FLUKA

Scoring physics quantities [I]

Introduction to built-in estimators

Fluence, a misunderstood fundamental quantity

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

FLUKA Scoring & Results - Estimators

- 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 be calculated using different kinds of estimators**

- FLUKA offers **numerous different estimators**, *i.e.* directly from the input file the users can request scoring for various quantities of interest

- As **built-in estimators** are implemented in a correct and complete manner, users are strongly encouraged to **use them** with respect to user-defined scorings

- For additional requirements, **FLUKA user routines** are provided (not covered here)

Built-In and User Scoring

- Several **pre-defined estimators** can be activated in FLUKA.
- One usually refers to these estimators as **“scoring”** capabilities
- Users have also the possibility to build their own scoring through **user routines.** However,
  - **Built-in scoring** covers most of the **common needs**
  - **Built-in scoring** has been **extensively tested**
  - **Built-in scoring** takes **biasing weights automatically into account**
  - **Built-in scoring** has **refined algorithms** for track subdivision (apportioning)
  - **Built-in scoring** comes with **utility programs** that allow to evaluate statistical errors
- Standard scoring can be adapted by means of simple **user routines (`fluscw.f`, `comscw.f`)** to be activated via **`USERWEIG`** card

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 number of atoms contained in a cylinder with a 1 cm^2 base)

- Microscopic and macroscopic cross sections 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 both 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

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

- Fluence is expressed in **particles per cm²** but in reality it represents the **density of particle tracks [cm / cm³]**

Fluence estimation

- **Track length** estimation **(USRTRACK)**:

$$\dot{\Phi}(v)\,dt = n(v)\,v\,dt = \frac{dN(v)}{dV}\frac{dl(v)}{dt}dt = \lim_{\Delta V \to 0}\frac{\sum_i l_i(v)}{\Delta V}$$

- **Collision density** estimation **(USRCOLL)**:

$$\dot{\Phi}(v) = \frac{\dot{R}(v)}{\sigma(v)\,N_o \Delta V} = \frac{\dot{R}(v)}{\Sigma(v)\Delta V} = \frac{\dot{R}(v)\lambda(v)}{\Delta V}$$

in vacuum (any low density material)
it does not work (well)

FLUKA Scoring & Results

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

Where? In regions, across boundaries, on region-independent grids

When?

At the end of each cycle

At each event

Output? Files `inpnnn_fort.##` where `##` is logical unit number chosen by user

Results?

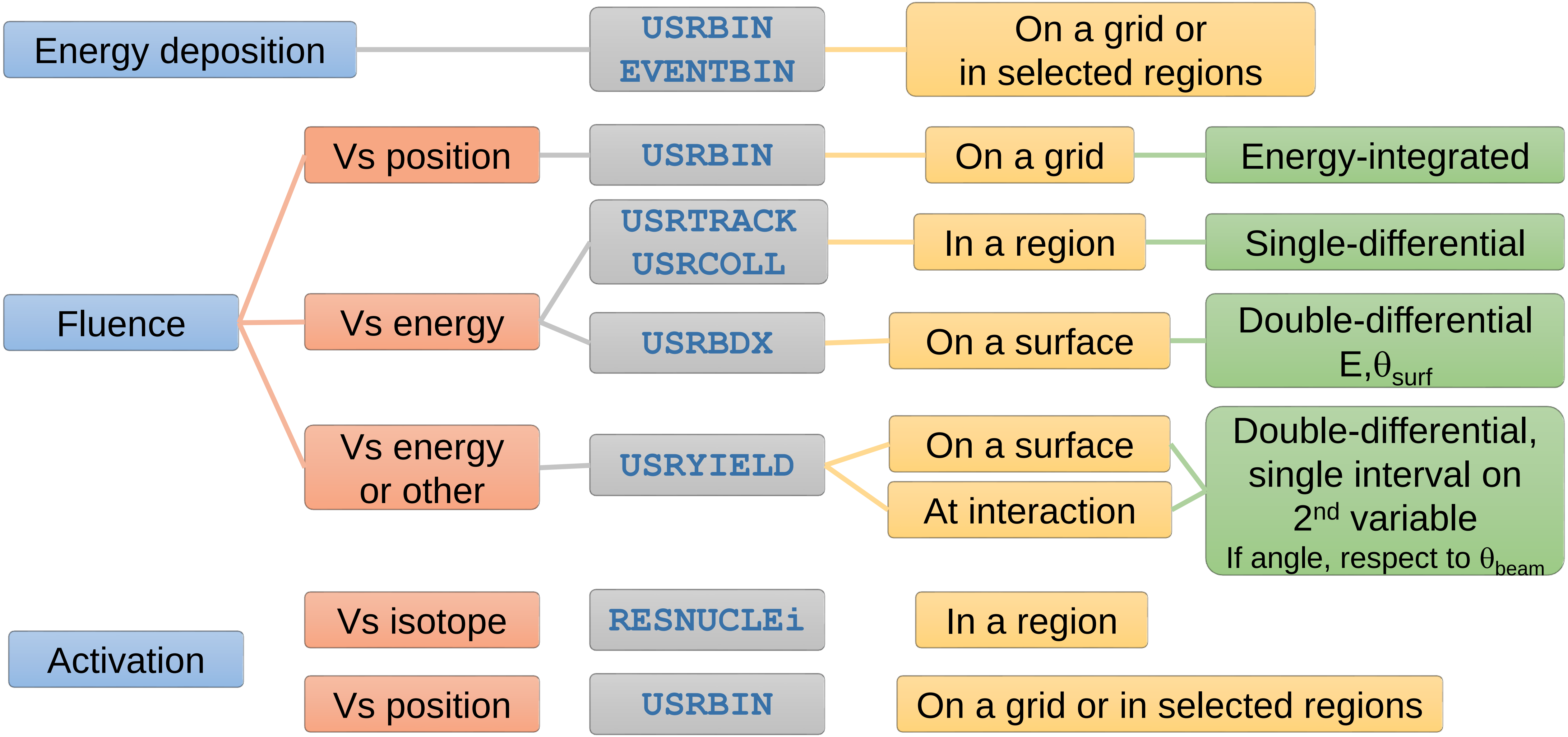
Post-processing utilities (in `/pathtofluka/bin`) merge cycles, calculate average and rms, provide data files for plotting

User code needed

Data merging and plotting available in **Flair**

Results normalized **per primary**

FLUKA estimator zoo



Main FLUKA estimators

- **USRBIN** scores the **spatial distribution** of **energy density** or **fluence** (or star density) in a **regular mesh** (cylindrical, Cartesian, or by region) 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

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

Standard Postprocessing Programs

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

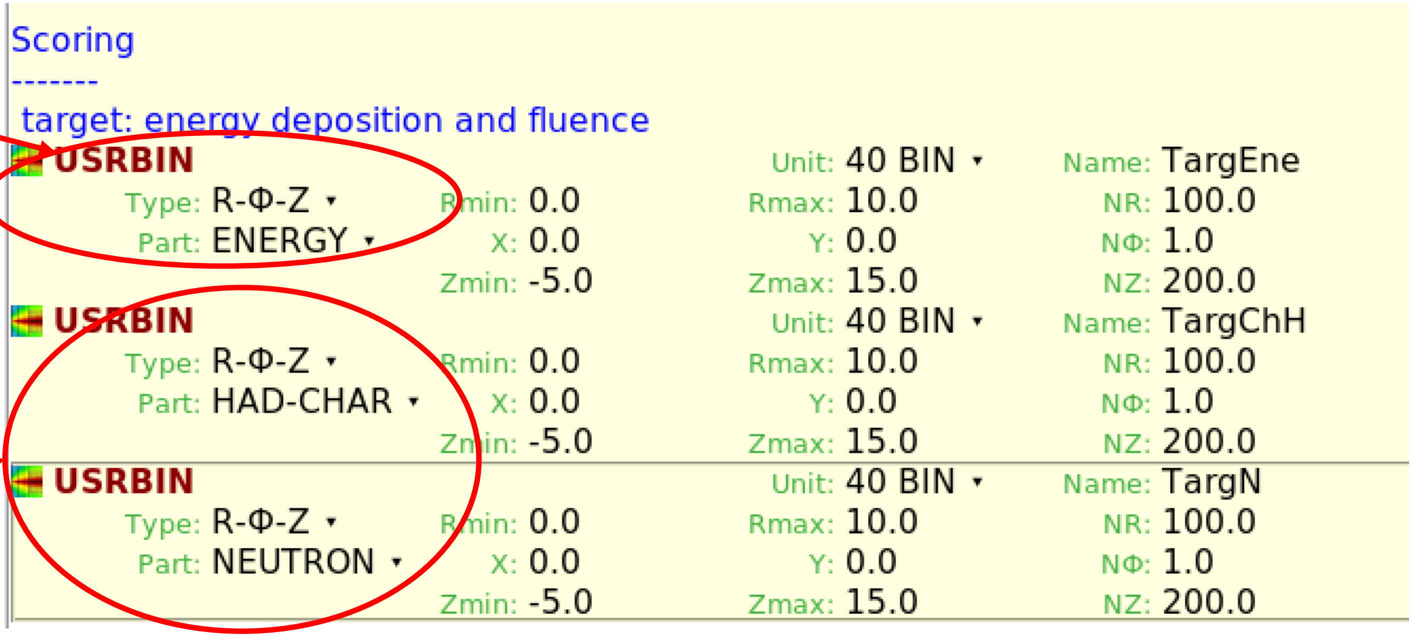
Volume normalization

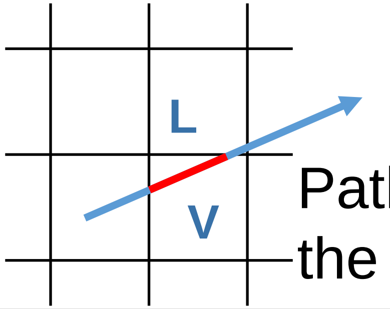
- FLUKA does not calculate **region** volumes.
- As scoring particle *fluence* (e.g. NEUTRON) *or energy density* (ENERGY) with **USRBIN by region**, the actual results will give instead *tracklength and total energy deposition*, respectively, which differ from the intended quantities by the region volume.
- 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 normalization.

USRBIN scoring definition (3D mesh)

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

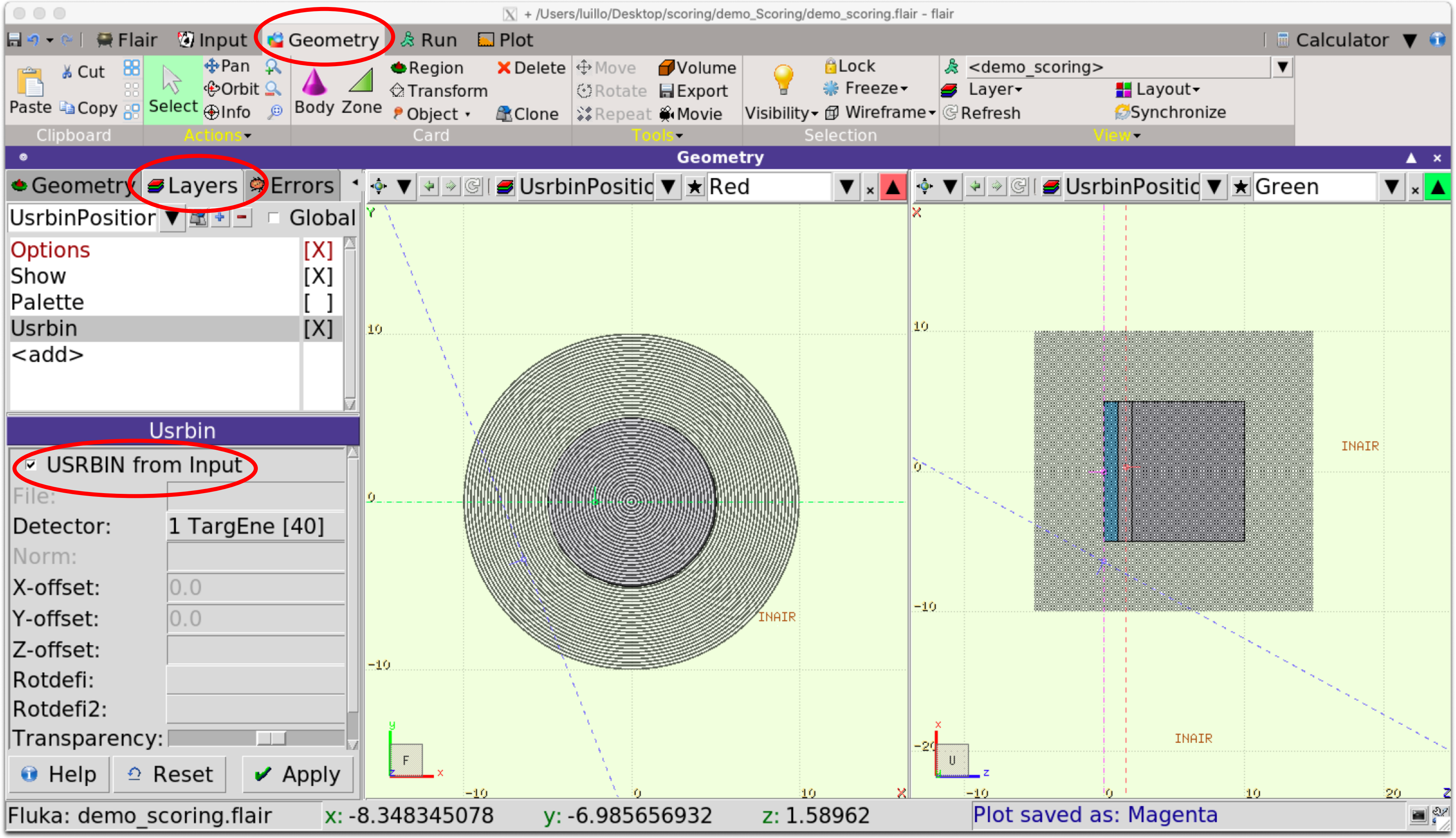
Particle fluence
(1/cm^2 per primary)



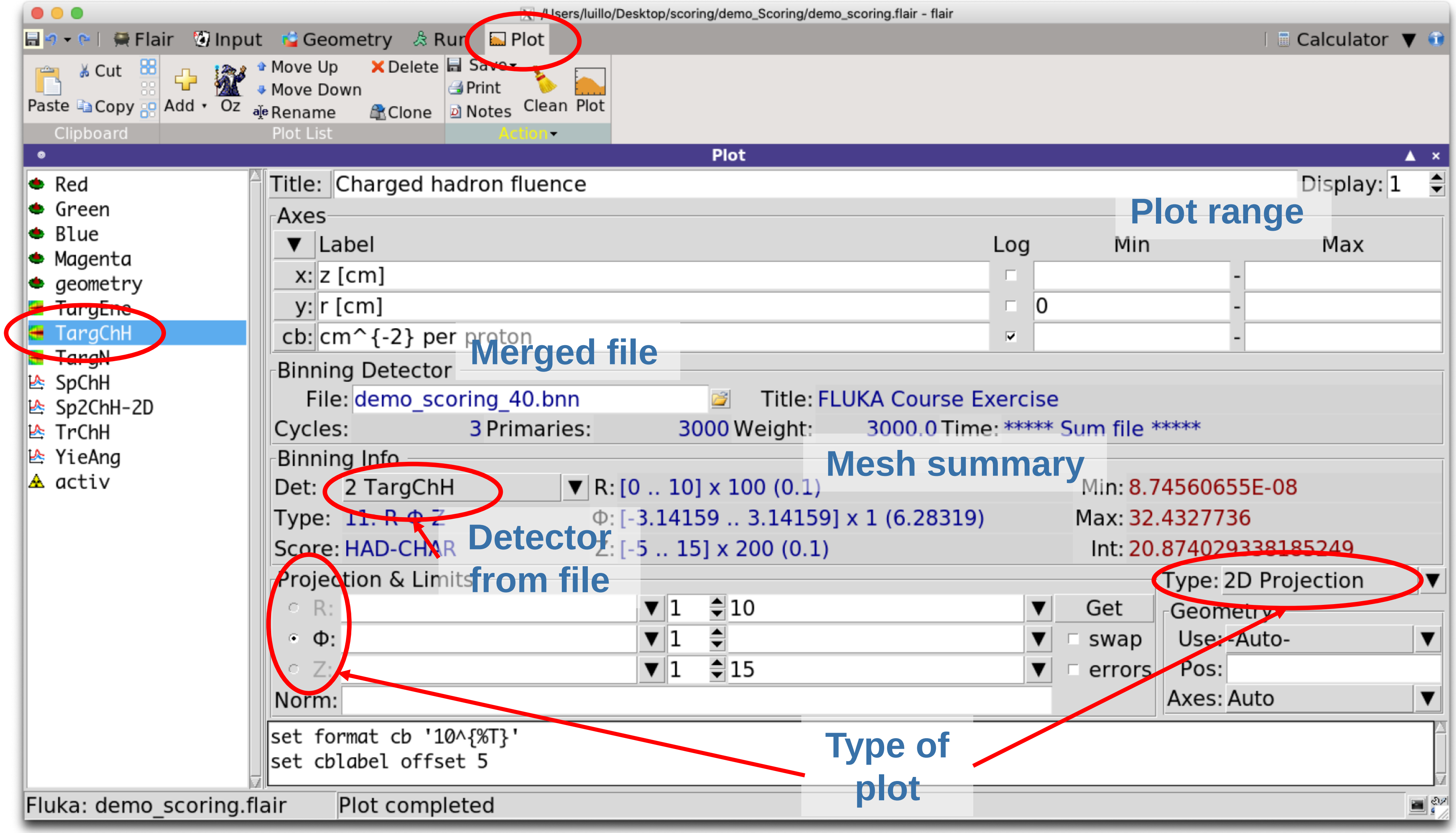


Path length **L** [cm] inside the bin divided by the bin volume **V** [cm^3] → [cm]*[1/cm^3] = [1/cm^2]

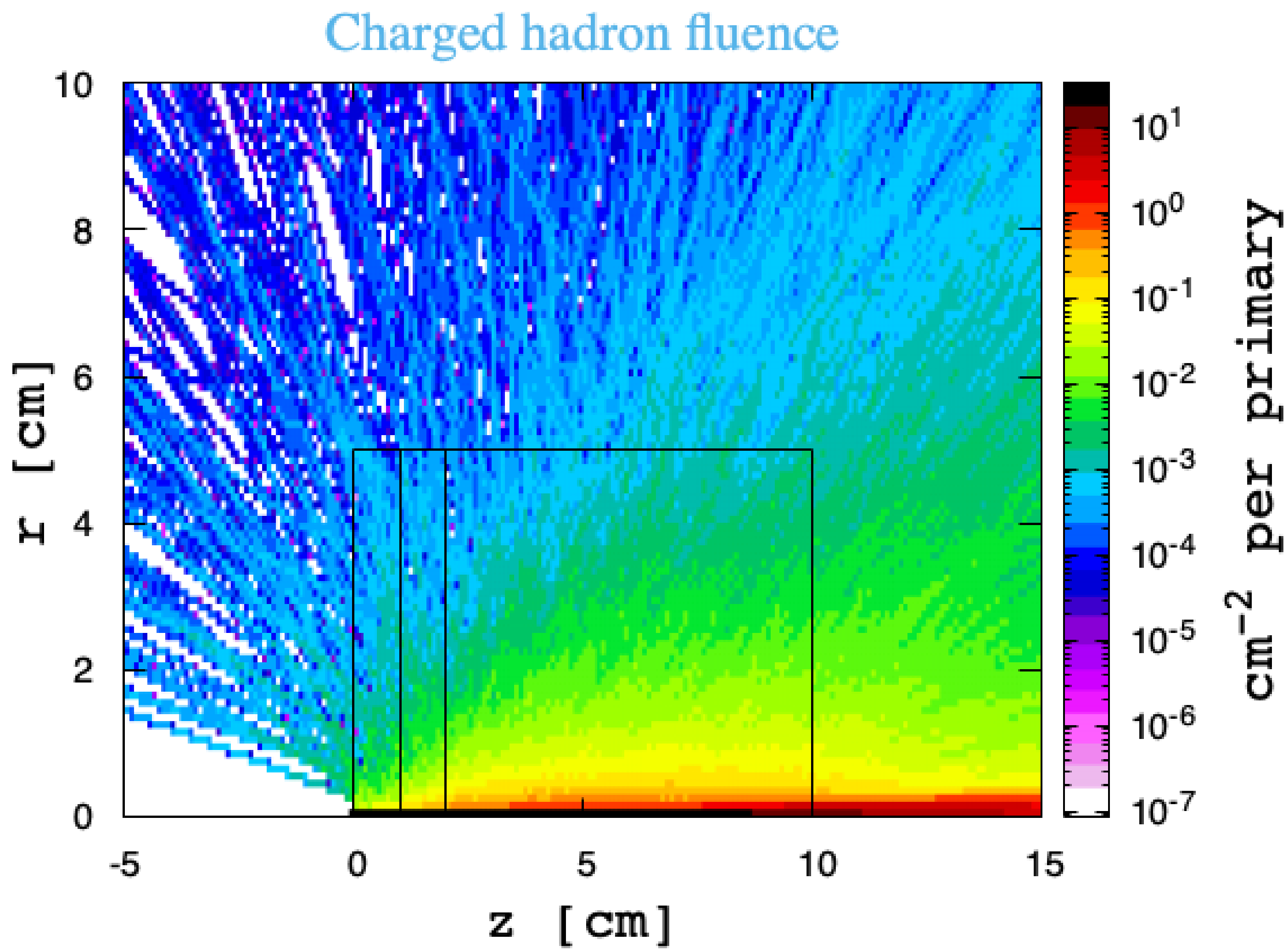
USRBIN scoring check (3D mesh)



Plotting – charged hadron fluence (USRBIN mesh)

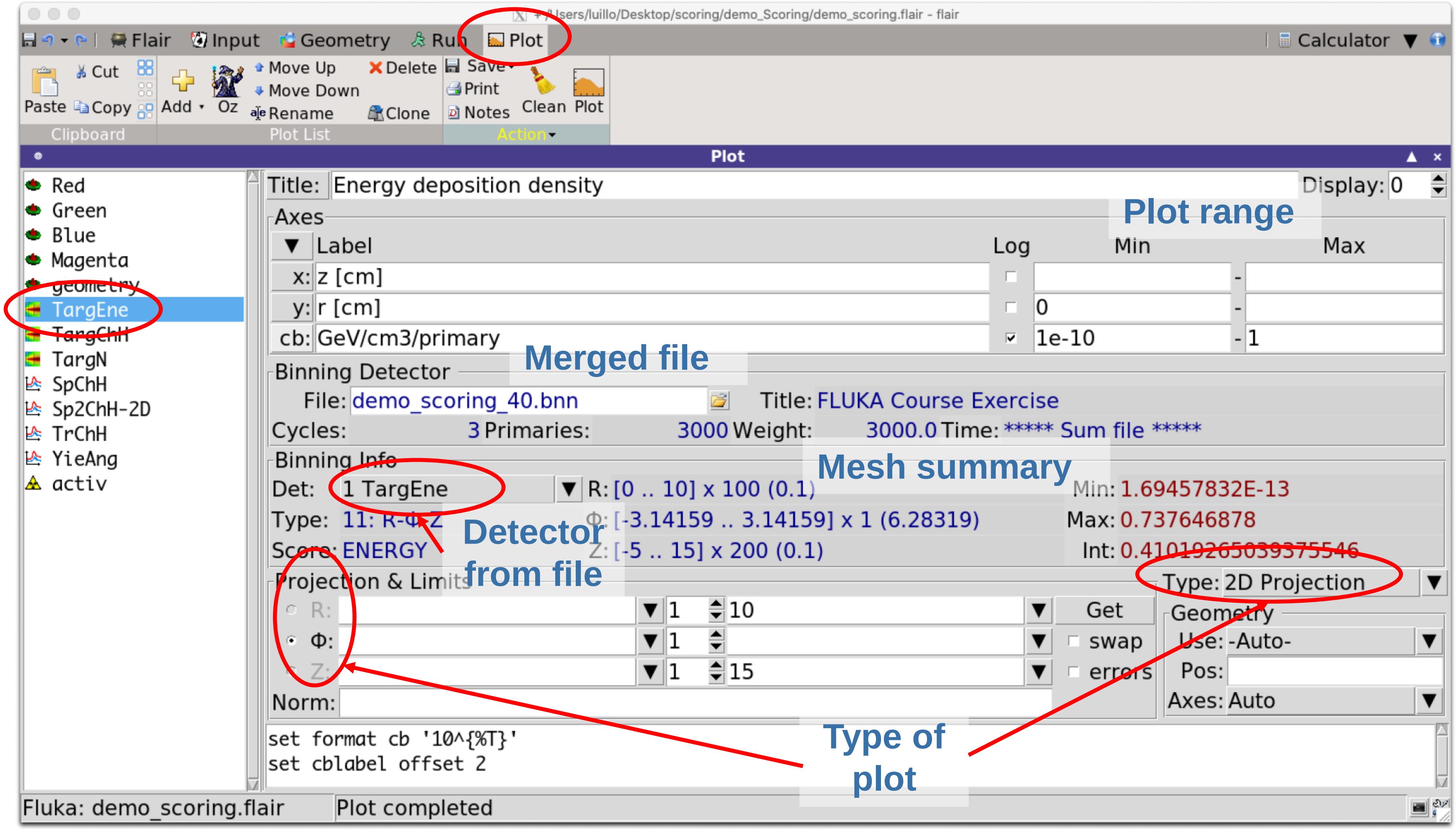


Plot result – charged hadron fluence (`USRBIN` mesh)

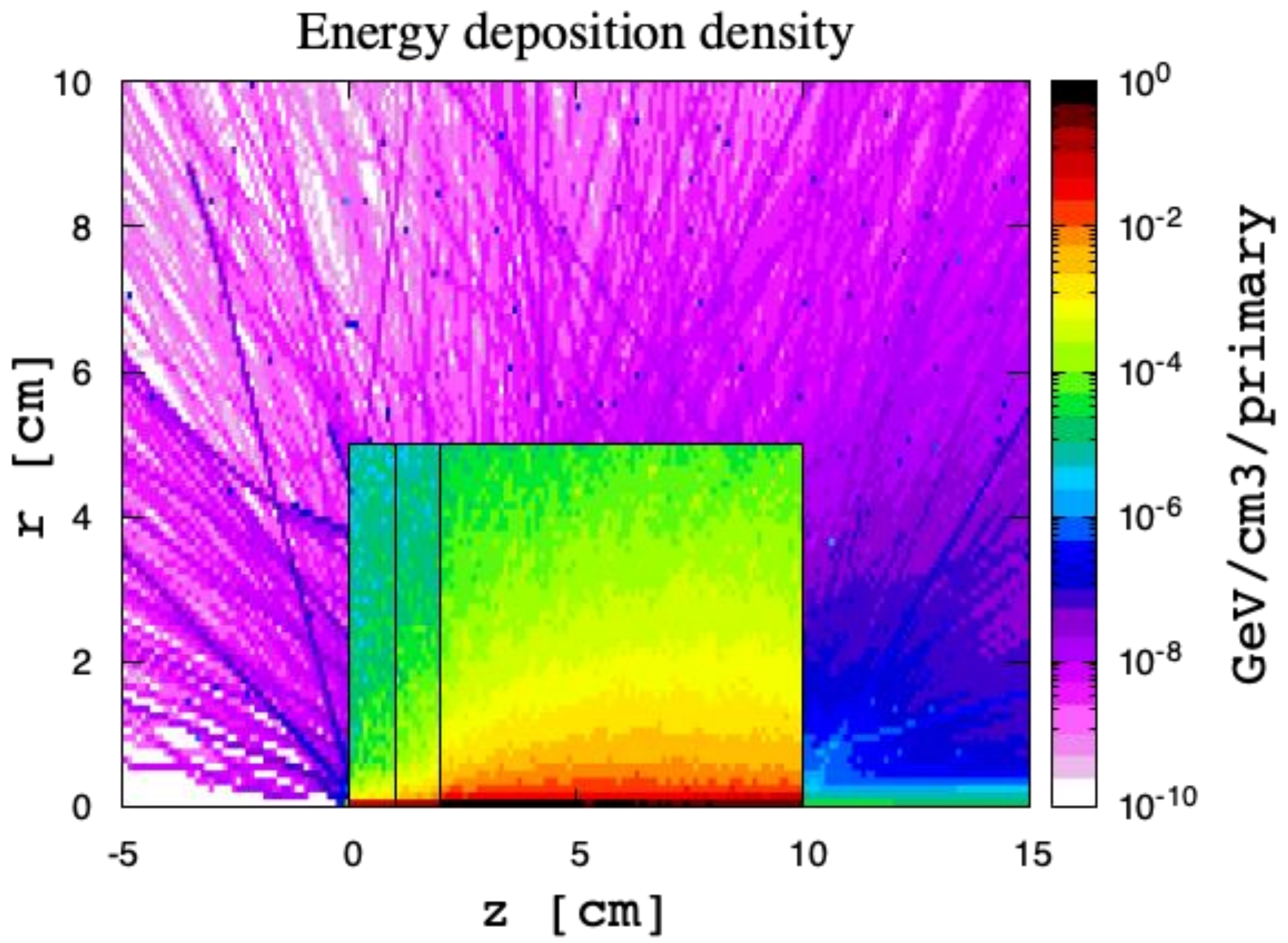


This plot is a **2D projection** of a 3D structure → **the result is the** average **over the 3rd coordinate**
Projection limits can be set in Flair

Plotting – energy deposition density (USRBIN mesh)



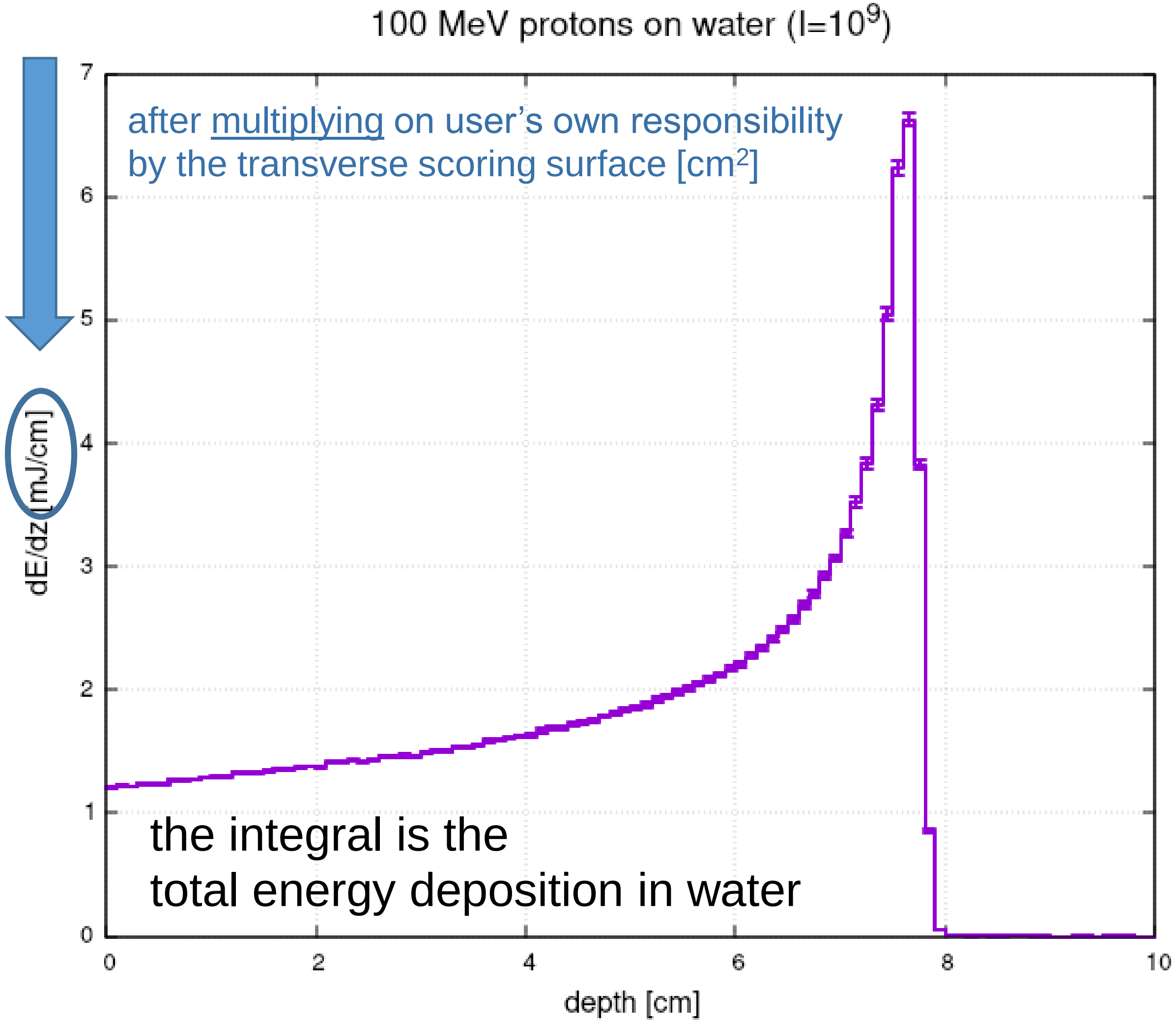
Plot result – energy deposition density (`USRBIN` mesh)



This plot is a **2D projection** of a 3D structure → **the result is the** average **over the 3rd coordinate**
Projection limits can be set in Flair

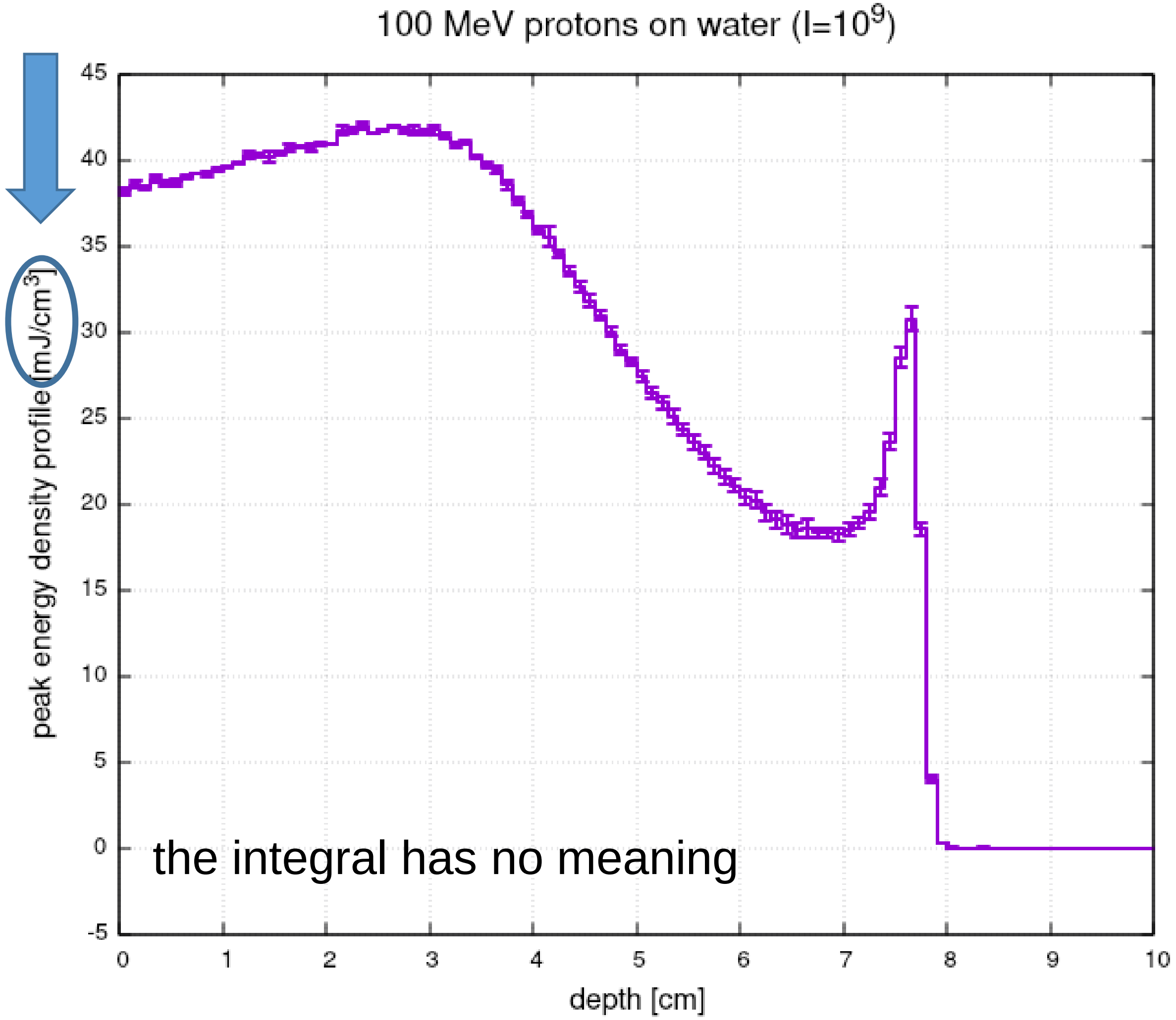
One dimensional plots!

1D Projection

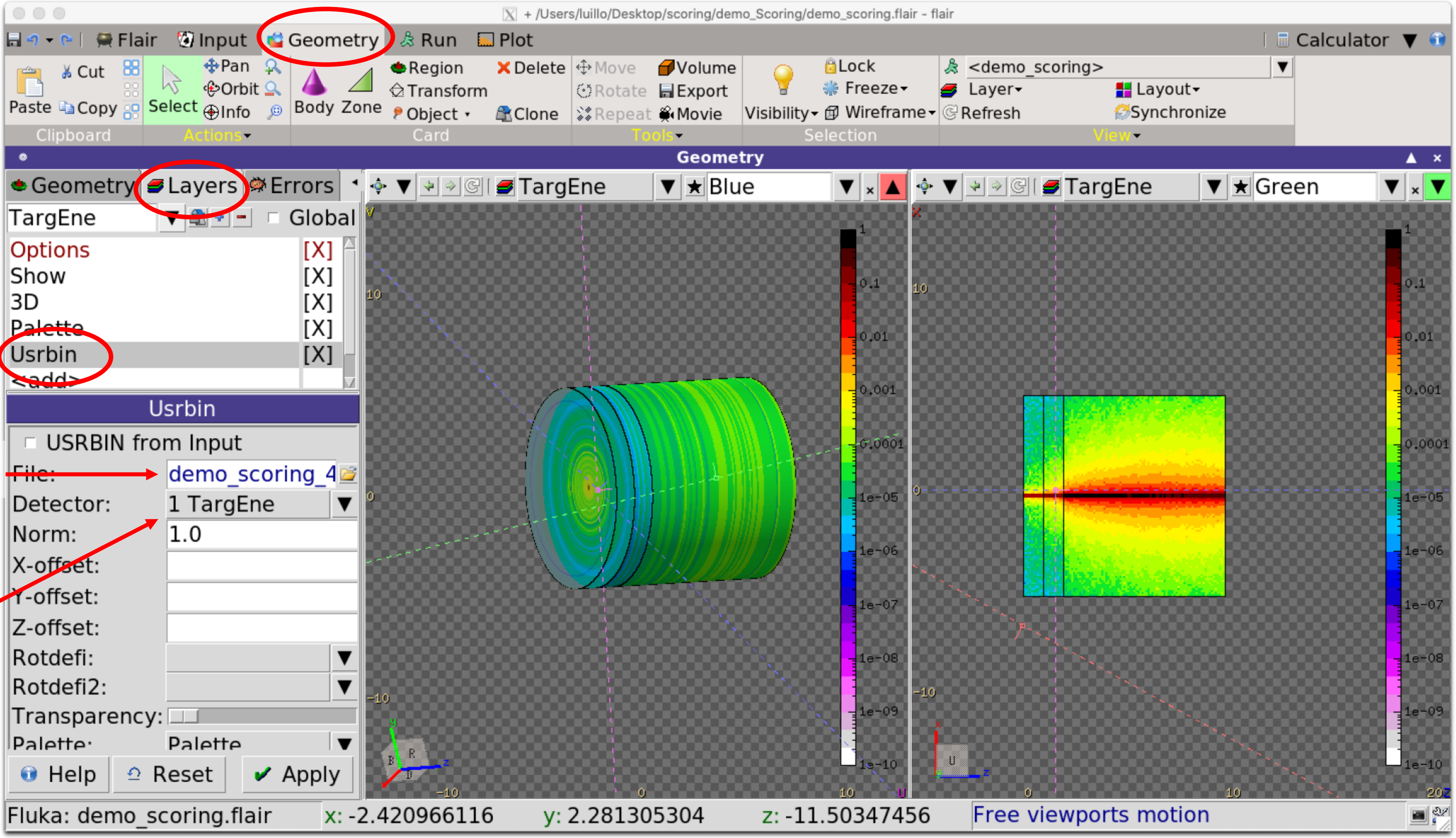


1D Max

for each z-bin, the highest value on the transverse scoring surface



Superimposing USRBIN on 3D geometry



Merged file

Select detector from file

FLUKA