Lattice

FLUKA geometry has *replication* (lattice) capabilities

Only one *level* is implemented

The user defines lattice positions in the geometry

And provides transformation rules in a routine

The lattice identification is available for scoring

In the geometry

The regions which constitute the Elementary cell to be replicated, have to be defined in detail

The Lattices have to be defined as "empty" regions in their correct location

The lattice regions are declared as such with a LATTICE card at the end of the geometry

In the LATTICE card , the user also assigns lattice numbers to the lattices. These numbers will identify the replicas in all FLUKA routines

Several basic cells and associated lattices can be defined within the same geometry, one LATTICE card will be needed for each set

Non-replicas carry the lattice number 0

Lattices and plain regions can coexist in the same problem

The Example

Sampling calorimeter, with circular section:

First basic cell : 3 layers, Pb, scintillator, air , 0.5, 1.4, 0.1 cm
correspond to regions 3,4,5
Repeated 1 + 14 times

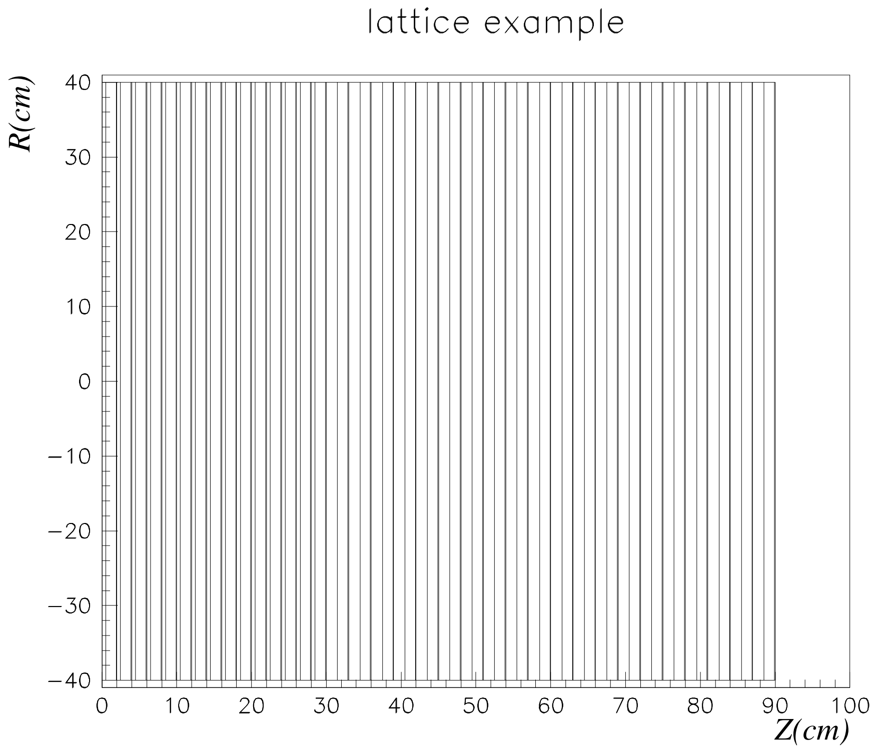
Second basic cell: Same, but with 1.5, 1.4, 0.1 cm thickness
correspond to regions 20,21,22
Repeated 1+19 times

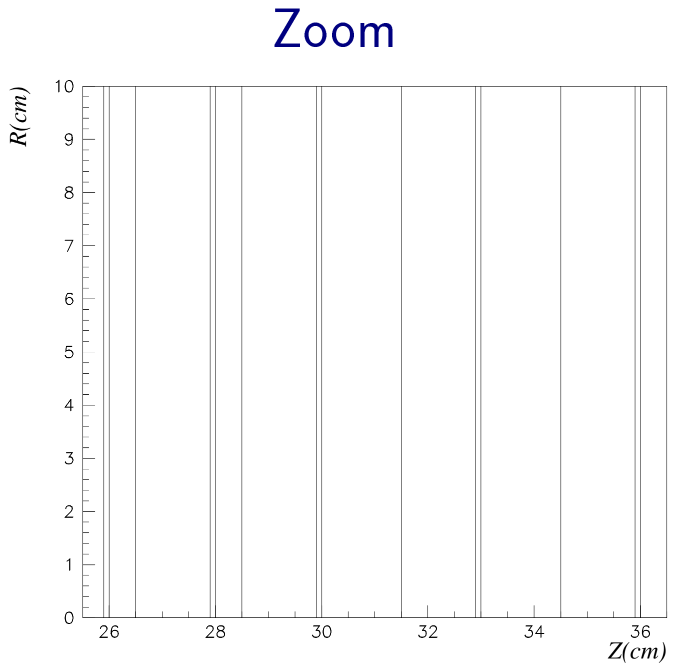
```
LATTICE       6.00000  19.00000             101.0000  114.00
LATTICE      23.00000  41.00000             116.0000  134.00
```

Region nb. 6 to 19 are the "placeholders" for the first set replicas. We assign to them lattice numbers from 101 to 114

Region nb. 23 to 41 are the "placeholders" for the second set replicas. We assign to them lattice numbers from 116 to 134

Plot of the Example





The lattic routine

The actual transformation from the lattice cell to the elementary cell must be provided through the lattic routine

SUBROUTINE LATTIC (XB, WB, DIST, SB, UB, IR, IRLTGG, IFLAG)

IRLTGG is the current lattice number (.. from the LATTICE card..)

XB,WB are vectors with the current particle position and directions

the routine must give back SB, UB, position and direction

transported to the elementary cell

The

ENTRY LATNOR (UN, IRLTNO)

must provide the transformation for *normal versors to boundaries*

(UN is both the in and out versor)

The lattic routine- example

In our example:

```
IF ( IRLTGG .LE. 114 ) THEN
     SB (1) = XB (1)
     SB (2) = XB (2)
     SB (3) = XB (3) - 2.0D+00 * ( IRLTGG - 100 )
     UB (1) = WB (1)
     UB (2) = WB (2)
     UB (3) = WB (3)
  ELSE IF ( IRLTGG .LE. 134 ) THEN
     SB (1) = XB (1)
     SB (2) = XB (2)
     SB (3) = XB(3) - 3.0D+00 * ( IRLTGG - 115 )
     UB (1) = WB (1)
     UB (2) = WB (2)
     UB (3) = WB (3)
  END IF
```

And the UN transformation is the identity

More complex cases can involve reflections and rotations. for instance,

The lattic routine- example

for a reflection around the z axis :

```
UN (1) = UN (1)
UN (2) = UN (2)
UN (3) =-UN (3)
```

Input file, and output quantities

Materials and other properties have to be assigned only to the regions constituting the basic cell(s).

In all (user) routines the region number refers to the corresponding one in the elementary cell. In our example, all the scintillator layers of the first set will carry region number 4

This applies also to the summary output in the .out file, and in the scoring by regions.

The lattice identity can be recovered by the *lattice number*, as set in the LATTICE card.

In particular, the LUSRBL routine allows to manage the scoring on lattices in the special EVENTBIN structure.

the EVENTBIN special binning

EVENTBIN or USERBIN with what(1)=8 :
Special user-defined 3D binning. Two variables are discontinuous (e.g. region number), the third one is continuous, but not necessarily a space coordinate.

Variable	type	default	Override routine
1st	integer	region number	MUSRBR
2nd	integer	lattice cell number	LUSRBL
3rd	continuous	pseudorapidity	FUSRBV

eventbin example

In our example, we use

on variable 1 the region number , (excl. 1==black hole)

on variable 2 the layer number, from 1 to 35 (LUSRBL)

on variable 3 the radius (FUSRBV), in 4 binnings 10 cm wide.

Energy deposition is scored with (binning 1) and without (binning 2) scintillation quenching

The results are written in a binary file (unit 25)

The program lattic_ntuple processes the binary files to produce a PAW ntuple

Assuming that the detector is read out in 7 longitudinal samplings, 5 cells each.

ntuple variables

nt number 2000

```
es1r1--es3r4 : electromagnetic  sampling 1to3 , radii 1 to 4
es1t--es3t   : electromagnetic sampling 1to3, radially integrated
etot         : em total
hs1r1--hs4r4 : hadronic  sampling 1to4 , radii 1 to 4
hs1t--hs4t   : hadronic  sampling 1to4, rad. integ.
htot         : had total
total        : ..guess..
```

same with “q” : quenched