



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.
- $\sigma = \frac{\Sigma}{N_0} = \text{atom effective area}$, [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/cm²**
- 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)

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

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

- $\Phi(\mathbf{r}, v) = n(\mathbf{r}, v) dl$, $[cm^{-2}]$: **fluence**, *i.e.* time integral of the flux density

- Fluence is expressed in **particles per cm^2** but in reality represents the **density of particle tracks $[cm / cm^3]$!**

- $\dot{\Phi}(\mathbf{r}, v) = n(\mathbf{r}, v) v$, $[cm^{-3} cm s^{-1}] = [cm^{-2} s^{-1}]$: **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

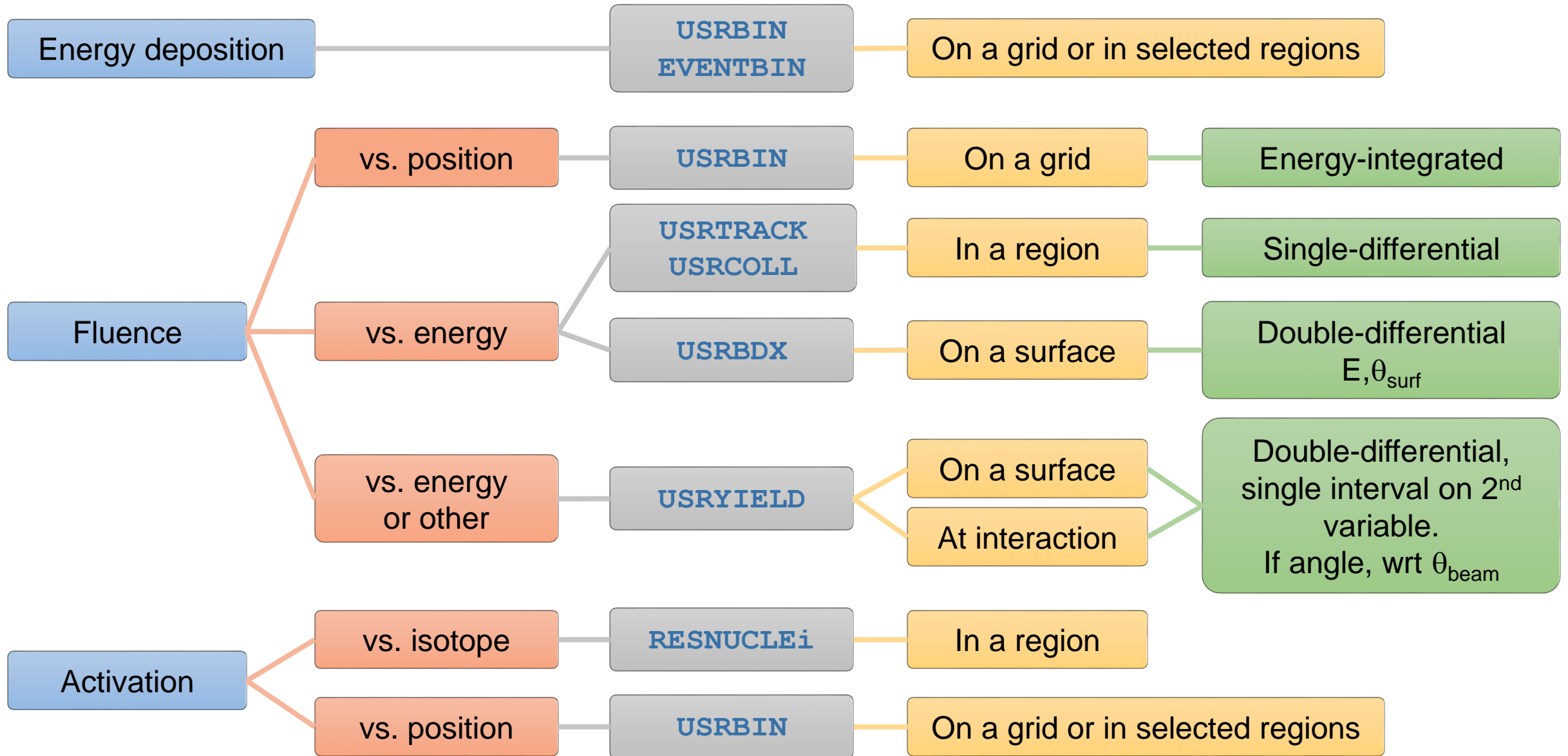
Results?

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 **selection of regions** or in a **regular mesh** (cylindrical, cartesian) described by the user.
- **USRTRACK** (**USRCOLL**) scores average **$d\Phi/dE$ (differential fluence)** of a given type or family of particles in a **given region**.
- **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**.
- **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**
 - Unit numbers must be >20.
 - The only exception is SCORE: its output is printed in the **standard output**.

* A star is a hadronic inelastic interaction

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
- **USERDUMP** allows the user to **dump selected information within each primary history**
- **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 **LATTICES**
- **AUXSCORE** defines **filters** and **conversion coefficients**
- **RESNUCLEi** scores **stopping nuclei in a given region**
- **DCYSCORE** assigns cooling times

} *See radiation protection lecture*

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

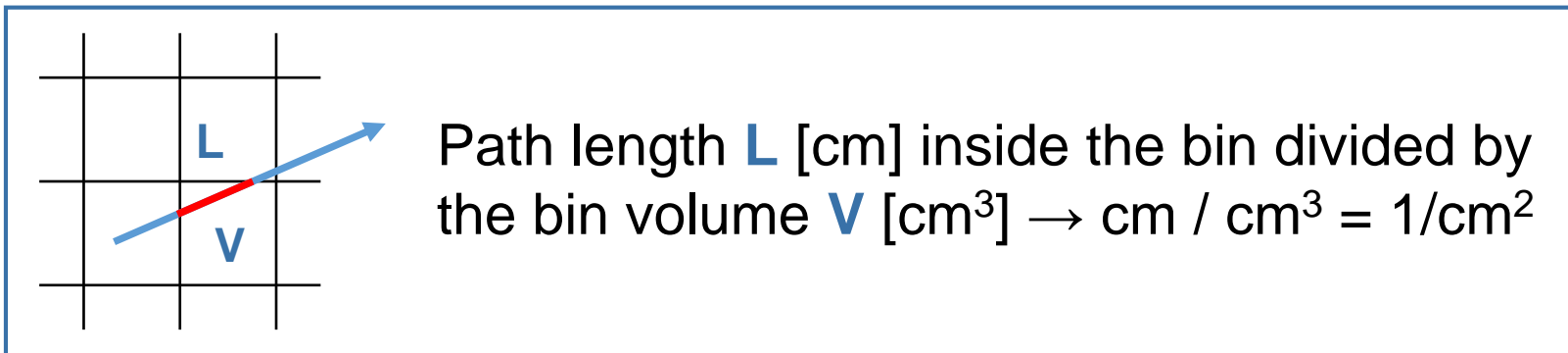
USRBIN: Input, visualisation & plotting

USRBIN scoring definition

Energy deposition density
(GeV/cm³ per primary)

Particle fluence
(1/cm² per primary)

Energy deposition density in cylindrical mesh					
 USRBIN	Type: R-Φ-Z ▼	Rmin: 0.0	Unit: 21 BIN ▼	Name: Edep	NR: 50.
	Part: ENERGY ▼	X: 0.0	Rmax: 5.0		NΦ: 1.
		Zmin: 0.0	Y: 0.0		NZ: 100.
			Zmax: 10.0		
Charged hadron fluence					
 USRBIN	Type: R-Φ-Z ▼	Rmin: 0.0	Unit: 31 BIN ▼	Name: ChHad	NR: 50.
	Part: HAD-CHAR ▼	X: 0.0	Rmax: 5.0		NΦ: 1.
		Zmin: 0.0	Y: 0.0		NZ: 100.
			Zmax: 10.0		
Neutron fluence					
 USRBIN	Type: R-Φ-Z ▼	Rmin: 0.0	Unit: 32 BIN ▼	Name: Neut	NR: 50.
	Part: NEUTRON ▼	X: 0.0	Rmax: 5.0		NΦ: 1.
		Zmin: 0.0	Y: 0.0		NZ: 100.
			Zmax: 10.0		



USRBIN input card

- **Type:**
 - * X-Y-Z: cartesian mesh
 - * R- Φ -Z: cylindrical mesh
 - * per region
 - * ...
- **Part:** generalised particle
- **Unit:** logical output unit
 - **BIN** (binary): unformatted output
Can be converted to ASCII
or directly post-processed via Flair.
 - **ASC** (ASCII): formatted output
Cannot be post-processed via Flair.
- **Name:** 8-character limit

The screenshot shows three USRBIN input cards for different particle types and units. Each card is for a cylindrical mesh with Rmin: 0.0, Rmax: 5.0, X: 0.0, Y: 0.0, Zmin: 0.0, and Zmax: 10.0. The first card is for Energy deposition density (Edep) with Unit: 21 BIN, NR: 50, NPhi: 1, and NZ: 100. The second card is for Charged hadron fluence (ChHad) with Unit: 31 BIN, NR: 50, NPhi: 1, and NZ: 100. The third card is for Neutron fluence (Neut) with Unit: 32 BIN, NR: 50, NPhi: 1, and NZ: 100. All cards have Type: R- Φ -Z and Part: ENERGY, HAD-CHAR, and NEUTRON respectively.

Mesh boundaries

Bins per dimension

Note on "Type":

Do not choose the "point" versions, which do not return e.g. fluence when a particle is selected, but density of stars produced by the particles.

Convert USRBIN bnn file to ASCII (3D mesh data)

- Debugging
- Visualization with external tools (e.g. Matlab)

Run	Spawn	Cycles	File	Type	Size
<ex_scoring_solution>	2	compile	ex_scoring_solution_21.bnn	bnn	40268
ex_scoring_solutior		data	ex_scoring_solution_22.bnn	bnn	20268
ex_scoring_solutior		input	ex_scoring_solution_23.bnn	bnn	292
		plot	ex_scoring_solution_24.bnn	bnn	348
		run			
		temporary			

Let's convert this .bnn file

Result: bnn.lis file
Always 10 columns per line

```

1 1
2 R - Z binning n. 1 "EneSmall " , generalized particle n. 208
3 R coordinate: from 0.0000E+00 to 5.0000E+00 cm, 50 bins ( 1.0000E-01 cm wide)
4 Z coordinate: from 0.0000E+00 to 1.0000E+01 cm, 100 bins ( 1.0000E-01 cm wide)
5 axis coordinates: X = 0.0000E+00, Y = 0.0000E+00 cm
6 Data follow in a matrix A(ir,iz), format (1(5x,1p,10(1x,e11.4)))
7
8 accurate deposition along the tracks requested
9 2.3867E-01 1.8430E-04 1.6286E-05 9.9585E-06 3.3918E-07 2.1157E-06 7.4462E-06 6.5266E-06 6.3637E-06 4.9691E-06
10 0.0000E+00 0.0000E+00 1.6428E-06 3.8209E-06 4.4210E-06 3.7473E-06 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
11 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 1.1102E-06
12 1.8253E-08 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
13 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
14 2.3794E-01 3.4083E-04 5.7607E-05 2.7670E-05 2.7495E-05 2.0694E-05 0.0000E+00 0.0000E+00 0.0000E+00 2.5324E-06
15 3.5606E-06 3.5531E-06 1.8965E-06 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
16 0.0000E+00 0.0000E+00 2.7077E-09 0.0000E+00 0.0000E+00 0.0000E+00 6.8063E-08 0.0000E+00 0.0000E+00 0.0000E+00
17 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 6.1228E-08 0.0000E+00 0.0000E+00 0.0000E+00 2.7059E-07 0.0000E+00
18 0.0000E+00 0.0000E+00 4.4389E-11 0.0000E+00 1.6188E-10 2.2633E-14 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
    
```

R-Z binning: (ir, iz)
First loop on ir,
[then on iphi (if exists),]
then on iz

Cartesian binning: (ix, iy, iz)
First loop on ix,
then on iy,
then on iz

First 50 values (first 5 lines) are $\{val_{(ir, iz=1)}\}$, with $ir \in [1, 50]$. Next 50 values are $\{val_{(ir, iz=2)}\}$, with $ir \in [1, 50]$. And so on, for each $iz \in [1, 100]$.

USRBIN mesh geometry check

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

(1) Click "+"
to create a new layer.
Here for example,
we name it "ViewUSRBIN".

(2) Click "add"
and select "Usrbin"

(3) Select "From input"

(4) Select the
USRBIN card name
from your input file

(5) Select the layer you created: "ViewUSRBIN"

(6) Mesh overlay appears with a checker pattern

Options:
Show:
Palette:
Usrbin:
<add>

- Filled box: option is selected
- Empty box: option is unselected

Usrbin:
From Input:
File:
Detector: 2 EneLarge [22]
Norm: LB
X-offset: 0.0
Y-offset: 0.0
Z-offset:
Rotdefi:
Rotdefi2:
Transparency:
Prompt draw:
Palette: Palette

Fluka: ex_scoring_solution.flair
x: 0 y: 6.917738409 z: 11.34723899
Plot saved as: Magenta

Volume normalisation

- **USRBIN region: FLUKA results are NOT normalized by volume.**

When scoring particle *fluence* (e.g. NEUTRON) or *energy density* (ENERGY) with USRBIN

by region, FLUKA results are *total track-length and energy deposition*, respectively.

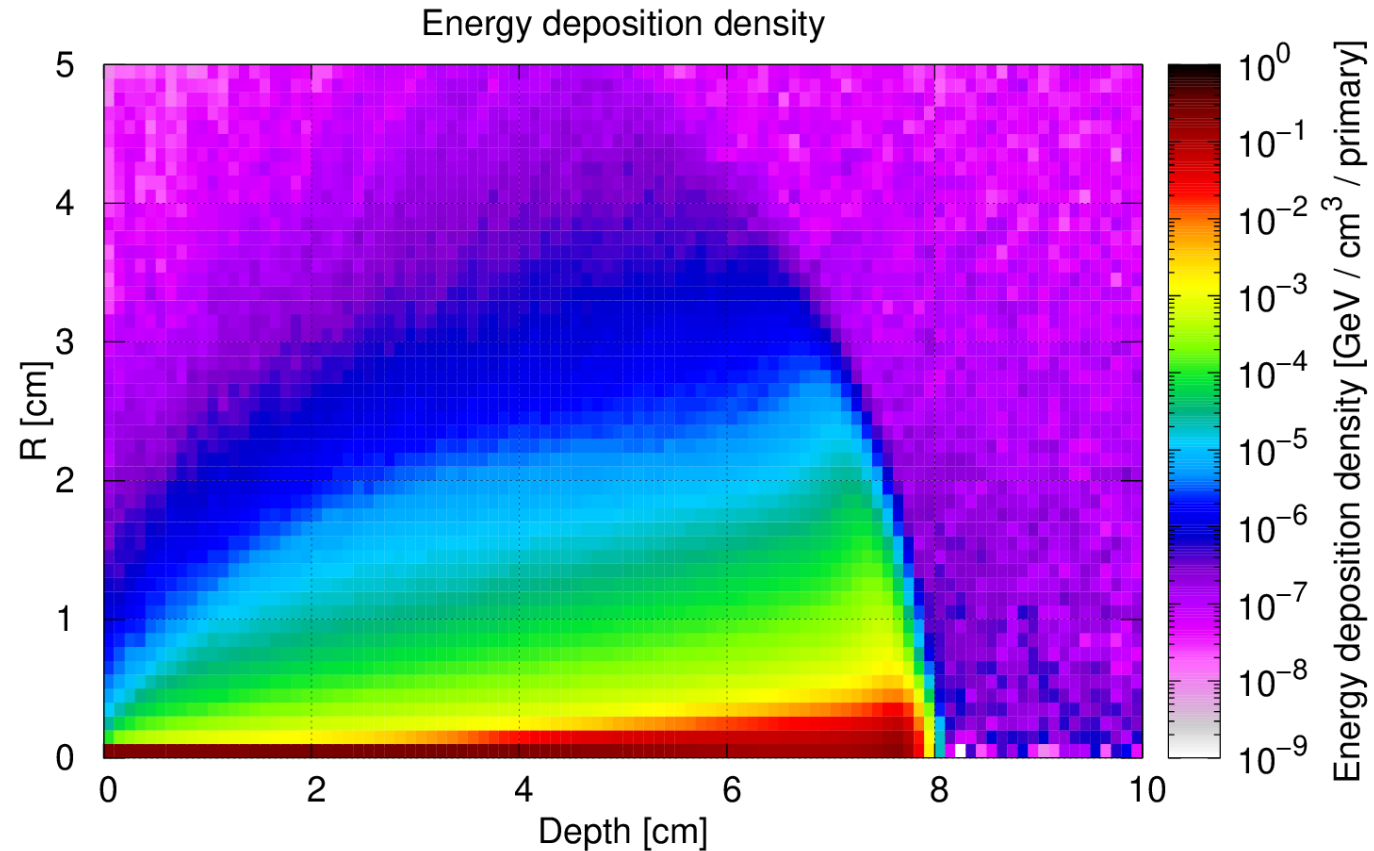
- Indeed, FLUKA does not calculate regions volumes (which can be arbitrarily complex!).
 - Results are provided assuming the region has a volume equal to 1 cm³.
 - The user needs to divide by the region volume to get the intended quantities.
-
- **USRBIN meshes: FLUKA results are normalized by volume.**
- Conversely, as USRBIN scoring on regular (Cartesian, cylindrical) meshes is requested, *particle fluence and energy density* will be automatically provided.
- Results are already normalized by the bins volume(s).
 - Indeed, FLUKA can easily compute the mesh bins volume(s). In addition, it would not make sense for the user to calculate each cylindrical mesh bin volume (varying radially)!

Plotting – Energy deposition density (USRBIN mesh)

The screenshot displays the FLUKA software interface for plotting. The main window is titled "Plot" and shows the configuration for an "Energy deposition density - 2D plot".

- Geometry:** The "Geometry" button in the top toolbar is circled in red.
- Plot ranges:** A blue box labeled "Plot ranges" points to the "Log" checkbox and the "Min" and "Max" input fields in the "Energy deposition density [GeV / cm³ / primary]" row.
- Horizontal and Vertical axes:** A blue box labeled "Horizontal axis" points to the "x: Z [cm]" field, and another labeled "Vertical axis" points to the "y: R [cm]" field. A third blue box labeled "Min/max values in 2D plot" points to the "Min" and "Max" input fields.
- Merged file:** A blue box labeled "Merged file" points to the "File: ex_scoring_solution_21.bnn" field.
- Mesh summary:** A blue box labeled "Mesh summary" points to the "Binning Info" section, which includes:
 - Det: 1 EneSmall (circled in red)
 - R: [0.0 .. 5.0] x 50 (0.1)
 - Φ: [-3.141592653589793 .. 3.141592653589793] x 1 (6.283185307179586)
 - Z: [0.0 .. 10.0] x 100 (0.1)
- Detector from file:** A blue box labeled "Detector from file" points to the "Det: 1 EneSmall" field.
- Type of plot:** A blue box labeled "Type of plot" points to the "Type: 2D Projection" dropdown menu (circled in red) and the "Φ:" field (circled in red).

Plot result – Energy deposition density (USRBIN mesh)



- This plot is a **2D projection** of a 3D structure
→ **the result is the volume-weighted average over the 3rd coordinate** (Φ in this case)
- The 2D limits, and the range on which to perform the average, can be set in Flair

Plot result – Neutron fluence (USRBIN mesh)

Geometry (Menu)

Legend: Red, Green, Blue, Magenta, Energy_density_small_bin_2D, Charged_hadron_fluence, **Neutron_fluence**

Title: Neutron fluence - 2D plot

Options: font: Arial 18, color: [], options: [], grid: [], aspect: [], Auto, lines: solid

Axes:
x: Z [cm]
y: R [cm]
cb: Neutron fluence [cm⁻² per primary]

Binning Detector: File: ex_scoring_solution_32.bnn, Title: Scoring I exercise

Binning Info:
Cycles: 5 Primaries, 1000000 Weight: 1000000.0 Time: **** Sum file ****
Det: 1 Neut, R: [0.0 .. 5.0] x 50 (0.1)
Type: 11 X- Φ -Z, Φ : [-3.141592653589793 .. 3.141592653589793] x 1 (6.283185307179586)
Score: NEUTRON, Z: [0.0 .. 10.0] x 100 (0.1)

Mesh summary:
Min: 2.94608817E-05
Max: 4.52569611E-02
Int: 0.27563673590458149

Projection & Limits:
R: 1, Φ : 1, Z: 1
Type: 2D Projection

set format cb '10^{%T}'
set clabel offset 2

Horizontal axis
Vertical axis
Min/max values in 2D plot

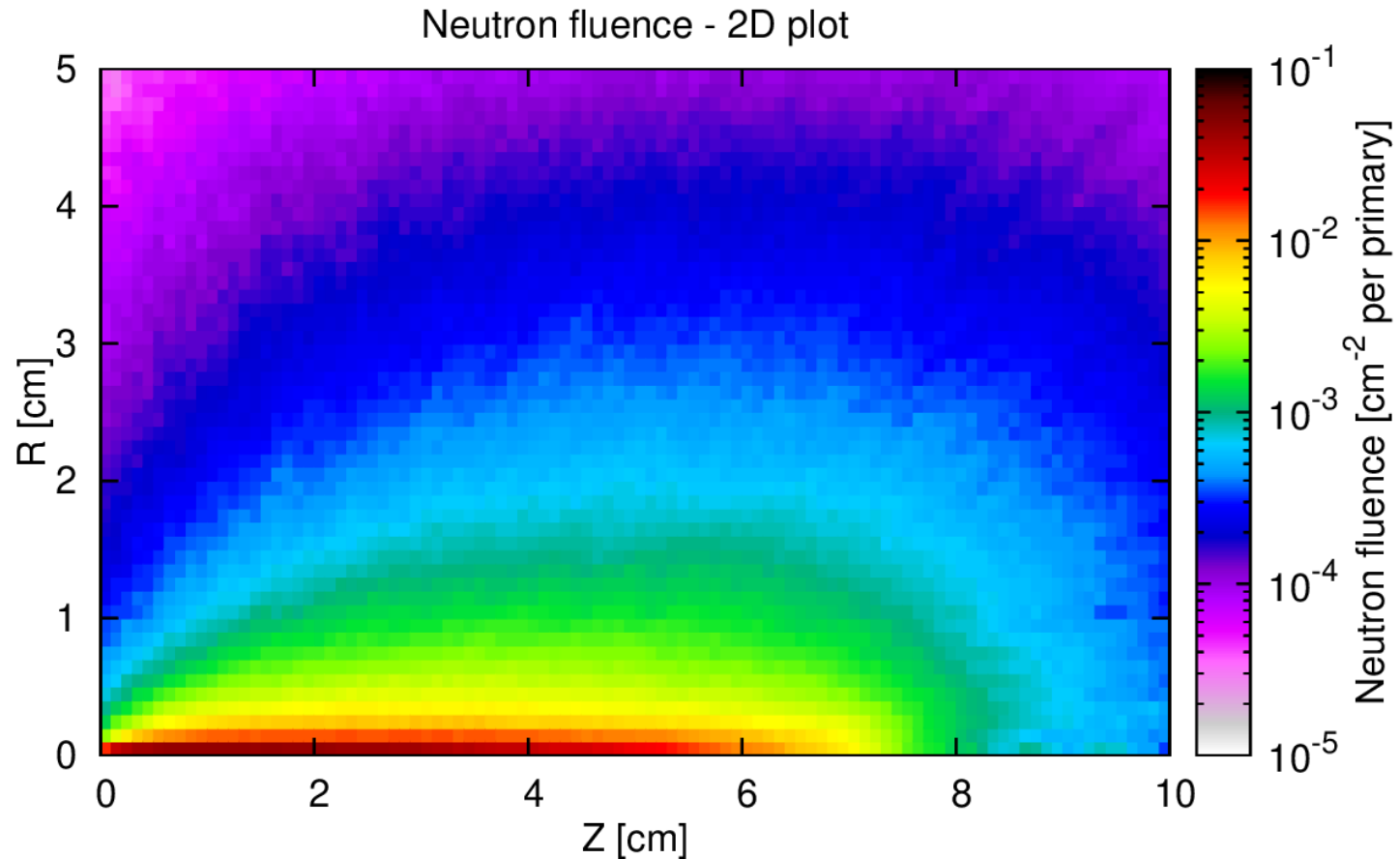
Merged file

Mesh summary

Detector from file

Type of plot

Plotting – Neutron fluence (USRBIN mesh)



- This plot is a **2D projection** of a 3D structure
→ **the result is the volume-weighted average over the 3rd coordinate** (Φ in this case)
- The 2D limits, and the range on which to perform the average, can be set in Flair

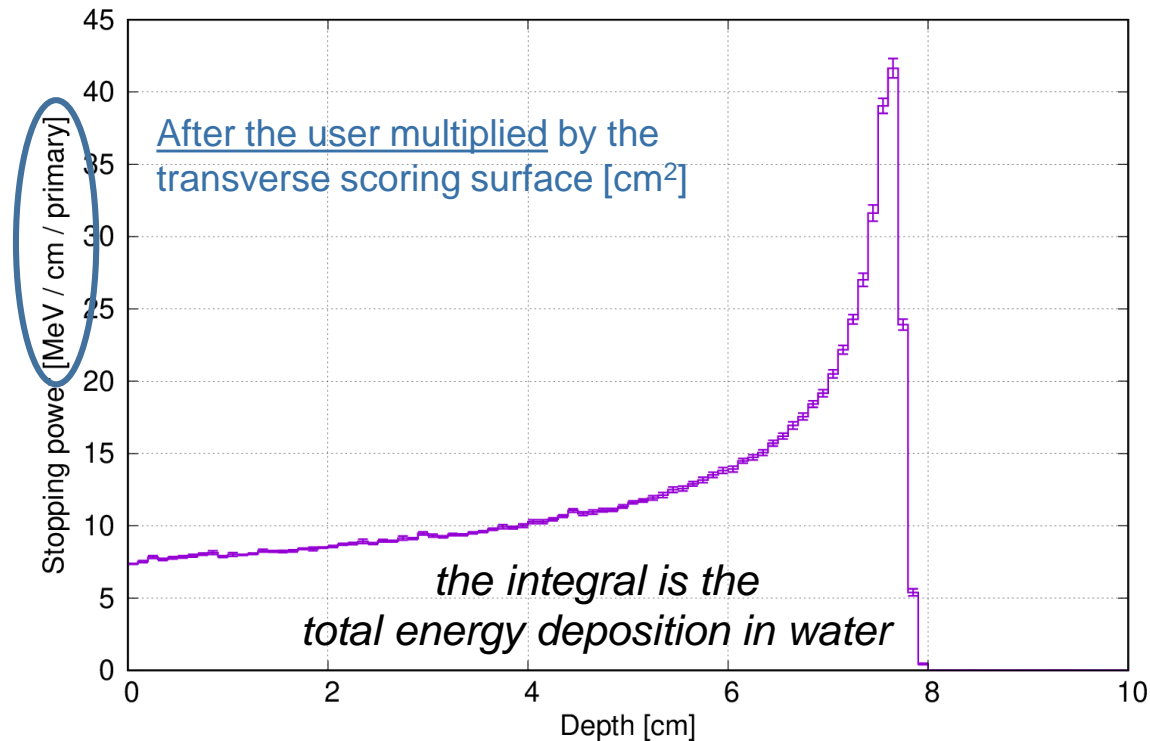
One dimensional plots

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

1D Projection

- For each z bin:
average over all {R, Φ } bins

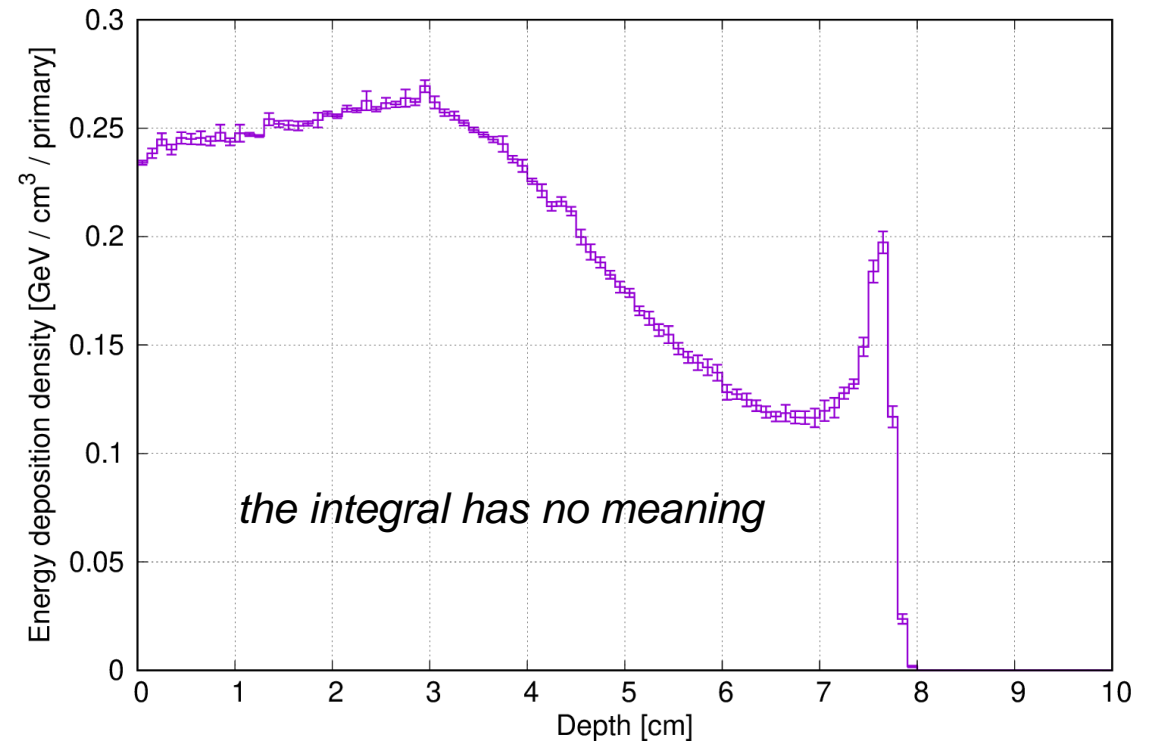
100 MeV proton Bragg peak in water



1D Max

- For each z bin:
highest over all {R, Φ } bins

Peak energy deposition density



Overlaying USRBIN mesh results on 2D/3D geometry

(1) Click "+" to create a new layer. Here for example, we name it "Edep_3D".

(2) Click "add" and select "Usrbin". Also add "3D".

(3) Select merged file from run results

(4) Select the USRBIN card name from your input file

(5) Select the layer you created: "Edep_3D"

(6) USRBIN mesh results overlay the geometry: 2D projection 3D view

This is another layer example, where "3D" option is **unselected**

- Filled box: option is selected
- Empty box: option is **unselected**

red: time-consuming

Fluka: ex_scoring_solution.flair x: 9.435994261 y: 0 z: -1.240894938 Plot saved as: Magenta

