Scoring physical quantities I

Introduction to built-in estimators

3D distributions (**USRBIN**) & 1D-2D plots

FLUKA scoring

- It is said that Monte Carlo (MC) is a **“mathematical experiment”**; the MC equivalent of the result of a real experiment (*i.e.*, of a **measurement**) is called an **estimator**
- Just as a real measurement, an estimator is obtained by sampling from a statistical distribution and has a **statistical error** (and in general also a **systematic** one)
- There are often several different techniques to measure the same physical quantity: in the same way, **the same quantity can also be calculated using different kinds of estimators**
- FLUKA offers **numerous different estimators**, *i.e.* **scoring** for various quantities of interest can be requested directly from the input file

Definitions

- N : **number of identical particles**
- N_0 : **number of atoms per unit volume**
- λ : **mean free path**, *i.e.* average distance travelled by a particle in a material before an interaction. It depends on the material, particle type and energy
- l : **total distance travelled**
- v : **average particle velocity**

Cross-section

- $\Sigma\,[cm^{-1}] = 1/\lambda\,[cm]$: **macroscopic cross-section**, *i.e.* probability of interaction per unit distance. It depends on the material, particle type and energy.

- $\sigma = \frac{\Sigma}{N_0} = atom\ effective\ area\,, [barn = 10^{-24} cm^2]$: **microscopic cross-section**, *i.e.*
  - the **area of an atom weighted with the probability of interaction** (hence the name "cross-section")
  - or the **probability of interaction per unit length, with the length measured in atoms/cm^2**

- The microscopic and macroscopic cross-section have a similar physical meaning of "probability of interaction per unit length", with length measured in different units. Thus, **the number of interactions** can be obtained by multiplying them by the corresponding particle **track-length**

Reaction rate and fluence

- $R = \Sigma\Phi V$: number of reactions in a given time interval inside the volume V
  (where Φ is the fluence and the product $\Sigma\Phi$ is integrated over energy or velocity)

- $\dot{R} = N\frac{dl}{dt}\Sigma = Nv\Sigma$: **reaction rate**

- $\frac{d\dot{R}}{dV} = \frac{dN}{dV}v\,\Sigma = n(\boldsymbol{r},v)v\Sigma$: reaction rate inside the volume element dV

- $\Phi(\boldsymbol{r},v) = n(\boldsymbol{r},v)dl\,,[cm^{-2}]$: **fluence**, *i.e.* time integral of the flux density
  - Fluence is expressed in **particles per cm²** but in reality represents the **density of particle tracks [cm / cm³] !**

- $\dot{\Phi}(\boldsymbol{r},v) = n(\boldsymbol{r},v)v\,,[cm^{-3}cm\;s^{-1}] = [cm^{-2}s^{-1}]$: **fluence rate** or **flux density**

Built-in and user scoring

- Several **pre-defined estimators** can be activated in FLUKA; one usually refers to these estimators as **built-in scoring** capabilities
- Users may build their own custom scoring through **user routines**
- However, **built-in scoring**:
  - covers most **common needs**
  - has been **extensively tested**
  - takes **biasing automatically into account**
  - has **refined algorithms** for track subdivision (apportioning)
  - comes with **utility programs** that allow to evaluate statistical errors
- Therefore users are strongly encouraged to **prefer built-in scorings** wherever possible
- Standard scoring can be adapted by means of simple user routines (**`fluscw.f`**, **`comscw.f`**), activated via **`USERWEIG`** card

FLUKA scoring

What?

Energy deposition and derivatives (dose), fluence or current versus energy, angle or other kinematic variables, time, DPA, residual activity…

Where?

In regions, across boundaries, on region-independent grids

When?

At the end of each cycle or at each event

Output?

Saved in `[inputname]nnn_fort.##` files, where `nnn` is the cycle number & `##` is the logical unit number chosen by the user

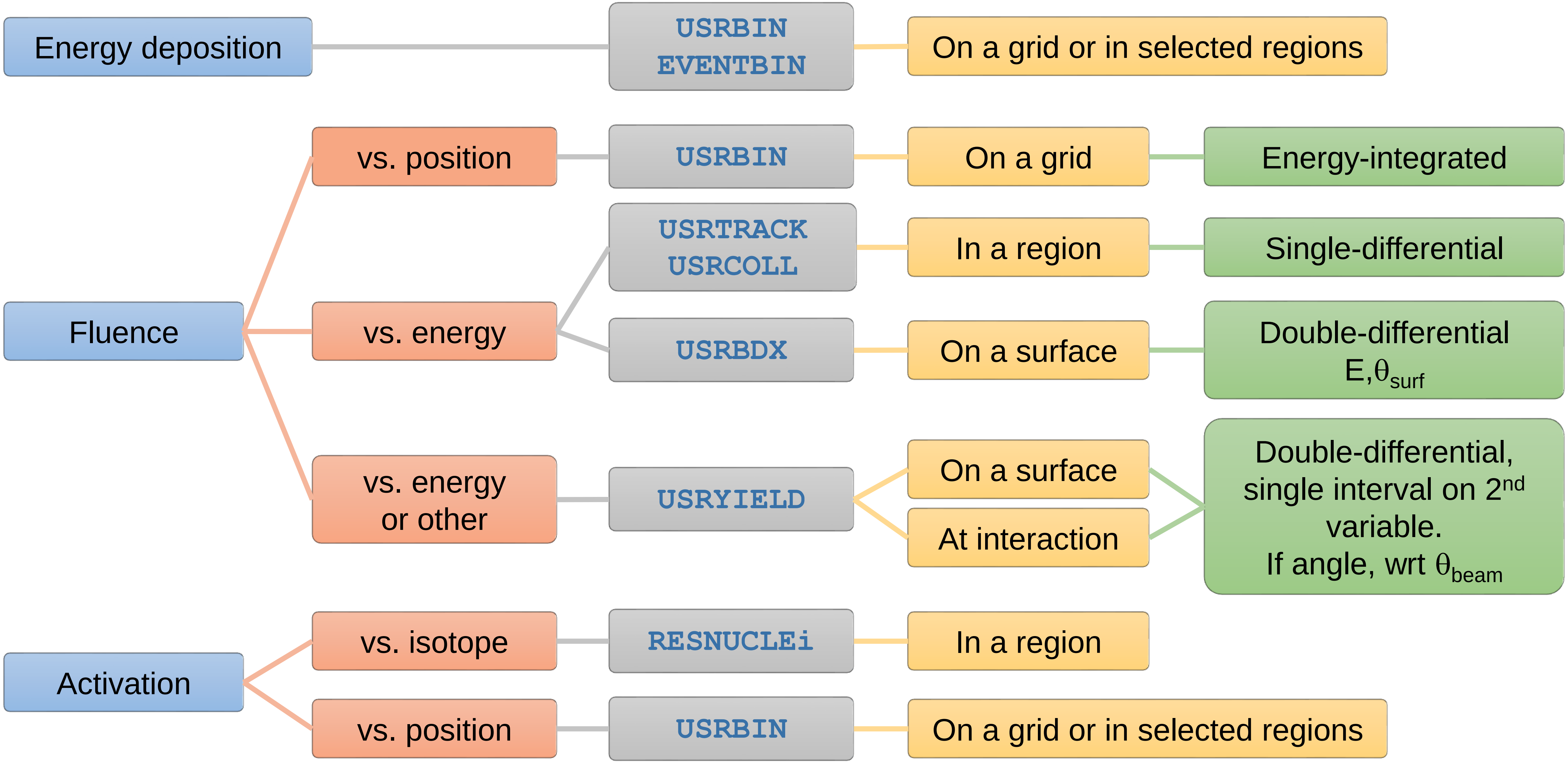
Results?

Post-processing utilities merge cycles, calculate average and rms, provide data files for plotting. Available via **Flair**

Results normalised **per primary**

User code needed for processing of custom scoring!

The FLUKA estimator zoo



Main FLUKA estimators

- **USRBIN** scores the **spatial distribution** of **energy density** or **fluence** (or star density) in a **region** or **regular mesh** (cylindrical, Cartesian) described by the user
- **USRBDX** scores average **$d^2\Phi/dEd\Omega$ (double-differential fluence or current)** of a given type or family of particles on a **given surface**
- **USRTRACK** (**USRCOLL**) scores average **$d\Phi/dE$ (differential fluence)** of a given type or family of particles in a **given region**
- **USRYIELD** scores a **double differential yield** of particles on a **given surface**
  - The distribution can be with respect to energy and angle, but also other more “exotic” quantities
- All scorings write their results into **logical output units assigned by the user**
  - the unit numbers must be **>20**
  - The only exception is **SCORE** – which scores **energy deposition** (or number of stars) in all regions – whose output is printed in the **standard output**

More "special" scoring cards

- `DETECT` scores **energy deposition** for each event (primary history) in coincidence or anti-coincidence with a trigger
- `EVENTBIN` is like `USRBIN`, but prints the binning output **after each event** instead of an average over histories
- `TCQUENCH` sets scoring **time cut-offs** and/or **Birks quenching** parameters for binnings (`USRBIN` or `EVENTBIN`) indicated by the user
- `ROTPRBIN` assigns **rotations/translations** for a given user-defined binning (`USRBIN` or `EVENTBIN`) (and sets the **storage precision**, single or double). Useful with `LATTICE`s
- `USERDUMP` defines the events to be written onto a **"collision tape" file**
- `AUXSCORE` defines **filters** and **conversion coefficients**
- `RESNUCLEi` scores **stopping nuclei in a given region**
- `DCYSCORE` assigns cooling times

} *See lecture on activation*

Standard post-processing programs

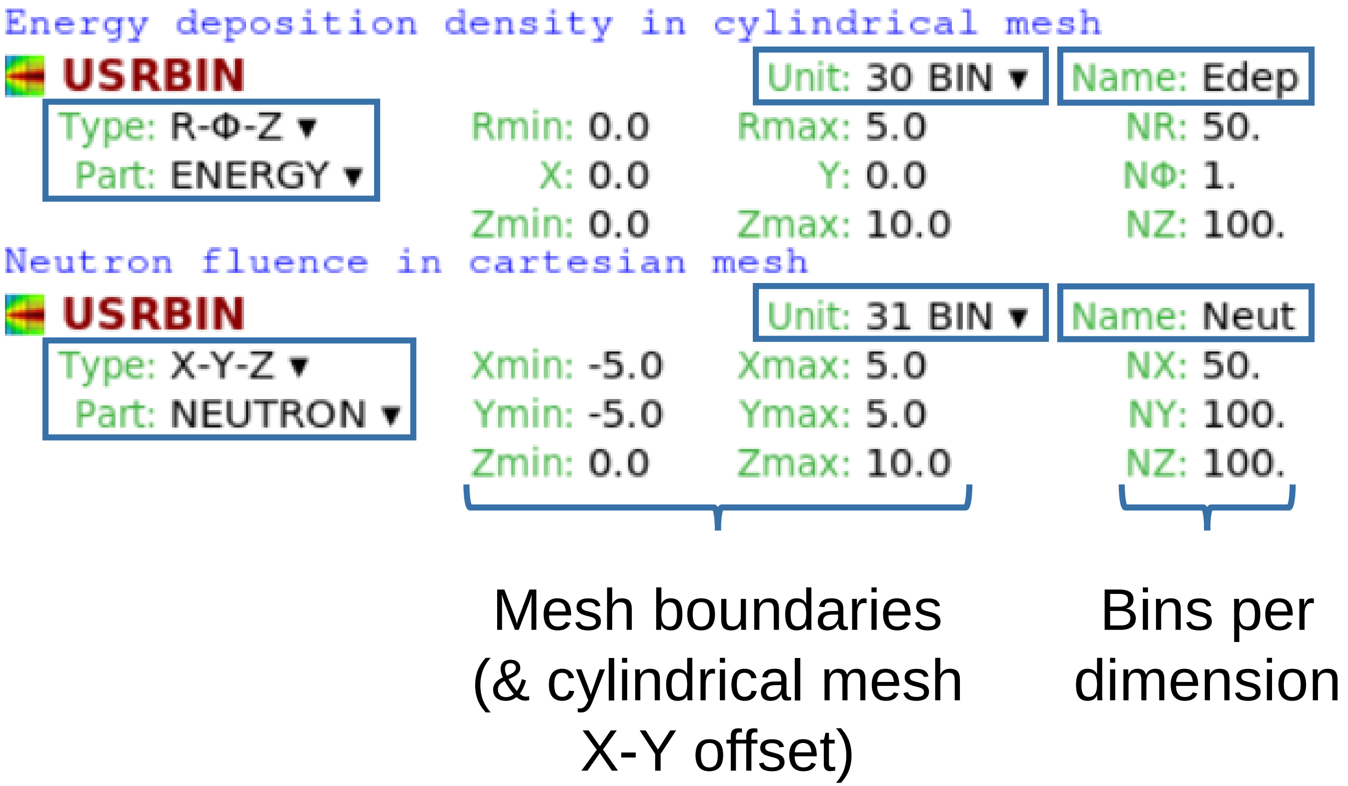
- To analyse the results of the different scoring options, several programs are made available
- **Behind the scenes, Flair uses these programs**

- The executables are in `/pathtofluka/bin`, while the sources are available in `/pathtofluka/src/tools` in case modifications are needed
- They assume that the estimator files are **unformatted**, and can calculate standard deviations and average values over many cycles:
  - `ustsuw.f` to analyze `USRTRACK` and `USRCOLL` outputs
  - `usxsuw.f` to analyze `USRBDX` outputs
  - `usysuw.f` to analyze `USRYIELD` outputs
  - `usbsuw.f` to analyze `USRBIN` outputs
  - `usrsuw.f` to analyze `RESNUCLEi` outputs
  - `usbrea.f` to convert `USRBIN` outputs to ASCII file

`USRBIN` examples, visualisation and plotting

USRBIN definition - examples

- **Type**: X-Y-Z, R-Φ-Z, Region, …
- **Part**: generalised particle
- **Unit**: logical output unit
  - **BIN** (binary): unformatted output
    Can be post-processed via Flair
  - **ASC** (ASCII): formatted output
    Cannot be post-processed via Flair
- **Name**: 8-character limit



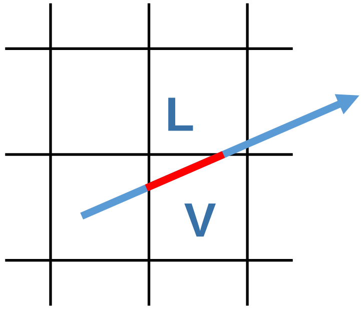
USRBIN definition - examples

Energy deposition density
(GeV/cm^3 per primary)

```
Energy deposition density in cylindrical mesh
USRBIN                         Unit: 30 BIN ▾   Name: Edep
  Type: R-Φ-Z ▾      Rmin: 0.0   Rmax: 5.0      NR: 50.
  Part: ENERGY ▾        X: 0.0      Y: 0.0      NΦ: 1.
                     Zmin: 0.0   Zmax: 10.0     NZ: 100.
```

Particle fluence
(1/cm^2 per primary)

```
Neutron fluence in cartesian mesh
USRBIN                         Unit: 31 BIN ▾   Name: Neut
  Type: X-Y-Z ▾      Xmin: -5.0  Xmax: 5.0      NX: 50.
  Part: NEUTRON ▾    Ymin: -5.0  Ymax: 5.0      NY: 100.
                     Zmin: 0.0   Zmax: 10.0     NZ: 100.
```



Path length **L** [cm] inside the bin divided by the bin volume **V** [cm^3] → cm / cm^3 = 1/cm^2

Volume normalisation

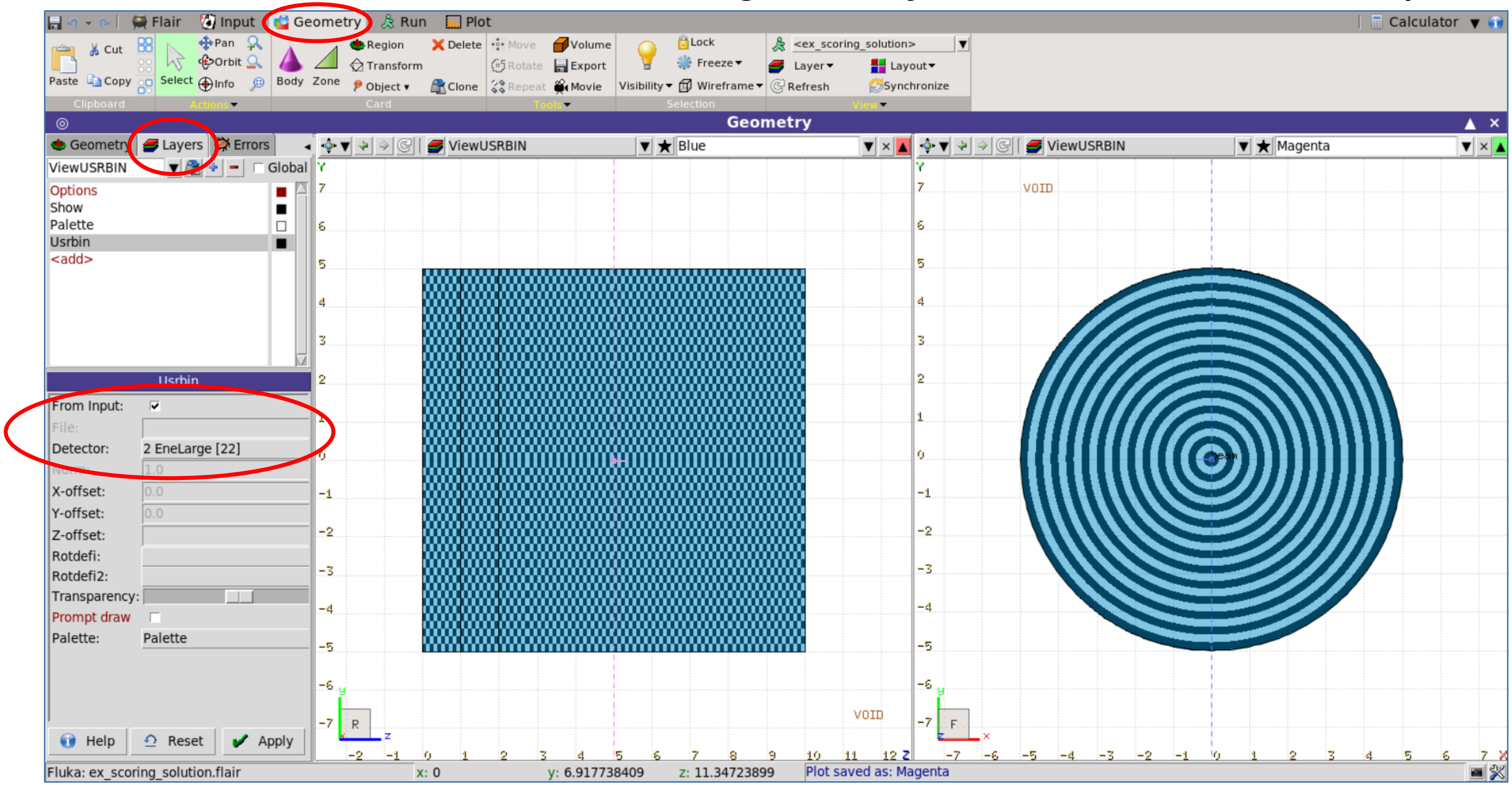
- FLUKA does not calculate **region** volumes

- When scoring particle *fluence* (e.g. **`NEUTRON`**) *or energy density* (**`ENERGY`**) with **`USRBIN` by region**, the actual results will give instead *total track-length and energy deposition*, respectively; these differ from the intended quantities by a factor equal to the region volume
  - In other words, the code assumes a volume equal to 1 cm^3

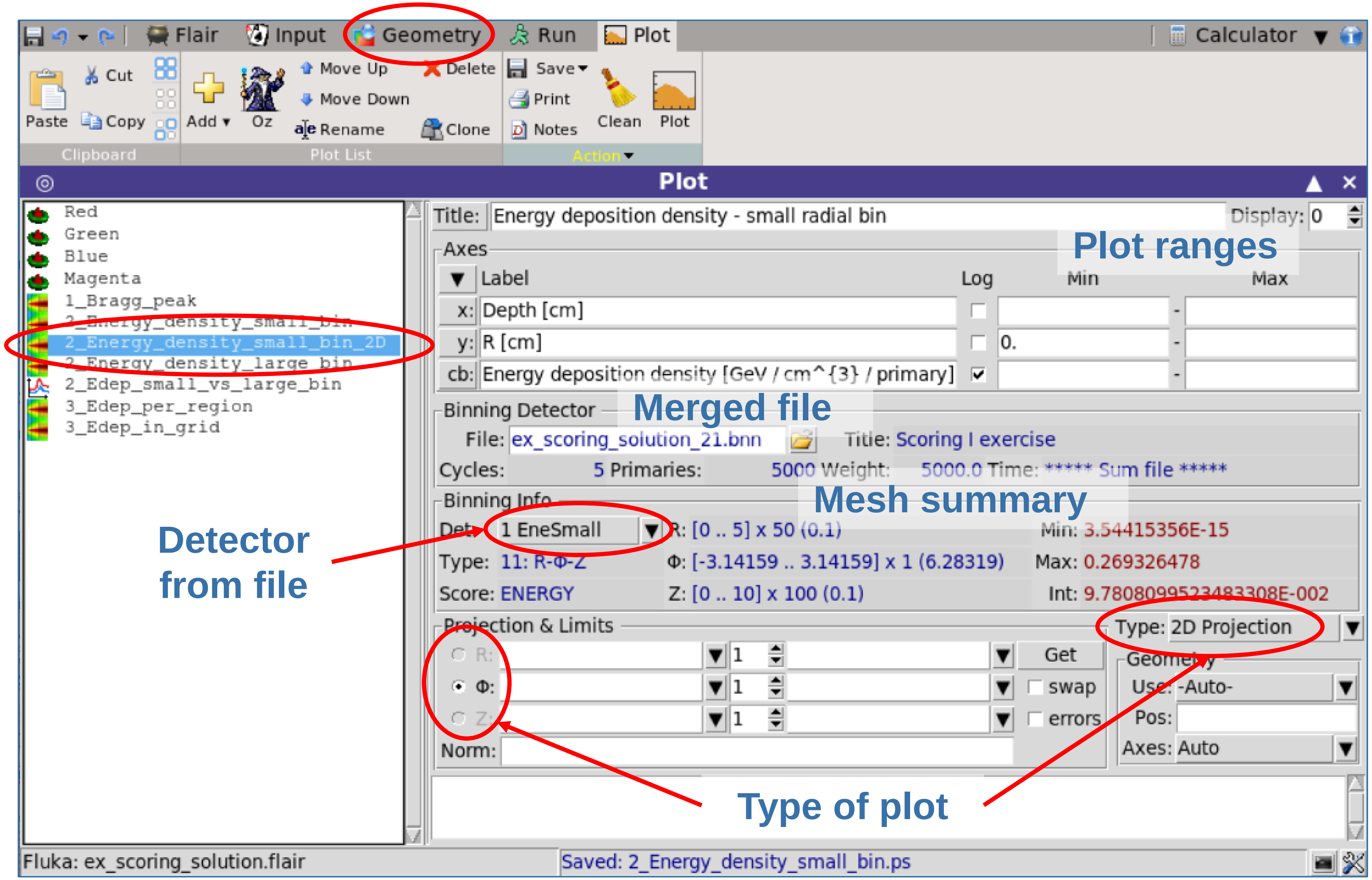
- Conversely, as **`USRBIN`** scoring **on regular (Cartesian, cylindrical) grids** is requested, *particle fluence and energy density* will be automatically provided, since FLUKA performs the bin volume normalisation

Visualisation

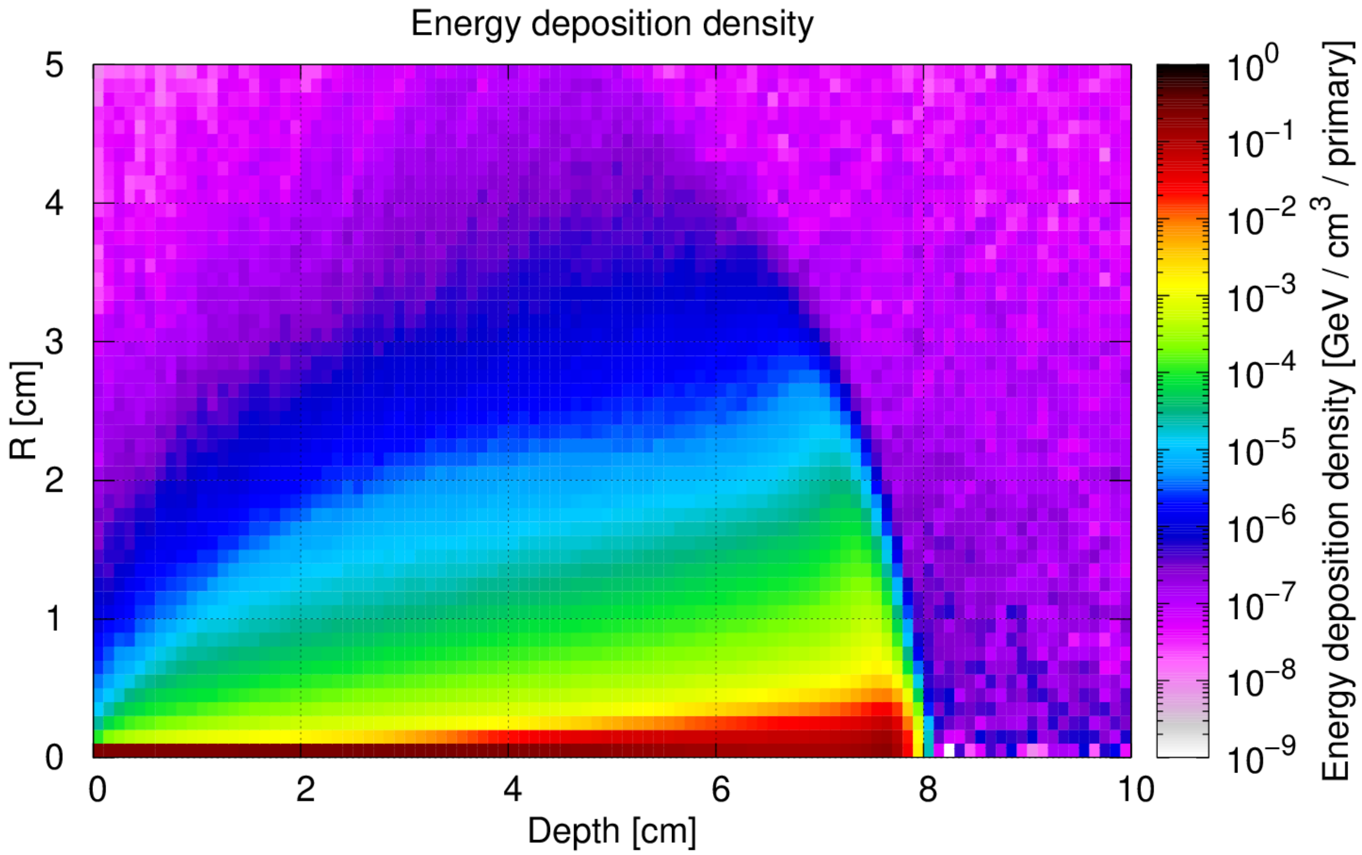
- A defined mesh can be overlaid on the geometry to check that it is well-positioned



Plotting example – energy deposition density



Plotting example – energy deposition density



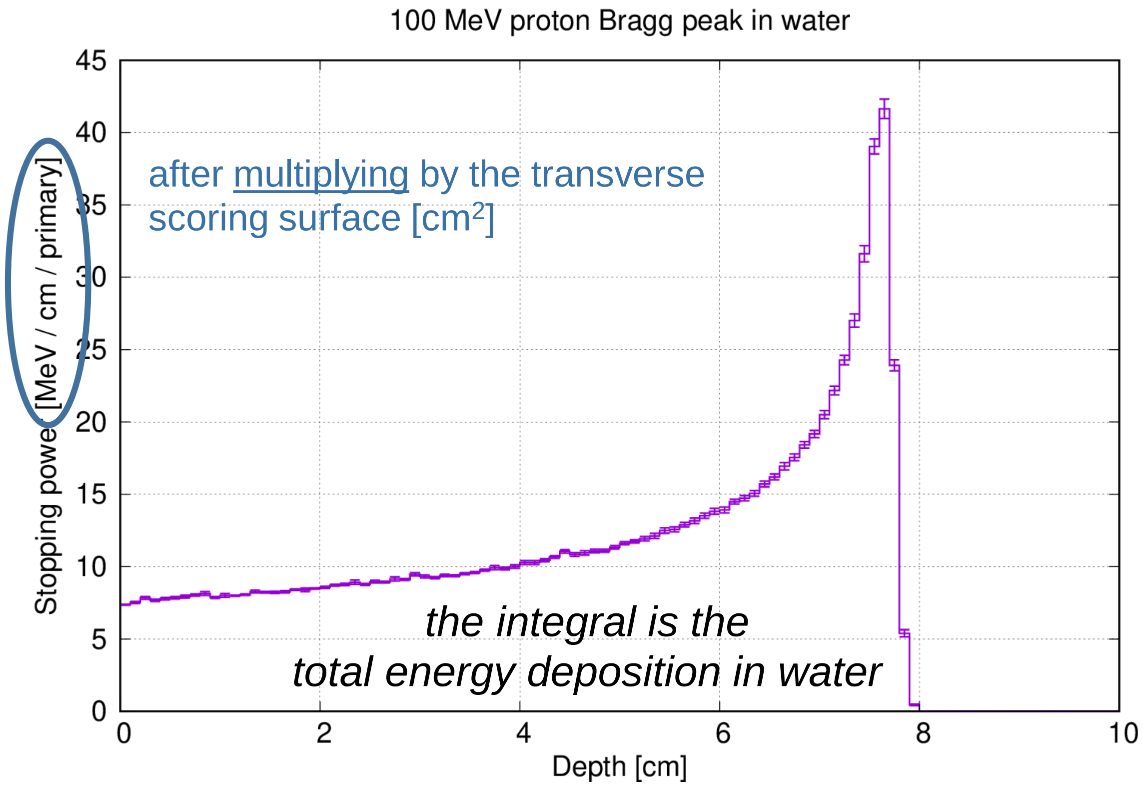
- **N.B.:** This plot is a **2D projection** of a 3D structure → **the result is the average over the 3rd coordinate** (Φ in the above case)
- Projection limits can be set in Flair

One dimensional plots

- 1D plots can be obtained from 3D **USRBIN** meshes, e.g. (for an R-Φ-Z mesh):

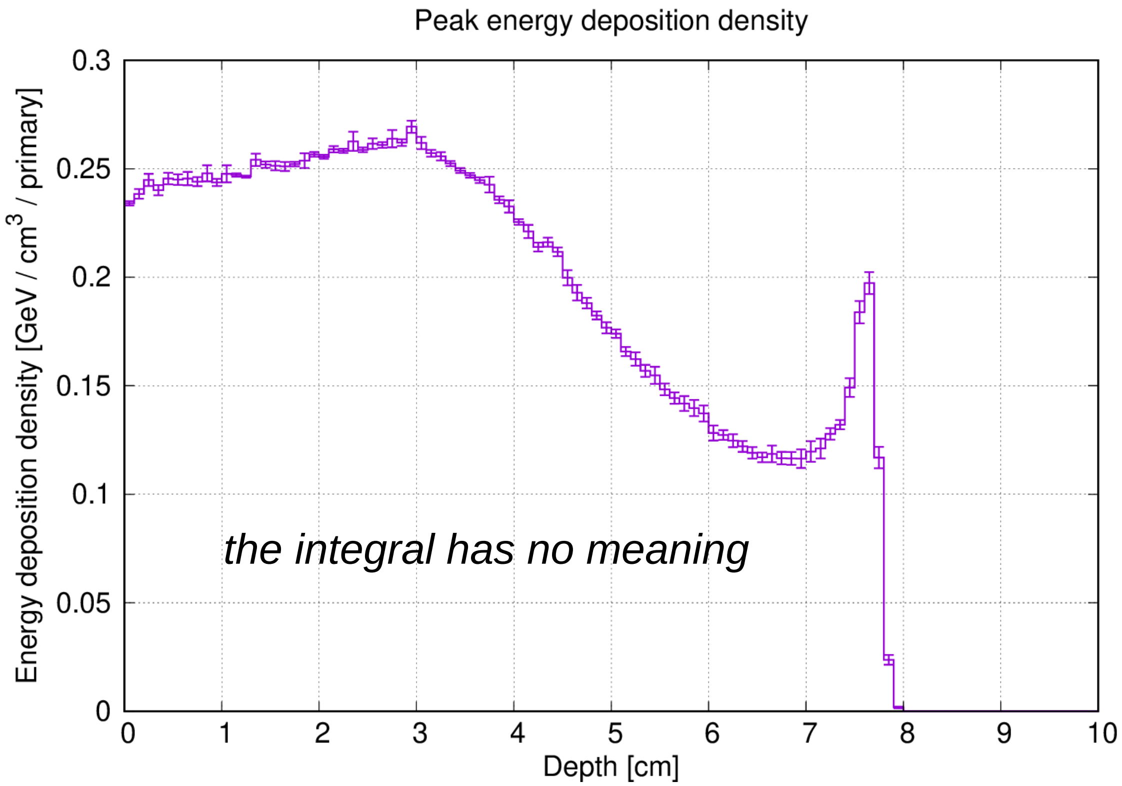
1D Projection

- The average of the values in each z-bin (i.e. over all R, Φ)



1D Max

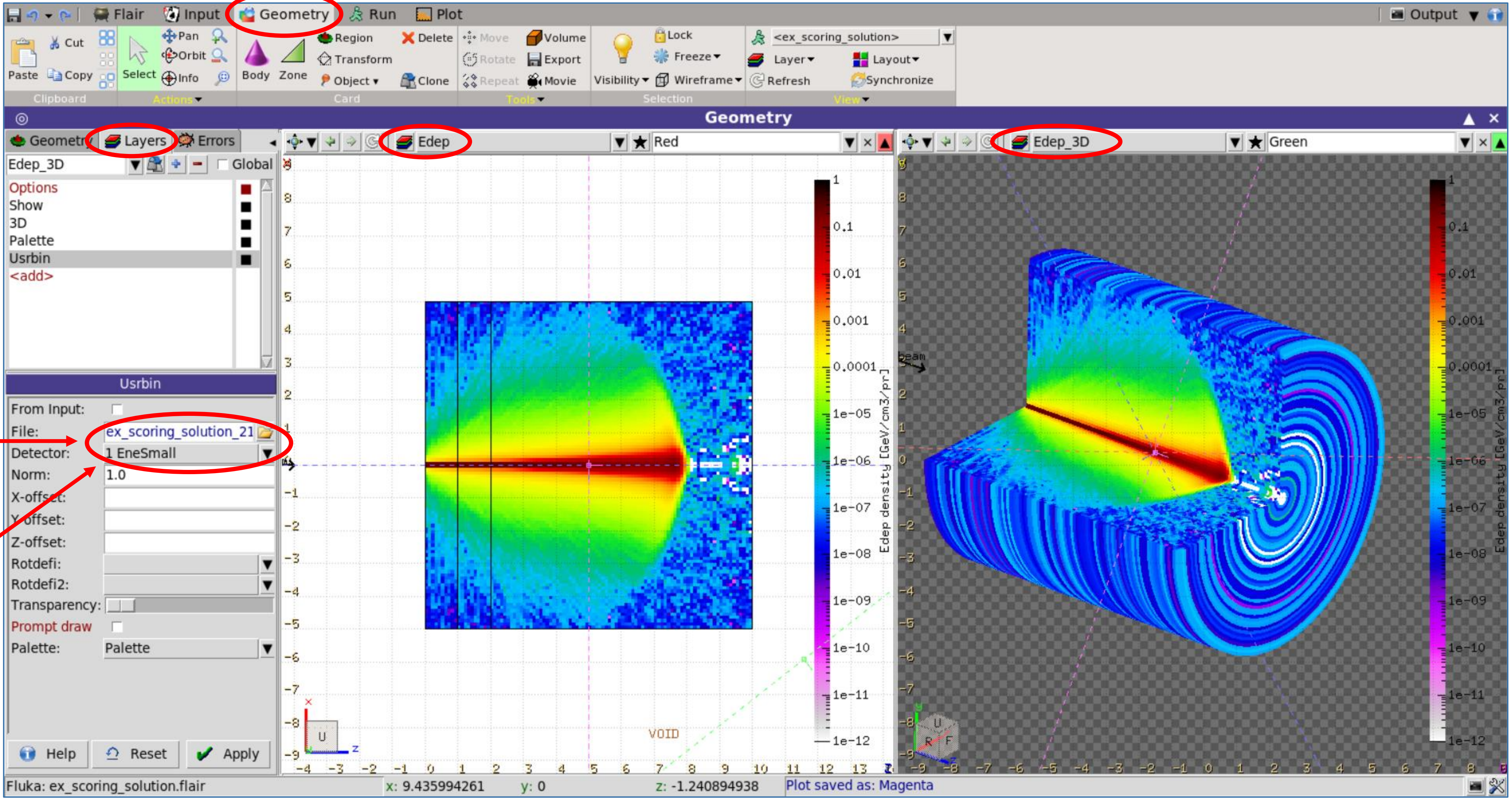
- The highest value in each z-bin



Overlaying `USRBIN` mesh results on 2D/3D geometry

Merged file

Detector from file



FLUKA