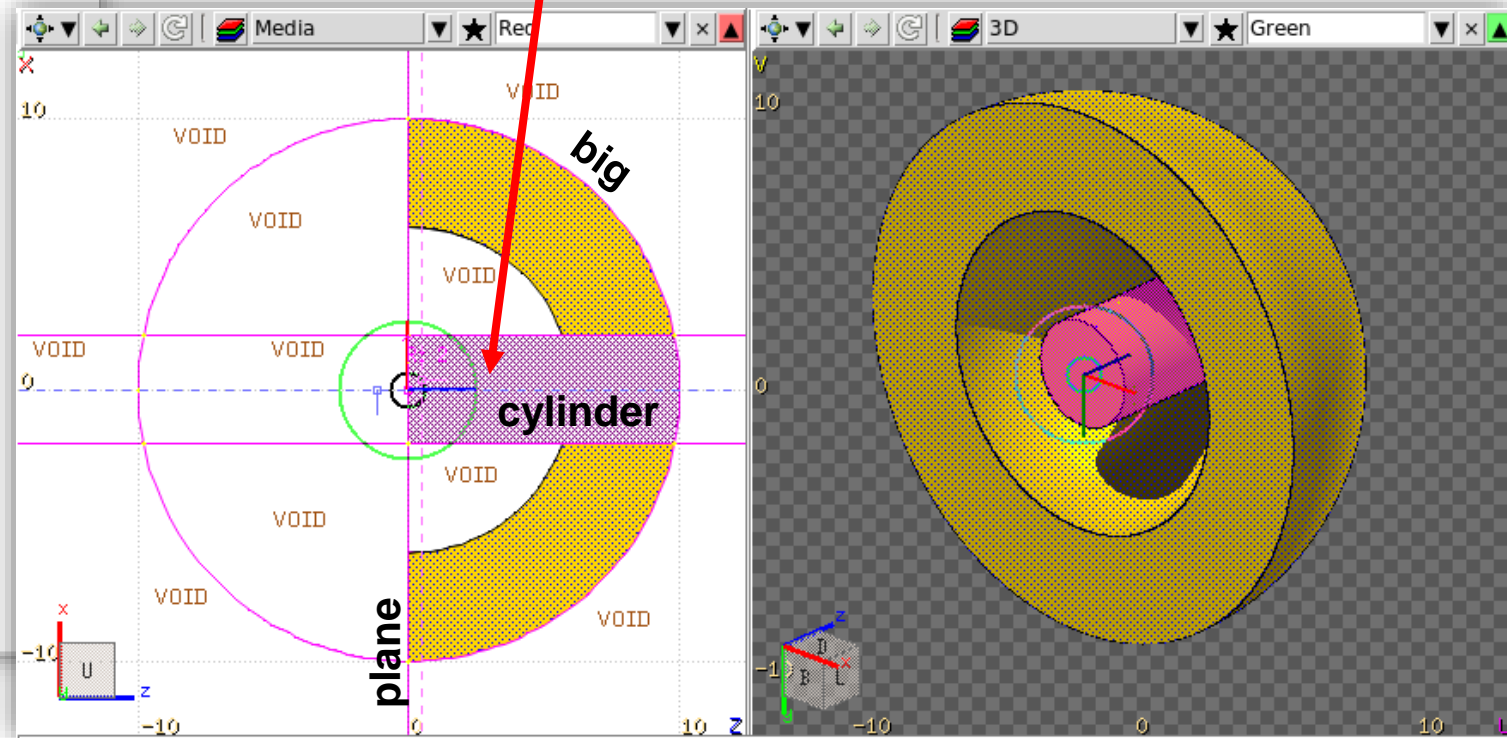
◆ GEOEND

inside of sphere "big"

outside of plane "plane"

zone = +big +cylinder -plane

inside of infinite cylinder "cylinder"



Step 3: the VOID region

GEOBEGIN Accuracy: Geometry: Option: Parent: Fmt: COMBNAME

Title: Gold target

bodies

SPH	spout	x: 0.	y: 0.	z: 0.
		R: 2000.		
SPH	spin	x: 0.	y: 0.	z: 0.
		R: 1000.		
SPH	big	x: 0.	y: 0.	z: 0.
		R: 10.		
SPH	small	x: 0.	y: 0.	z: 0.
		R: 6.		
XYP	plane	z: 0.		
ZCC	cylinder	x: 0.	y: 0.	R: 2.

END

Black hole

REGION BLKBODY Neigh: 5
expr: +spout -spin

Void around

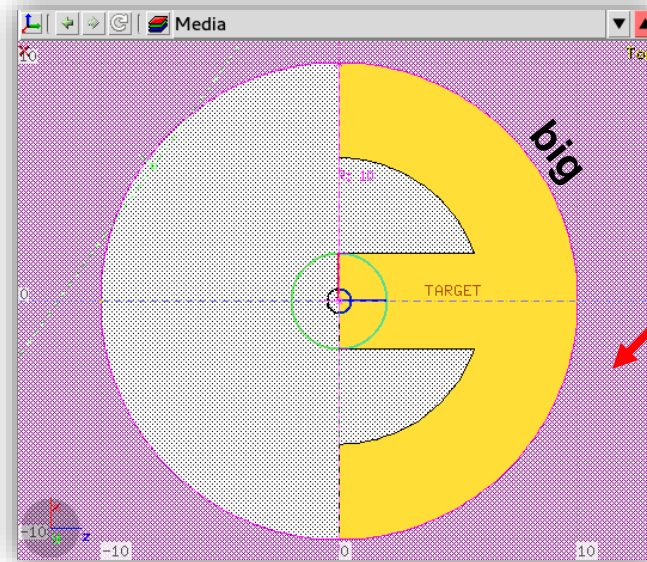
REGION VOID Neigh: 5
expr: +spin -big | +big +plane | +small -plane -cylinder

REGION TARGET
expr: +big -small -plane | +big +plane

VOID region (3 zones)

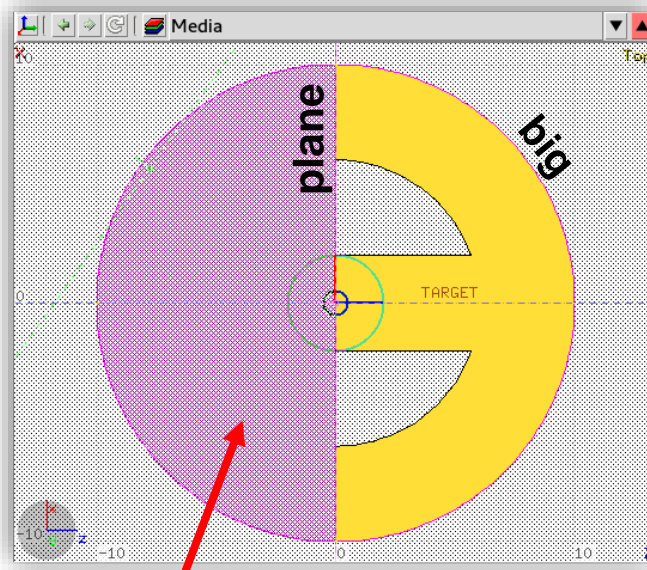
END

GEOEND

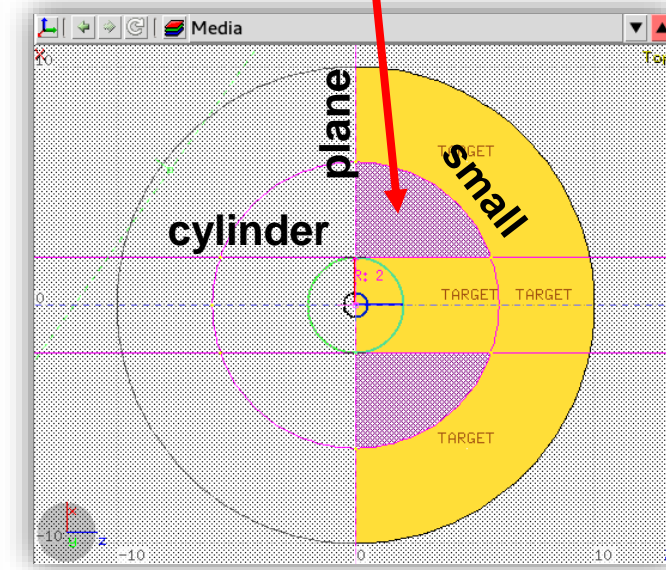


Inner boundary of BLCKHOLE region

zone = +spin -big



zone = +big +plane



zone = +small -plane -cylinder

Step 3 (cont.): the VOID region

GEOBEGIN Accuracy: Option: Parent: Geometry: Fmt: COMBNAME

Title: Gold target

bodies

SPH	spout	x: 0.	y: 0.	z: 0.
		R: 2000.		
SPH	spin	x: 0.	y: 0.	z: 0.
		R: 1000.		
SPH	big	x: 0.	y: 0.	z: 0.
		R: 10.		
SPH	small	x: 0.	y: 0.	z: 0.
		R: 6.		
XYP	plane	z: 0.		
ZCC	cylinder	x: 0.	y: 0.	R: 2.

END

Black hole

REGION BLKBODY Neigh: 5
expr: +spout -spin

Void around

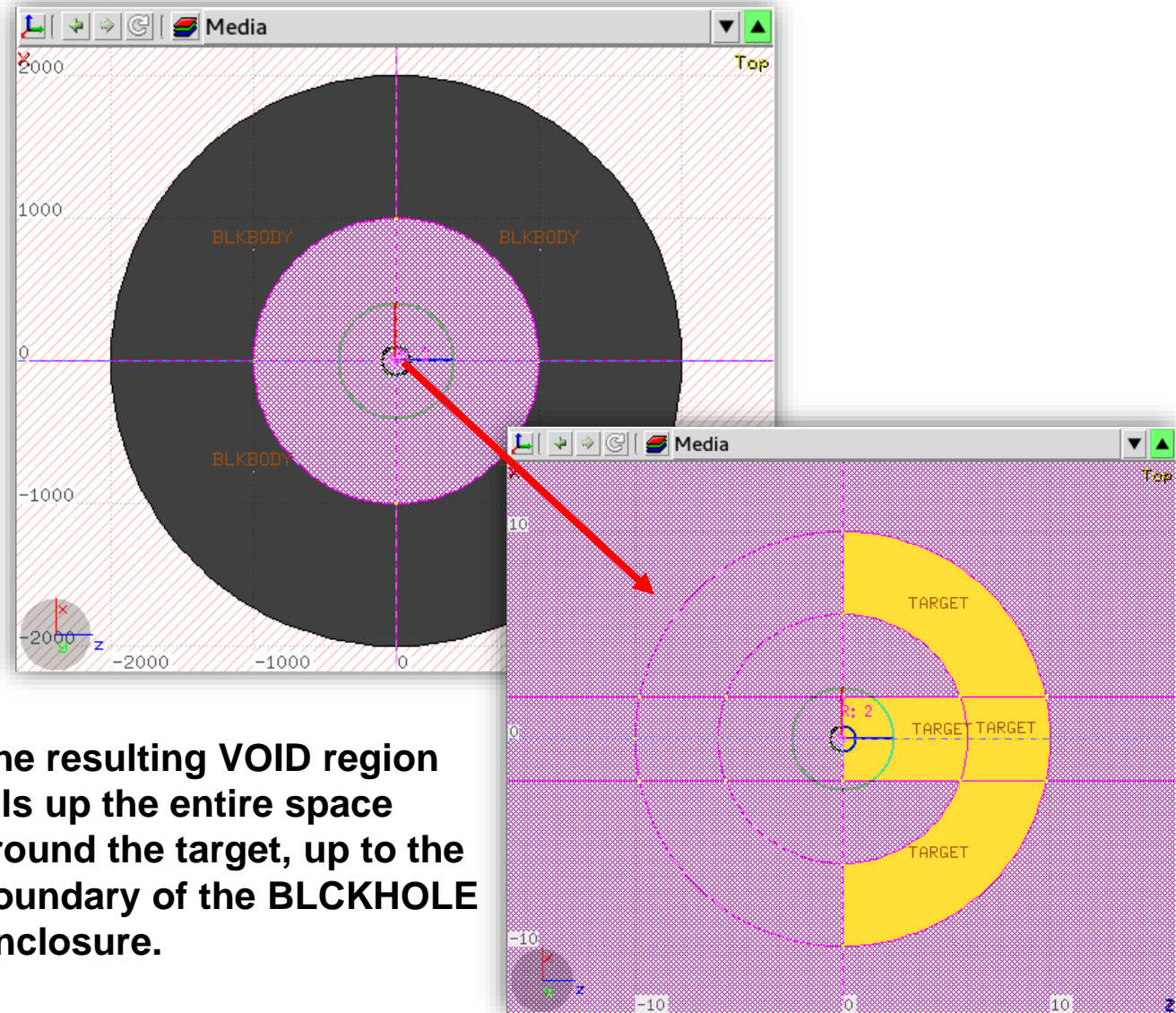
REGION VOID Neigh: 5
expr: +spin -big | +big +plane | +small -plane -cylinder

REGION TARGET
expr: +big -small -plane | +big +small -plane -cylinder

VOID region (3 zones)

END

GEOEND



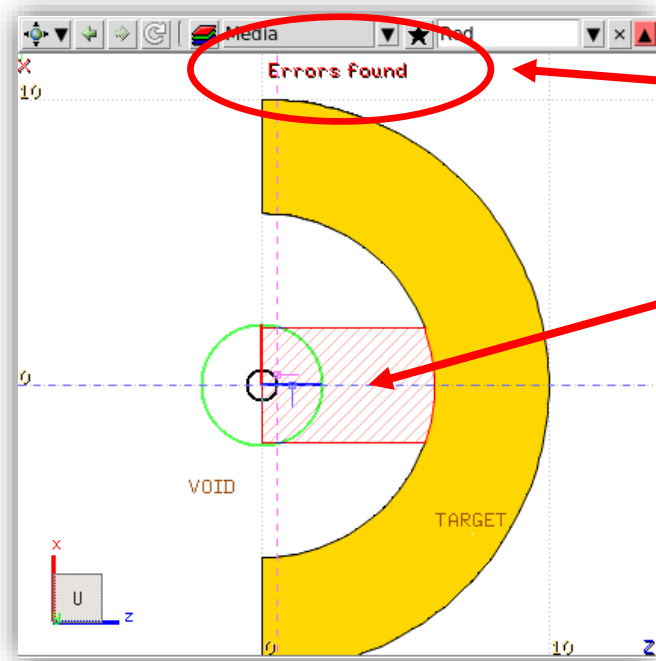
The resulting VOID region fills up the entire space around the target, up to the boundary of the BLCKHOLE Enclosure.

Some remarks on geometry errors

- During execution, the code always needs to know the region where a particle is located at every step:
 - The program **will stop** only if a particle position **does not belong to any region**. An error message will be printed in the **.err** file with the particle position.
 - **IMPORTANT:** the code **will NOT stop** if a particle position **belongs to more than one region**. It will accept the first region it finds but the results will be completely unreliable.
- Further types of errors:
 - Problem space not enclosed by a BLACKHOLE region
 - Never start a primary particle along a surface. You could get a geometry error even if the geometry is correct because FLUKA cannot determine the region.
 - Precision errors (see remarks on the choice of bodies in the previous section)
 - Lattice replica and basic cell mismatch (see advanced topics)

Debugging of geometry errors

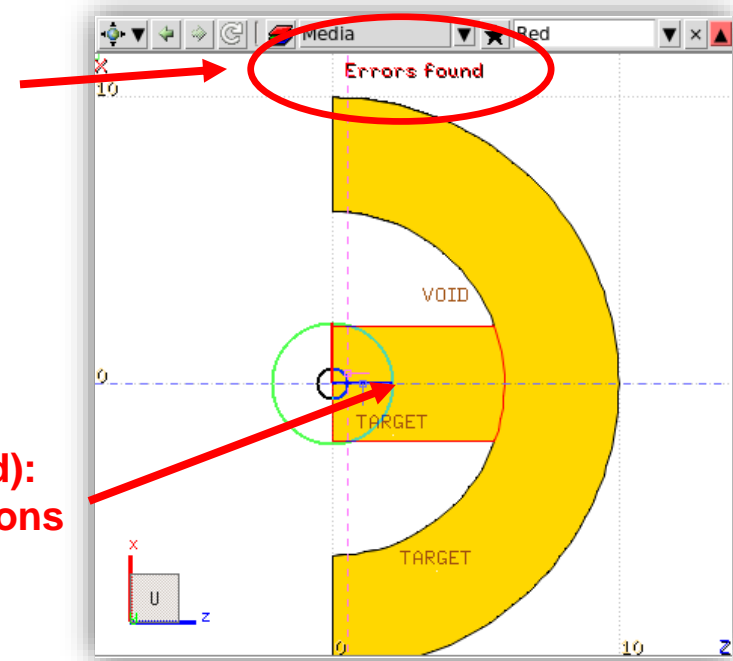
- Inspect the .err file for relevant messages
- Use the Flair geoviewer
 - Note that errors are only detected if they are within the view planes (i.e., if they are outside the viewed planes, they are not shown)



Pay attention if the label "Errors found" appears.

Red hatched area: volume not occupied by any region

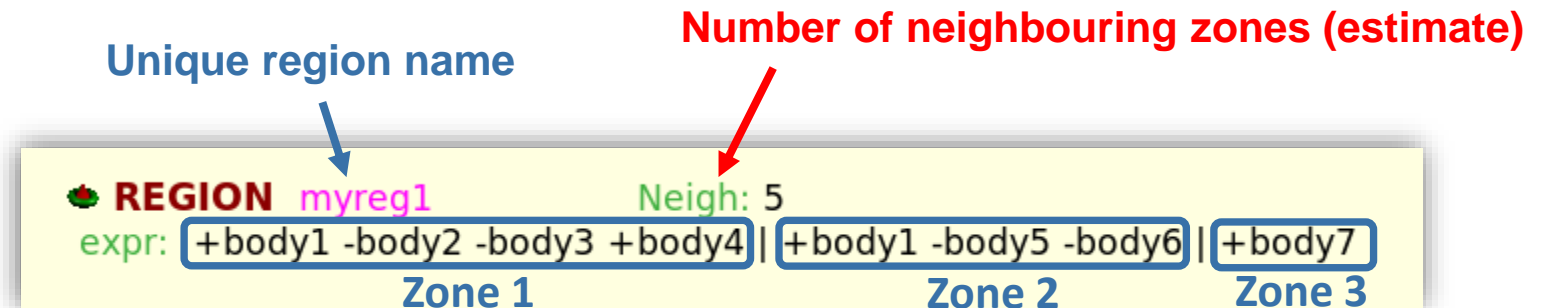
Red framed area (not hatched): volume occupied by two regions



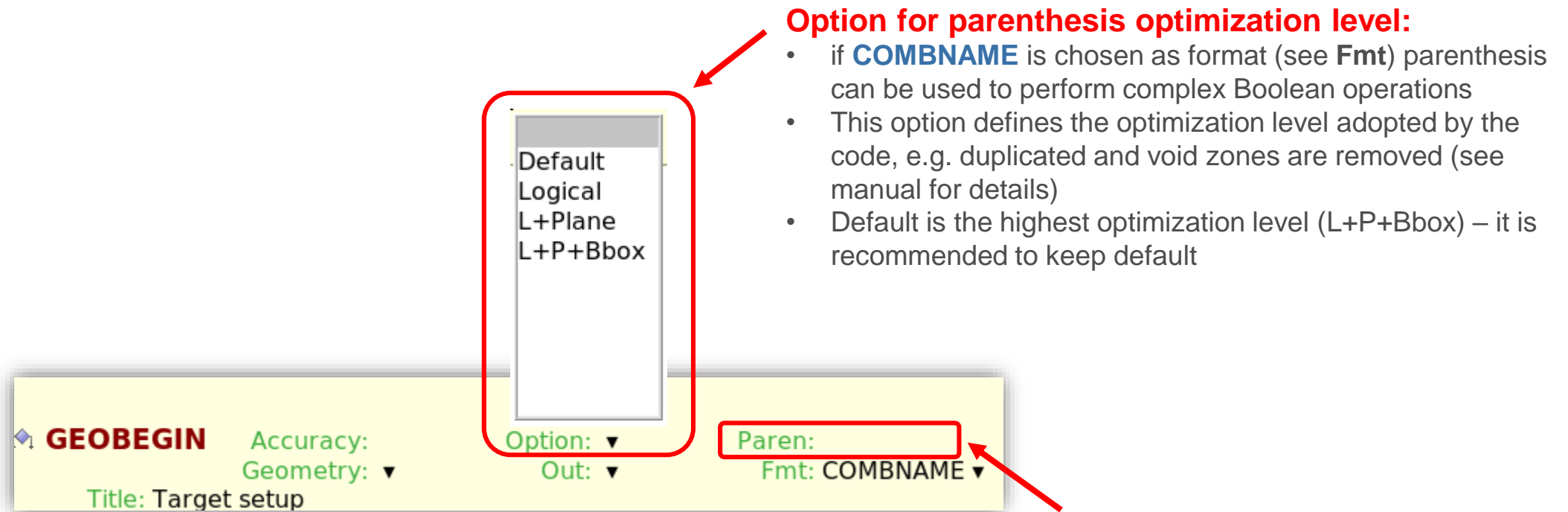
Backup

The number of adjacent zones (REGION card)

- “**Neigh**” parameter on the **REGION** card:
 - Is a rough estimate of the number of zones a particle can enter when leaving the current region zones (**5 by default**). What actually matters is the sum over all regions, defining the size of the contiguity list. **Note: for many simulation cases, the default value is fine.**
 - While tracking, the program searches in the contiguity list for the neighboring zones of each zone. If the zone is not yet in the list, the whole geometry is scanned and it is added to the list with its neighboring zones.
 - When the limit is reached (i.e. the list is full) the code prints a warning: **GEOMETRY SEARCH ARRAY FULL. This is not lethal: the calculation continues but with a reduced efficiency.**
 - If you have more than 1000 regions, you must issue a GLOBAL card putting in WHAT(1) a higher limit (not beyond 10000).



Parenthesis-related settings on GEOBEGIN card



Option for parenthesis optimization level:

- if **COMBNAME** is chosen as format (see **Fmt**) parenthesis can be used to perform complex Boolean operations
- This option defines the optimization level adopted by the code, e.g. duplicated and void zones are removed (see manual for details)
- Default is the highest optimization level (L+P+Bbox) – it is recommended to keep default

Option for selecting runtime parenthesis evaluation:

- Depending on the chosen factor, parenthesis are not expanded during initialization but at runtime

