Estimators and Scorings

FLUKA Beginner’s Course

FLUKA Scoring & Results - Estimators

- It is often 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 the respective quantities they are interested in.
- As the latter is implemented in a very complete way, users are strongly encouraged to preferably use the built-in estimators with respect to user-defined scoring
- For additional requirements FLUKA user routines are provided

FLUKA Scoring & Results –summary

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

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

When?

At the end of each cycle

At each event

Output? Files inp*nnn*_fort.## where ## is logical unit number chosen by user

Results?

Postprocessing utilities (in $FLUPRO/flutil) merge cycles, calculate average and rms, provide data files for plotting

Data merging and plotting available in FLAIR

User code needed

Results normalized **per primary**

Before giving details

- A couple of slides on important definitions / quantities:
- Fluence and the difference wrt current
- Fluence as the correct quantity to evaluate the potential effect of a particle field on a “detector”
- Different methods to evaluate fluence

Reaction Rate and Cross Section (1/3)

- We call mean free path $\lambda[cm]$ the average distance travelled by a particle in a material before an interaction. Its inverse, $\Sigma\ [cm^{-1}]$ is the probability of interaction per unit distance, and is called macroscopic cross section. Both λ and Σ depend on the material and on the particle type and energy.

- For N identical particles, the number of reactions R occurring in a given time interval will be equal to the total distance travelled Nl times the probability per unit distance Σ: $R = Nl\Sigma$

- The reaction rate will be $\dot{R} = N\mathrm{d}l/\mathrm{d}t\ \Sigma = Nv\Sigma$, where v is the average particle velocity.

Reaction Rate and Cross Section (2/3)

- Assume now that $n(\mathbf{r},v)=dN/dV$ $[cm^{-3}]$ be the density of particles with velocity $v=dl/dt$ $[cm/s]$, at a spatial position $\mathbf{r}$. The reaction rate inside the volume element $\mathrm{d}V$ will be: $\mathrm{d}\dot{R}/\mathrm{d}V = n(\mathbf{r},v)v\Sigma$
- The quantity $\dot{\Phi}(\mathbf{r},v) = n(\mathbf{r},v)v$ is called fluence rate or flux density and has dimensions $[cm^{-3}\ cm\ s^{-1}]=[cm^{-2}\ s^{-1}]$.
- The time integral of the flux density $\Phi(\mathbf{r},v) = n(\mathbf{r},v)\mathrm{d}l$ is the fluence $[cm^{-2}]$
- Fluence is measured in particles per cm^2 but in reality it describes the **density of particle tracks**
- The number of reactions inside a volume V is given by the formula: $R = \Sigma\Phi V$ (where the product $\Sigma\Phi$ is integrated over energy or velocity)

Reaction Rate and Cross Section (3/3)

- Dividing the macroscopic cross section by N_0, the number of atoms per unit volume, one obtains the microscopic cross section: $\sigma[barn=10^{-24}cm^2]$

$$\frac{\text{probability/cm}}{\text{atoms/cm}^3} = \frac{\text{probability} \times \text{cm}^2}{\text{atom}} = \text{atom effective area}$$

  i.e., the area of an atom weighted with the probability of interaction (hence the name "cross section");

- But it can also be understood as 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).
- In this way, both microscopic and macroscopic cross section are shown to have a similar physical meaning of "probability of interaction per unit length", with length measured in different units. Thus, the number of interaction can be obtained by both by multiplying by the corresponding particle track-length.

Fluence estimation (1/2)

- Track length estimation: USRTRACK

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

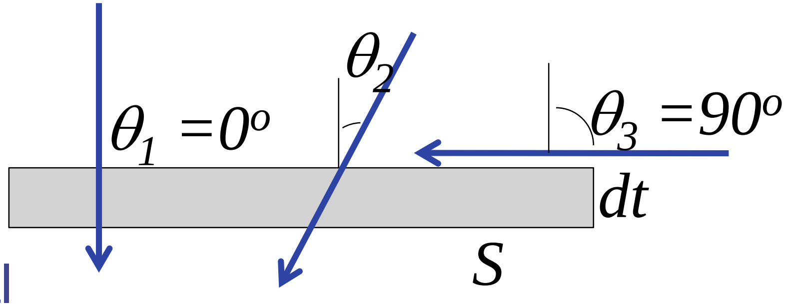
- Collision density estimation: USRCOLL

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

Fluence estimation (2/2)

Surface crossing estimation

- Imagine a surface having an infinitesimal thickness dt
  A particle incident with an angle θ with respect to the normal of the surface S will travel a segment $dt/\cos\theta$.



- Therefore, we can calculate an average surface fluence by adding $dt/\cos\theta$ for each particle crossing the surface, and dividing by the volume $S\,dt$:

$$\Phi = \lim_{dt \to 0} \frac{\sum_i \frac{dt}{\cos\theta_i}}{S\,dt}$$

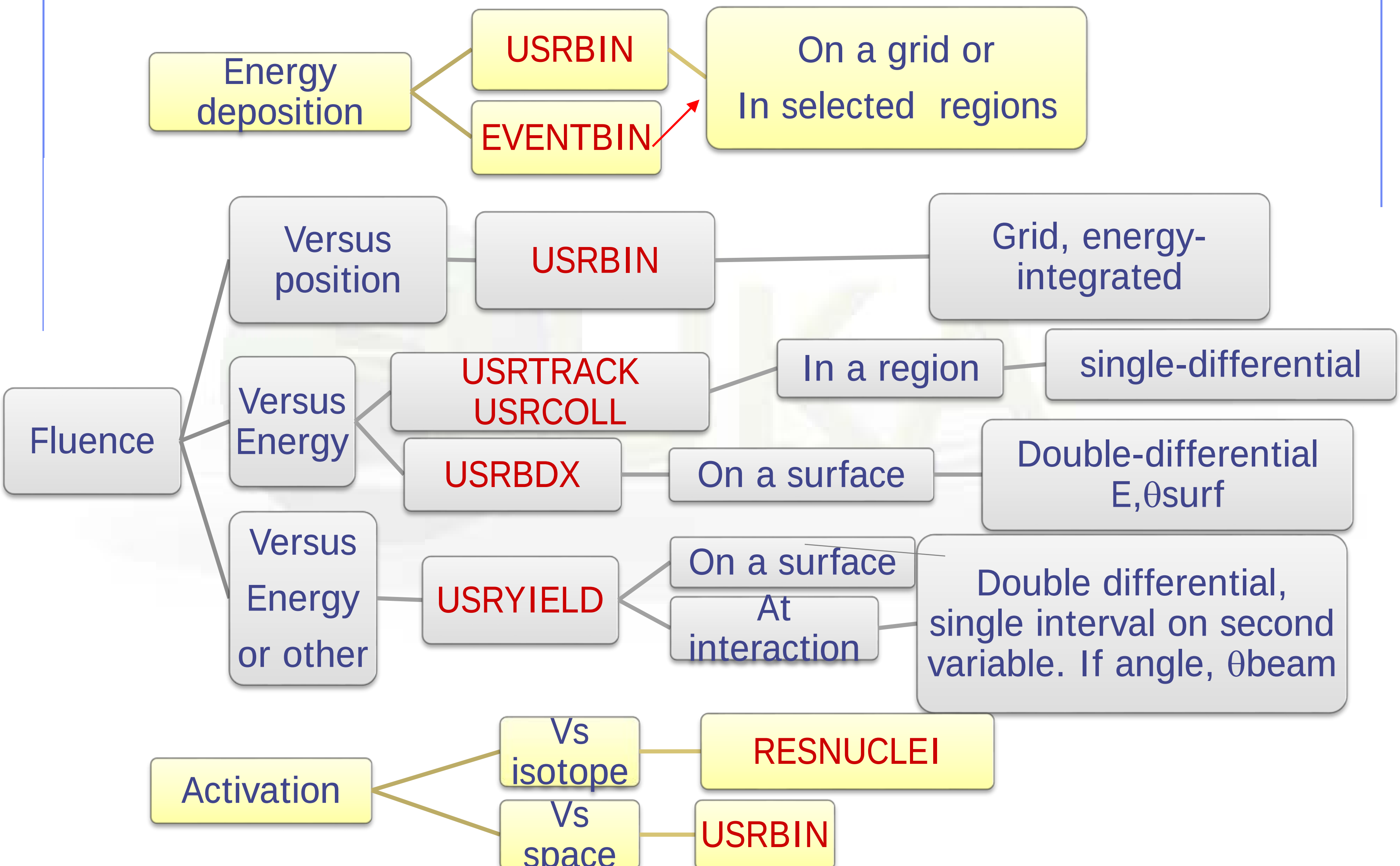
- While the current J counts the number of particles crossing the surface divided by the surface:

$$J = dN/dS$$

The fluence is independent from the orientation of surface S,
while the current is NOT!

In an isotropic field can be easily seen that on a flat surface $J = \Phi/2$

Fluka estimators



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
  - Built-in scoring comes with **utility programs** that allow to evaluate statistical errors
- Standard scoring can be weighted by means of simple user routines (fluscw, comscw)

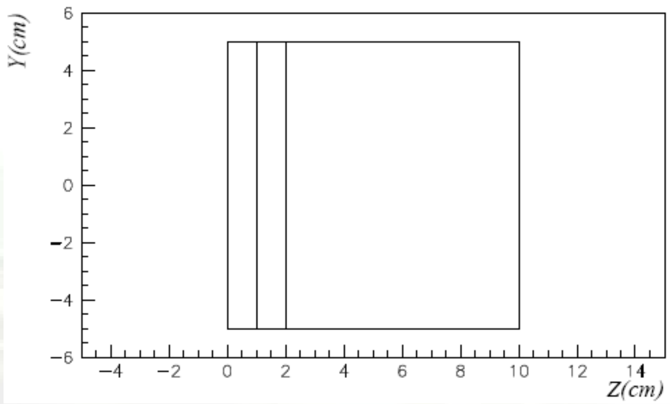
The FLUKA Output Files

The Fluka output consists of:

- A main (standard) output, written on logical output unit LUNOUT (predefined as 11 by default) [.out]
- A file with the last random number seeds, unit LUNRAN (2 by default) [ran*]
- A file of error messages, unit LUNERR (15 by default) [.err]
- Any number (including zero) of estimator output files. Their logical unit number is defined by the user [*fort_xx*]
- The available range of logical output numbers is: 21-99
- Generally, the user can choose between formatted (ASCII) and unformatted (binary) scoring (negative or positive sign in the logical unit number). Unformatted scoring is mandatory for the use of provided post-processing utilities.
- Several estimators can be output on the same file (same logical unit) provided they are of the same type
- Possible additional output generated by the user in any user routine;

Extending the example with Scoring

- Cylinder along Z, filled by water-aluminum-lead and surrounded by Air



- The USRBIN command allows to define a 3-D grid, either cartesian, cylindrical (R-Z-Φ) or by region.
- On this grid, one can score energy deposition, particle fluence (total or by particle type), as well as the density of interactions;
- There is an equivalent EVENTBIN command, that outputs the same quantities event-by-event;
- Using USERWEIG the results can be weighted by the comscw.f or fluscw.f external routines (Advanced).

Example: Scoring

- Start from the solution of ex_ (Either copy your .inp and .flair files and rename them to ex_score, or download from the site & rename):

```
mkdir example_score ;
cp ex_geo1/ex_geo1_final.inp example_score/example_score.inp ;
cd example_score
```

- Open in flair or with your preferred editor
- Add USRBIN to score ENERGY on a CYLINDRCAL GRID (r, z, ϕ) covering the target and surroundings: $0<r<10$, $-5<z<15$, with cells having $\Delta r=\Delta z=1$mm , $\Delta\phi=2\pi$
- Add USRBIN to score Neutron Fluence on the same grid as above
- Run 5 cycles, 1000 primaries each
- .. We do it together.
- First of all, open the manual!

WHATs

WHAT(1) : code indicating the type of binning selected.

Usrbin: 11.0 : Mesh: R-Z or R-Phi-Z, no symmetry

WHAT(2) : particle (or particle family) type to be scored

WHAT(3) = logical output unit

The first three fields have similar meanings for all estimators. The other what's will contain the limits or regions or boundaries, and the number of intervals for cells/histograms

SDUM (optional) : name of the estimator

For Ursbin, r-phi-z:

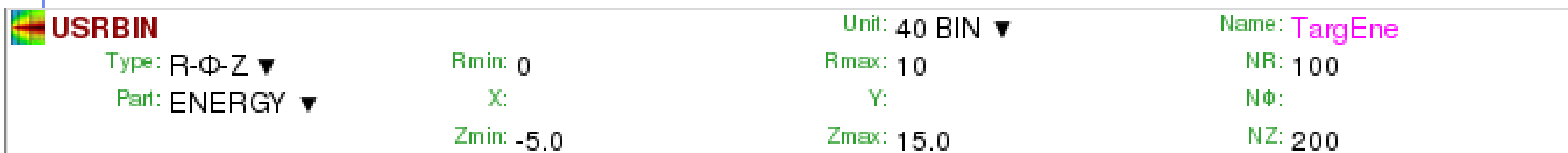
WHAT(4)= For R-Z and R-Phi-Z binning: Rmax

WHAT(5)= For Cartesian binning: Ymax For R-Z and R-Phi-Z binning: Y coordinate of the binning axis

…….. And so on

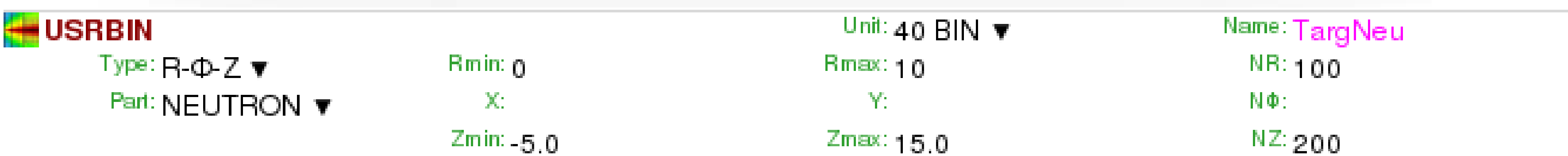
USRBIN

```
* Energy deposition [GeV/cm³]
*    +    1    +    2    +    3    +    4    +    5    +    6    +    7    +
USRBIN          11.0    ENERGY     -40.0      10.0                 15.0TargEne
USRBIN           0.0                -5.0     100.0                200.0&
```



- This is an R-Z-Φ binning (what(1)=11), scoring energy density (generalized particle ENERGY, or 208), writing the unformatted output on unit 40, spanning $0<R<10$ in 100 bins, $0<\Phi<2\pi$ in 1 bin (default), $-5<z<15$ in 200 bins.

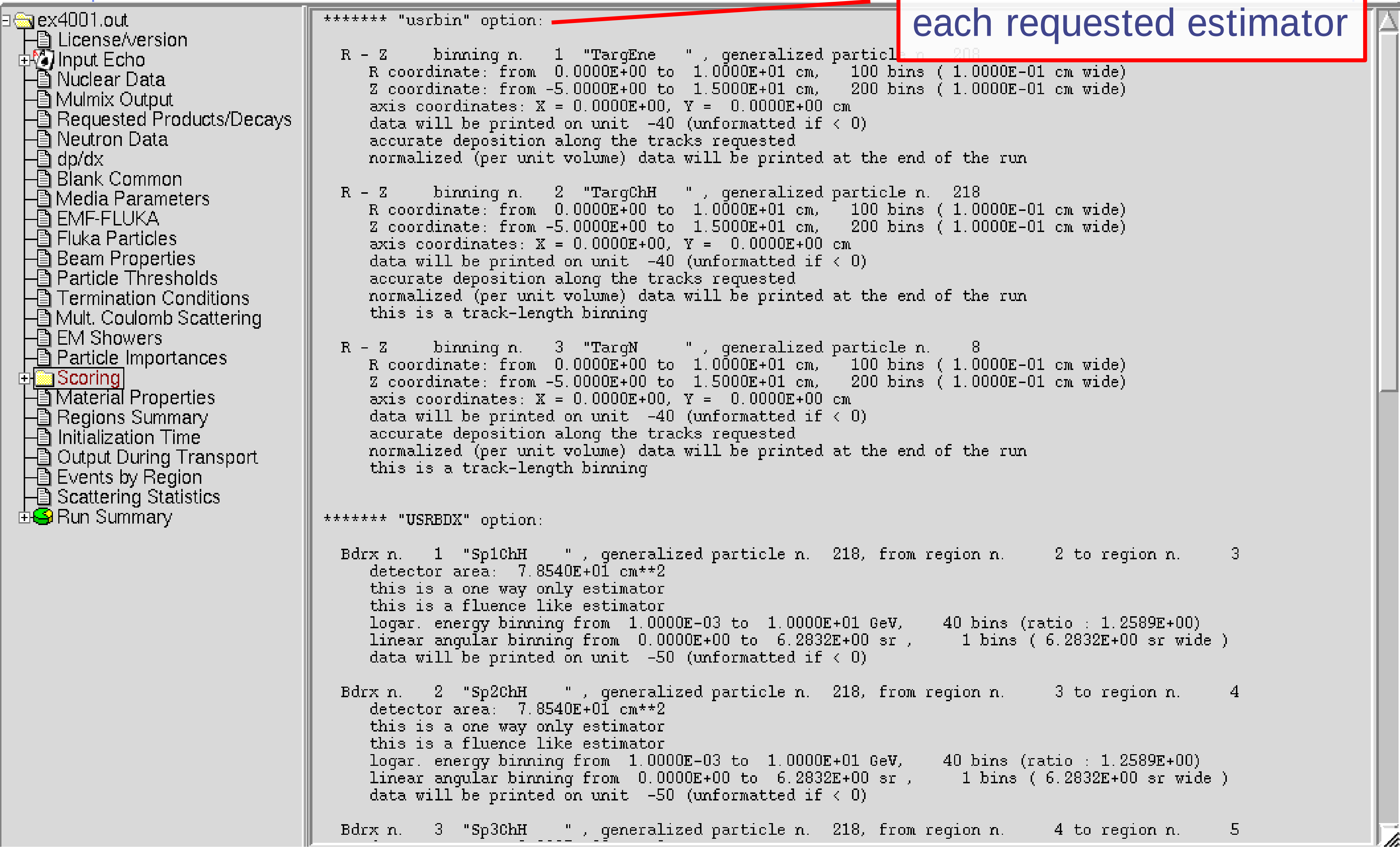
```
* Neutron fluence [cm⁻²]
*    +    1    +    2    +    3    +    4    +    5    +    6    +    7    +
USRBIN          11.0   NEUTRON     -40.0      10.0                 15.0TargNeu
USRBIN           0.0                -5.0     100.0                200.0&
```



- This is a R-Z-Φ binning (what(1)=11), scoring neutron fluence, writing the unformatted output on unit 40, spanning $0<R<10$ in 100 bins, $0<\Phi<2\pi$ in 1 bin (default), $-5<z<15$ in 200 bins.

Check in the .out file

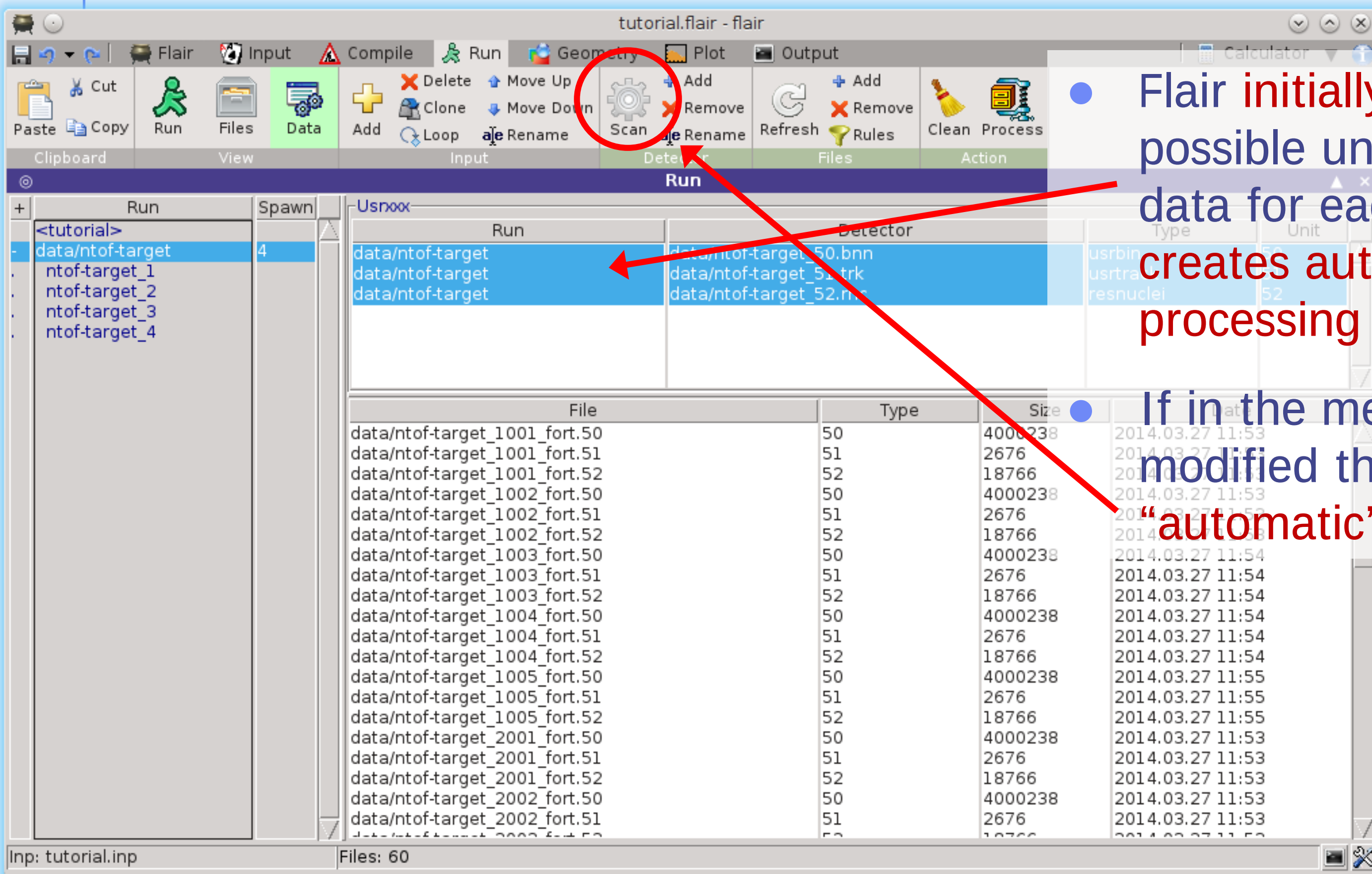
Complete description of each requested estimator




```
******* "usrbin" option:

  R - Z     binning n.   1  "TargEne    " , generalized particle n.  208
     R coordinate: from  0.0000E+00 to  1.0000E+01 cm,    100 bins ( 1.0000E-01 cm wide)
     Z coordinate: from -5.0000E+00 to  1.5000E+01 cm,    200 bins ( 1.0000E-01 cm wide)
     axis coordinates: X = 0.0000E+00, Y =  0.0000E+00 cm
     data will be printed on unit  -40 (unformatted if < 0)
     accurate deposition along the tracks requested
     normalized (per unit volume) data will be printed at the end of the run

  R - Z     binning n.   2  "TargChH    " , generalized particle n.  218
     R coordinate: from  0.0000E+00 to  1.0000E+01 cm,    100 bins ( 1.0000E-01 cm wide)
     Z coordinate: from -5.0000E+00 to  1.5000E+01 cm,    200 bins ( 1.0000E-01 cm wide)
     axis coordinates: X = 0.0000E+00, Y =  0.0000E+00 cm
     data will be printed on unit  -40 (unformatted if < 0)
     accurate deposition along the tracks requested
     normalized (per unit volume) data will be printed at the end of the run
     this is a track-length binning

  R - Z     binning n.   3  "TargN      " , generalized particle n.    8
     R coordinate: from  0.0000E+00 to  1.0000E+01 cm,    100 bins ( 1.0000E-01 cm wide)
     Z coordinate: from -5.0000E+00 to  1.5000E+01 cm,    200 bins ( 1.0000E-01 cm wide)
     axis coordinates: X = 0.0000E+00, Y =  0.0000E+00 cm
     data will be printed on unit  -40 (unformatted if < 0)
     accurate deposition along the tracks requested
     normalized (per unit volume) data will be printed at the end of the run
     this is a track-length binning


******* "USRBDX" option:

  Bdrx n.   1  "Sp1ChH    " , generalized particle n.  218, from region n.     2 to region n.     3
     detector area:  7.8540E+01 cm**2
     this is a one way only estimator
     this is a fluence like estimator
     logar. energy binning from  1.0000E-03 to  1.0000E+01 GeV,    40 bins (ratio : 1.2589E+00)
     linear angular binning from  0.0000E+00 to  6.2832E+00 sr ,     1 bins ( 6.2832E+00 sr wide )
     data will be printed on unit  -50 (unformatted if < 0)

  Bdrx n.   2  "Sp2ChH    " , generalized particle n.  218, from region n.     3 to region n.     4
     detector area:  7.8540E+01 cm**2
     this is a one way only estimator
     this is a fluence like estimator
     logar. energy binning from  1.0000E-03 to  1.0000E+01 GeV,    40 bins (ratio : 1.2589E+00)
     linear angular binning from  0.0000E+00 to  6.2832E+00 sr ,     1 bins ( 6.2832E+00 sr wide )
     data will be printed on unit  -50 (unformatted if < 0)

  Bdrx n.   3  "Sp3ChH    " , generalized particle n.  218, from region n.     4 to region n.     5
```

Flair: Data Processing

In the run tab

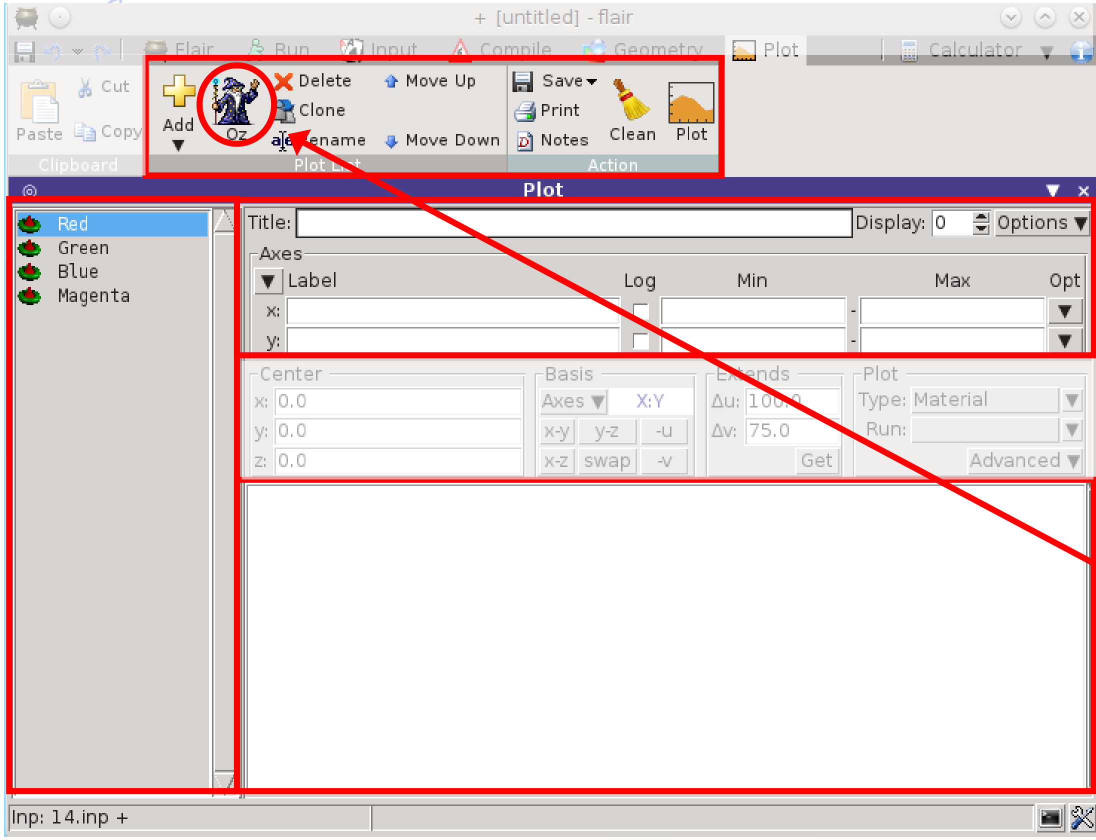


- Flair initially scans the input for possible unformatted output data for each scoring card. It creates automatic rules for processing (merging).
- If in the mean time you have modified the input click the “automatic” scan

FLAIR uses the auxiliary programs available in $FLUPRO/flutil to merge output data. For USRBIN: the usbsuw code

- The default names are generated by the rules specified in the preference dialog, can be changed if desired (advanced)

Plot List

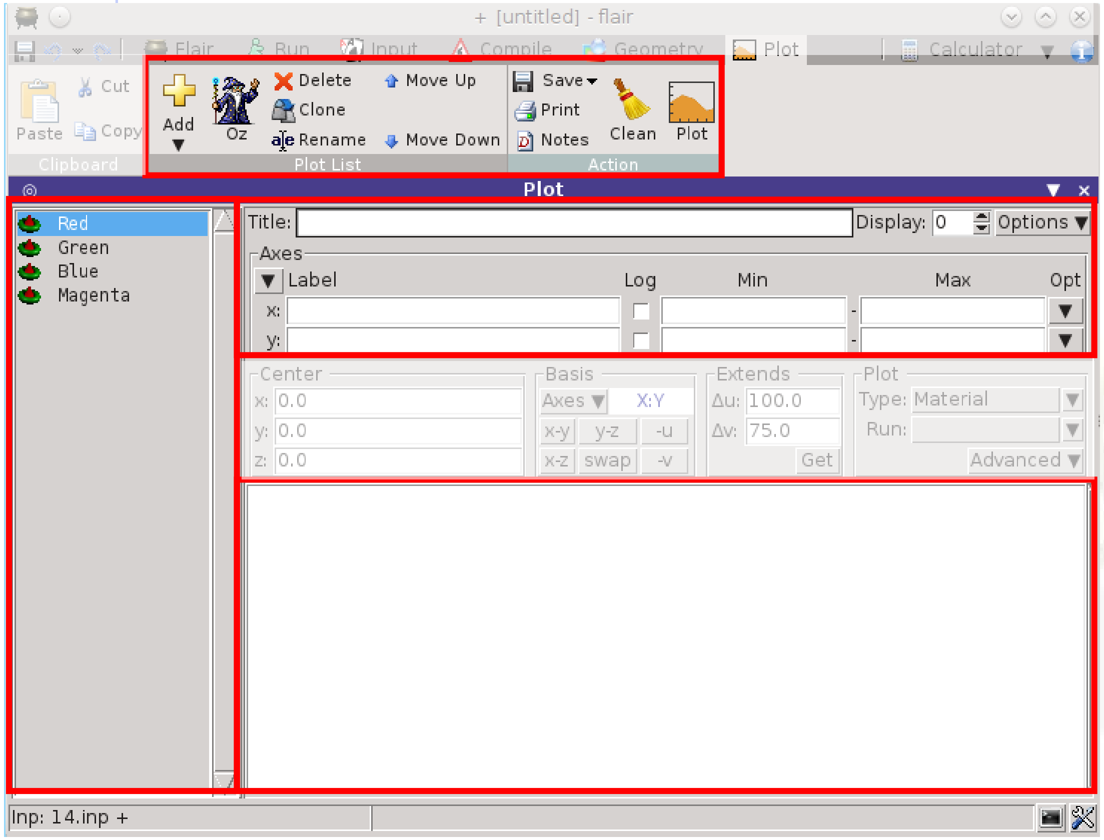


- Plots can be created in the “Plot” list frame. Either Add new plots or Clone from existing ones.
- It is important to set a unique filename for each plot.
  This filename will be used for every auxiliary file that the plot needs (with different extensions)
- The Wizard button creates automatically one plot for each processed unit
- Double click on a plot, or hit Enter or click the Edit icon to display the plotting dialog

Plot Types

Geometry	For geometry plots
USRBIN	For plotting the output of USRBIN
USR-1D	To plot single differential quantities from cards USRBDX, USRTRACK, USRCOLL, USRYIELD
USR-2D	To plot double differential from USRBDX
RESNUCLE	To plot 1d or 2d distributions of RESNUCLEi
USERDUMP	To plot the output of USERDUMP. Useful for visualizing the source distribution.

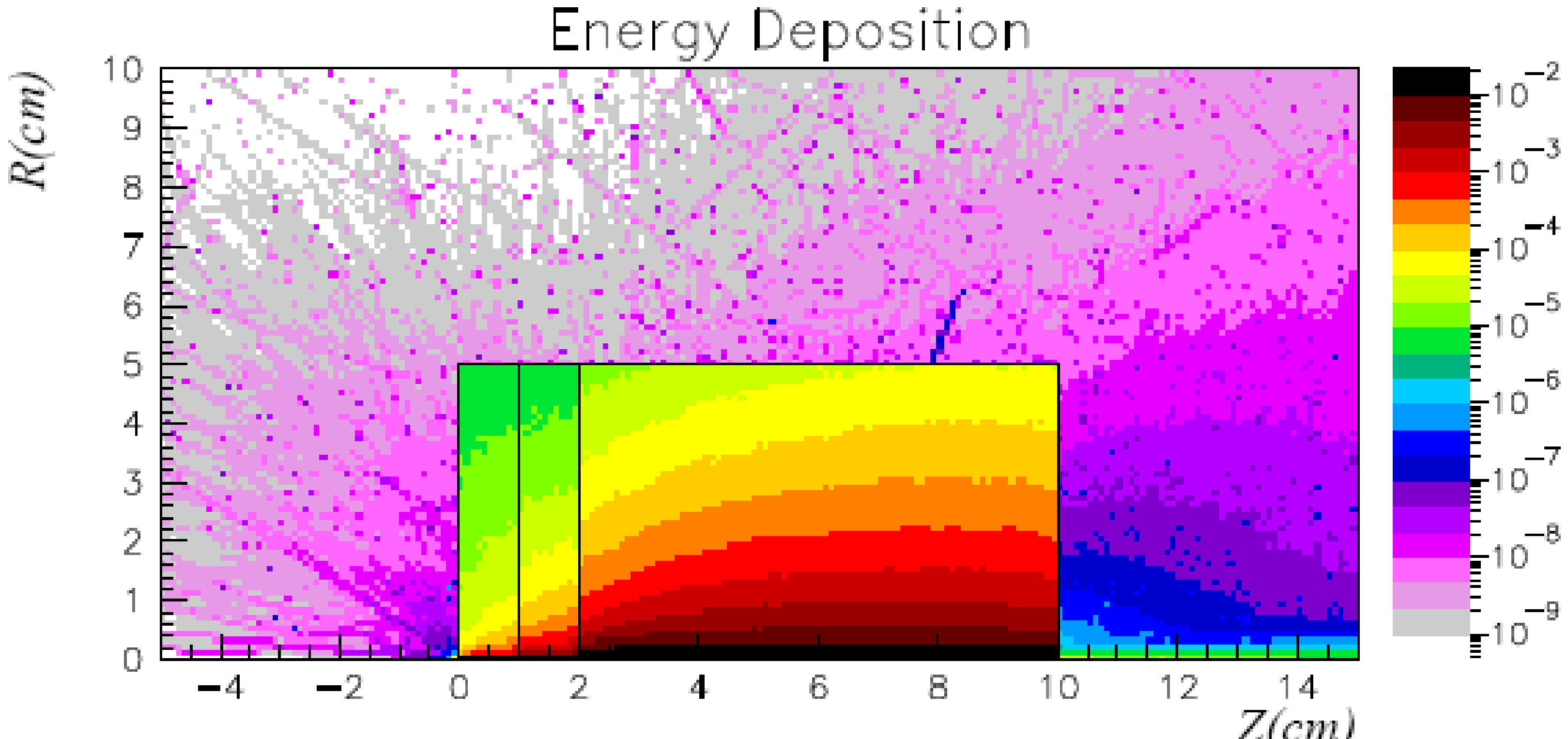
Plotting Frames



- **FLAIR uses gnuplot to produce plots**
- All plot types share some common fields: Title + options, Filename, Axis Labels, Legends (Keys) and Gnuplot Commands.
- Plot button (Ctrl-Enter) will generate all the necessary files to display the plot, ONLY if they do not exist.
- All plots are listed in the Plot List
- Clean button will remove all files generated by Flair during plotting process.
- Check the Gnuplot manual to provide additional customization commands: e.g. To change the title font to Times size=20, add in the Opt: field the command: font 'Times,20'

USRBIN → The Result

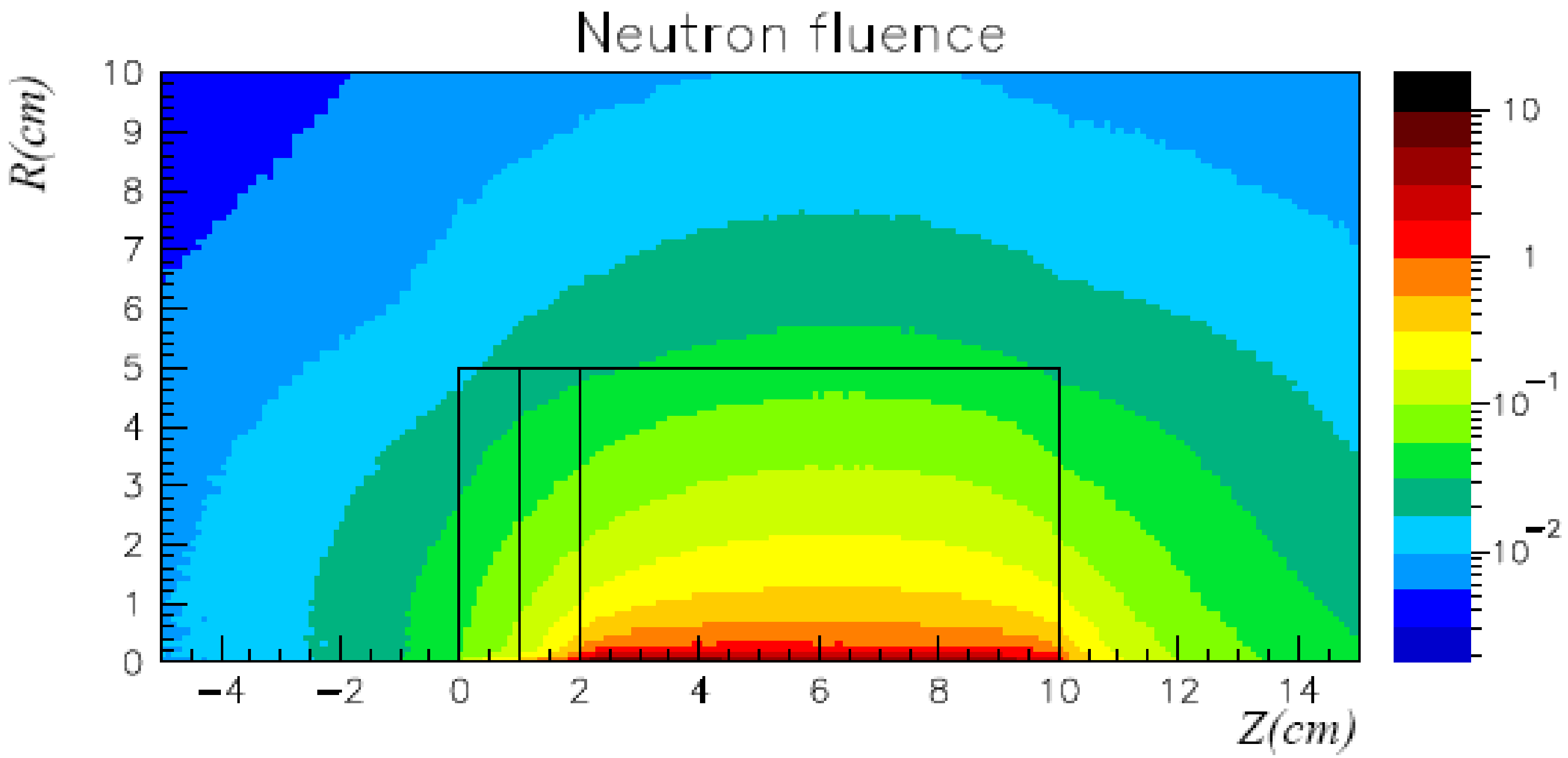
WHAT(2) = ENERGY : Energy deposition from a 3.5 GeV proton beam hitting at [0.,0.,0.] directed along z
results are normalized to **GeV/cm³ per primary**



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

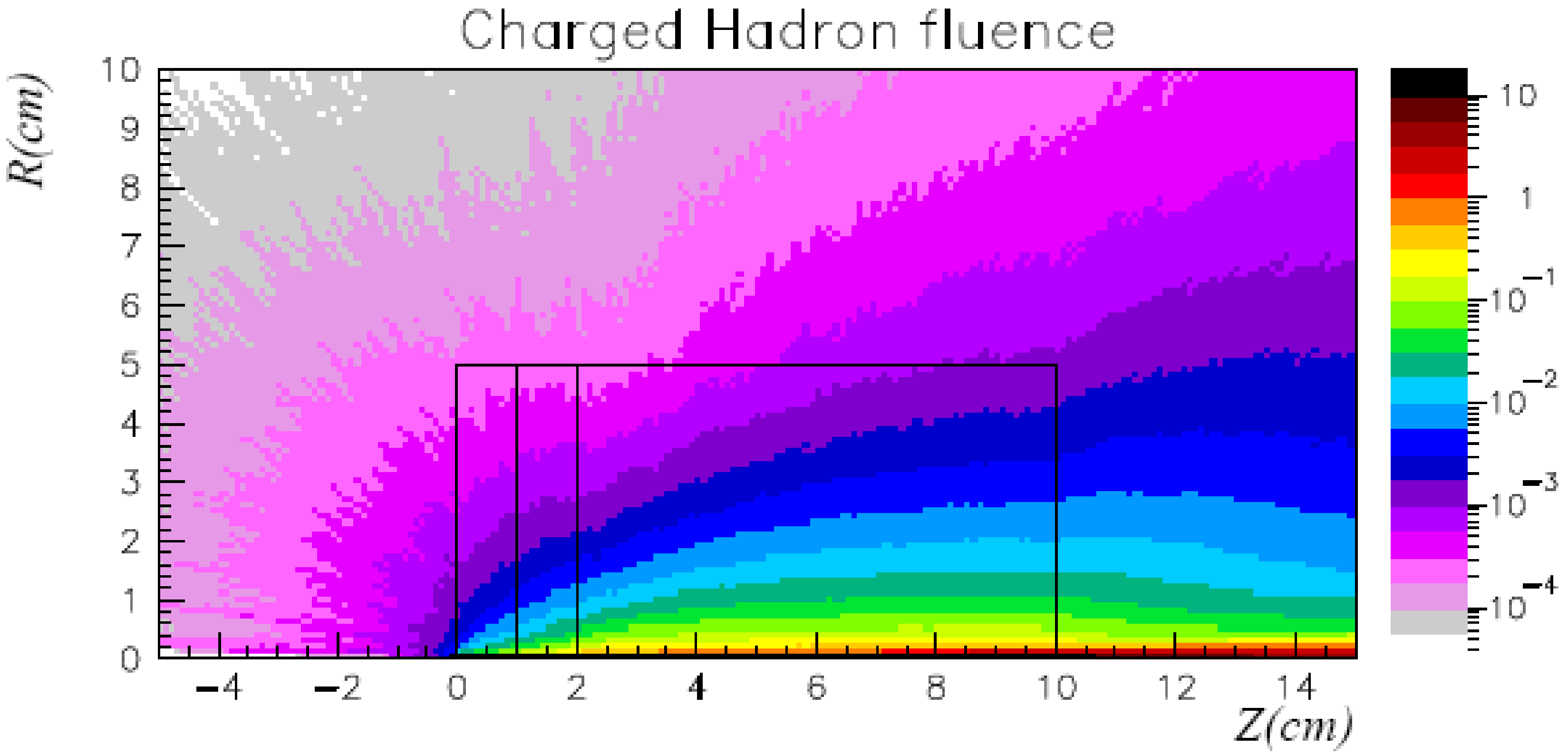
USRBIN → The Result

Same, WHAT(2)= NEUTRON to get neutron fluence
results are normalized to **particles/cm² per primary**



USRBIN → The Result

Same, WHAT(2)= HAD-CHAR to get charged hadron fluence
results are normalized to **particles/cm² per primary**



USRBIN – more quantities

USRBIN can score Particle fluence as well as "Generalized particles", either fluence-like or energy-like, for instance with what(2) =

- DOSE: Energy/unit mass (GeV/g)
- DPA-SCO: Displacements per atom (see the lecture on Ionization and transport)
- X-MOMENT: x-component of momentum transfer (GeV/c)
- ACTIVITY: activity per unit volume (Bq/cm^3) (see lecture on radioactivity)

- ... and more (see in the manual)

USRBDX

* in this case post-processed results are single differential (already integrated over the solid angle)

USRBDX scores double differential (energy and angle) particle distributions across a boundary surface. The angle is with respect to the normal of the surface. The distribution can be fluence or current, one-way or two-ways, according to WHAT(1)

- Score charged hadrons at the outer surface of the lead segment (from TARGS3 to INAIR). WHAT(1)=99 means: fluence, one-way only, log. intervals in energy. From 1 MeV to 10 GeV in 40 intervals, and *one angular interval (default)**. WHAT(6) is a normalization factor: setting it equal to the surface area provides results normalized to cm^{-2} (fluence unit) GeV^{-1} sr^{-1}. Output to unformatted unit 50

```
*    +    1    +    2    +    3    +    4    +    5    +    6    +    7    +
* out from lead
USRBDX          99.0  HAD-CHAR      -50.    TARGS3     INAIR    329.87Sp3ChH
USRBDX          10.0     0.001       40.                                &
```

USRBDX		Unit: 50 BIN ▼	Name: Sp3ChH
Type: Φ1,LogE,LinΩ ▼	Reg: TARGS3 ▼	to Reg: INAIR ▼	Area: 329.87
Part: HAD-CHAR ▼	Emin: 0.001	Emax: 10.	Ebins: 40
	Ωmin:	Ωmax:	Ωbins:

- Score at the surface between 2nd and 3rd target section, same as before but in 3 angular bins.

```
*    +    1    +    2    +    3    +    4    +    5    +    6    +    7    +
USRBDX          99.0  HAD-CHAR      -54.    TARGS2    TARGS3   78.5398Sp2ChHA
USRBDX          10.0     0.001       40.                             3.0&
```

USRBDX		Unit: 54 BIN ▼	Name: Sp2ChHA
Type: Φ1,LogE,LinΩ ▼	Reg: TARGS2 ▼	to Reg: TARGS3 ▼	Area: 78.5398
Part: HAD-CHAR ▼	Emin: 0.001	Emax: 10.	Ebins: 40
	Ωmin:	Ωmax:	Ωbins: 3

Standard Postprocessing Programs

- To analyse the results of the different scoring options, several programs are made available
- The most natural ones are kept in $FLUPRO/flutil.
- 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
- Each of these programs (except usbsuw) produces three files:
  - a text file with extension *_sum.lis* which contains averaged distributions, standard deviations, **cumulative (integral)** quantities
  - an unformatted file which can replace the *N* unformatted estimator files in further postprocessing
  - a text file with extension *_tab.lis* to be easily readout by graphics codes

[Simpler programs are also provided in the manual, as guide for users who would like to write their own analysis program].

Standard Postprocessing Programs

- Example of sum.lis from USRBDX

Total primaries run: 3000

Total weight of the primaries run: 3000.00000

Detector n: 1 (1) Sp2ChHA

(Area: 78.5398026 cmq,

distr. scored: 218 ,**Charged Hadrons**

from reg. 3 to 4 ,

one way scoring,

fluence scoring scoring)

User provided the area in what(6), thus normalization is /cm^2

Tot. resp. (Part/cmq/pr) 1.4218389E-02 +/- 1.486696 %

(--> (Part/pr) 1.116709 +/- 1.486696 %)

**** Different. Fluxes as a function of energy **** **** (integrated over solid angle) ****

....

**** Cumulative Fluxes as a function of energy **** **** (integrated over solid angle)

....

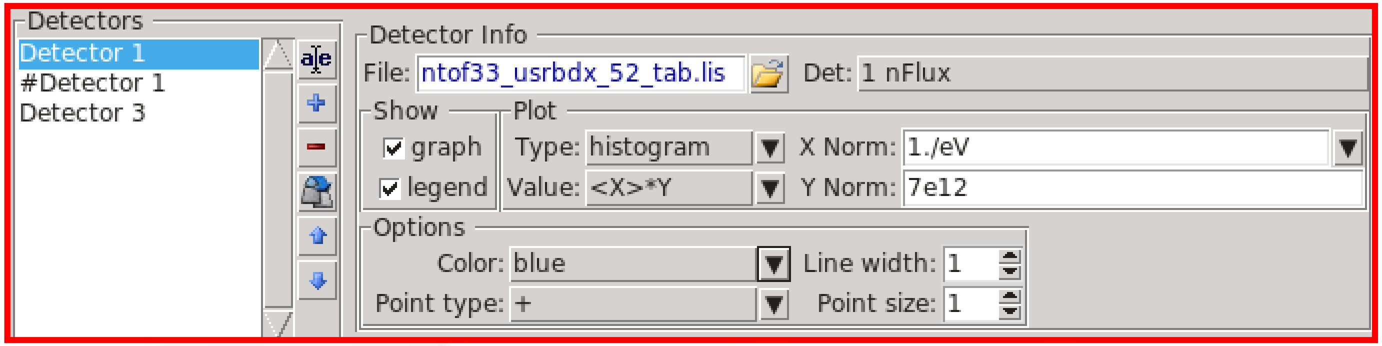
**** Double diff. Fluxes as a function of energy ****

Standard Postprocessing Programs

- Example of tab.lis for USRBDX

```
 # Detector n:          1   Sp2ChHA   (integrated over solid angle)
 # N. of energy intervals        40
   Emin        Emax        Result       Error  (%)
1.000E-02  1.259E-02  5.049E-03  6.420E+01
 1.259E-02  1.585E-02  1.115E-03  6.934E+01
 1.585E-02  1.995E-02  2.826E-03  4.675E+01
 1.995E-02  2.512E-02  2.356E-03  3.866E+01
 2.512E-02  3.162E-02  6.437E-03  2.857E+01
...
# double differential distributions
 # number of solid angle intervals        3
 # Block n:          1      0.00000000      :   2.09439516
1.000E-02  1.259E-02  2.980E-04  9.900E+01
 1.259E-02  1.585E-02  1.981E-04  9.900E+01
 1.585E-02  1.995E-02  3.866E-04  6.700E+01
 1.995E-02  2.512E-02  7.171E-04  3.402E+01
 2.512E-02  3.162E-02  5.544E-04  4.550E+01
```

FLAIR USR-1D Single Differential Plot



- USR-1D is able to plot the 1D single differential information from the USRBDX, USRCOLL, USRTRACK and USRYIELD cards (The 2D information is not handled).
- The file type in use should have the extension **_tab.lis** and are generated by the FLUKA data merging tools
- You can superimpose many scoring output in a single plot.
- Error bars can be plotted (for instance using histerr instaed of histogram

Bins, units etc

- Results from USRBDX, USRYIELD, USRTRACK, USRCOLL are given as DIFFERENTIAL distributions of fluence in energy, in units of (cm-2) GeV^{-1} per primary. Thus,
- A) results are independent on the chosen binning
- B) to obtain INTEGRAL results (fluence in cm^{-2} PER ENERGY INTERVAL per primary) one must multiply the value of each energy bin by the width of the bin (even for logarithmic binning):

$$N = \int \frac{dN}{dE} dE$$

- When scoring Neutrons (see lecture), the E bin limits below 20 MeV are automatically set==transport groups limits

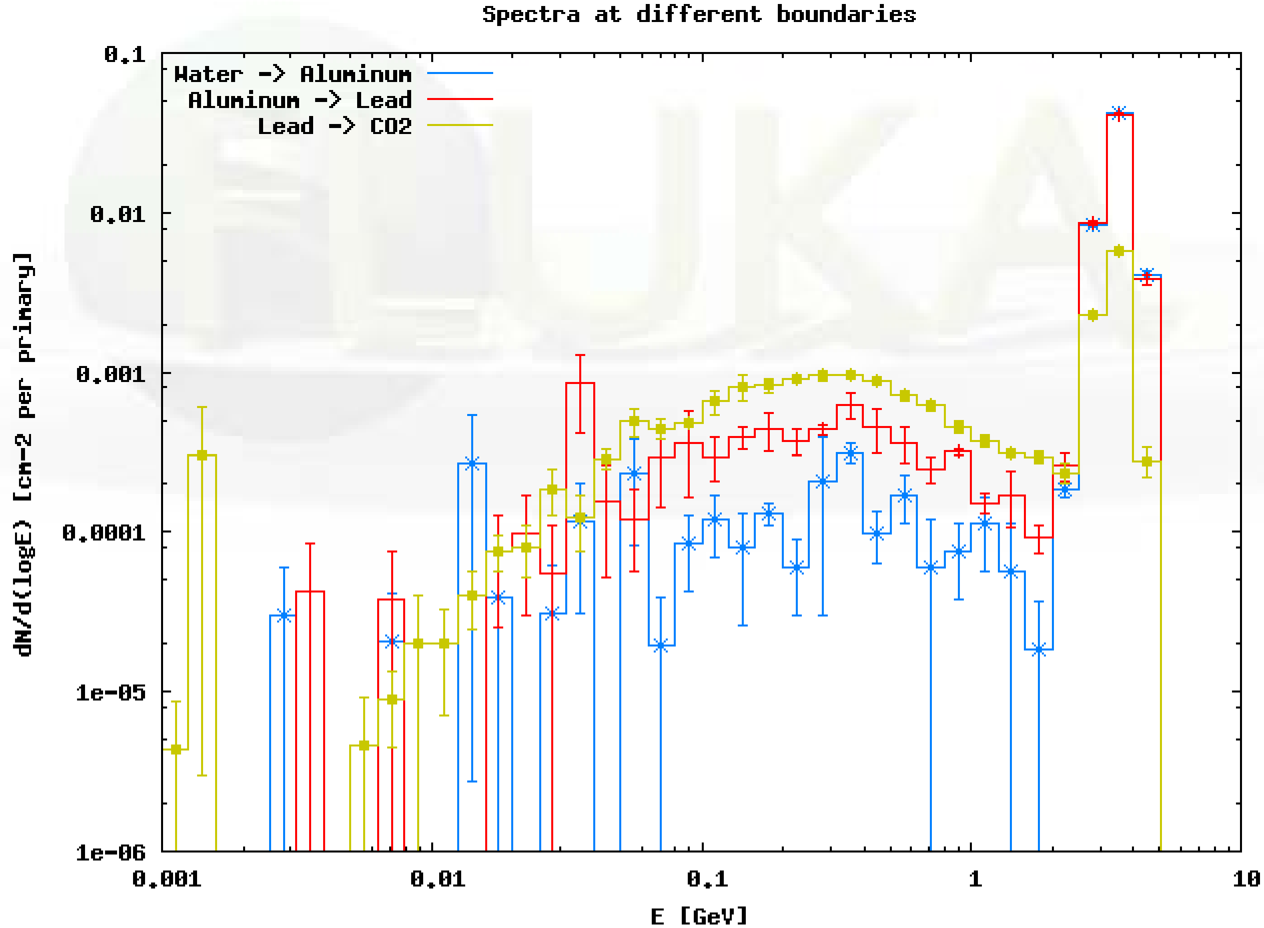
- The normalization is per cm^{-2} ONLY if the user provides the surface area or region volume. FLUKA is not able to calculate areas/volumes
- The same if USRBIN is used by region
- Instead, the normalization for USRBIN on grids is automatically in cm^{-2} (or cm^{-3} for deposited energy etc)

USRBDX → The Result

This is true only if the surface area is explicitly given

Evolution of charged hadron spectra at the various surfaces
post-processed results are normalized to GeV^{-1} cm^{-2} per primary

From post-processing we get **single** differential spectra
since we asked for one angular bin only



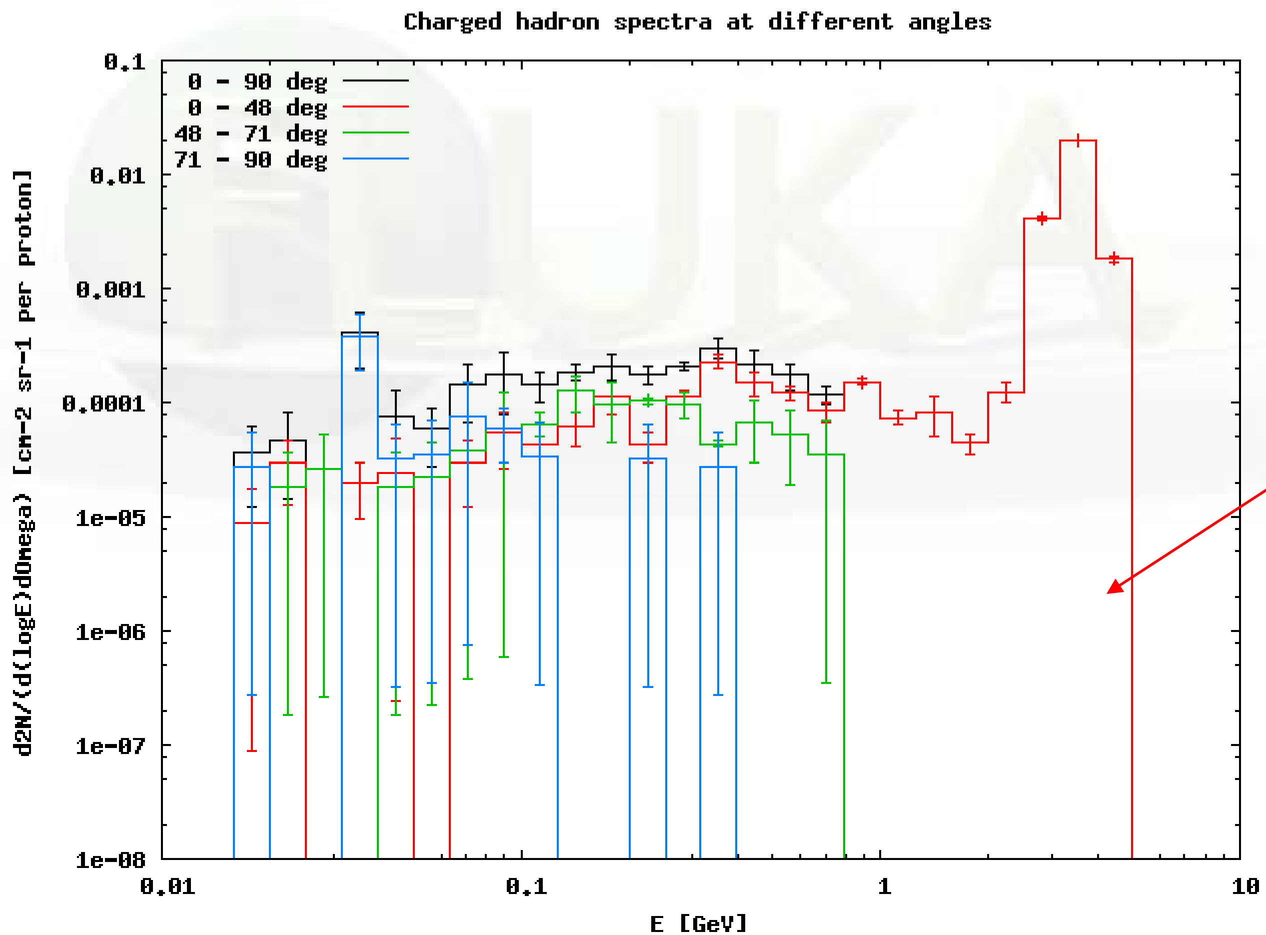
USRBDX → The Result

This is true only if the surface area is explicitly given

Double differential charged hadron spectra for consecutive solid angle portions results are normalized to GeV^{-1} sr^{-1} cm^{-2} per primary

From post-processing we get **double** differential spectra since we asked for more than one angular bin,

but the angle-integrated spectrum is provided as well on top



Forward is high energy

USRTRACK

- Calculates differential fluence as a function of energy by scoring track-length in a given region. Results are normalized to $GeV^{-1}\ cm^{-2}$ *per primary* if the region volume is provided (otherwise should be intended as GeV^{-1} *cm per primary*, i.e. differential track-length)

```
*   +    1    +    2    +    3    +    4    +    5    +    6    +    7    +
*                 log   partype  out.unit    region    volume      #bins
*                Emax      Emin
USRTRACK         -1.0   NEUTRON      -55.    TARGS3    628.31        40.TrChH
USRTRACK         10.0     0.001                                         &
```

- remember: USRBDX scores on a surface, while USRBIN scores fluence in volumes and gives no differential information
- WHAT(4) = @ALLREGS activates scoring over all regions

USRYIELD

- Scores a double-differential particle yield across a boundary surface or at interaction points. Angles wrt beam direction. Only 1 interval in the "second" quantity
- "Energy-like" quantities

Kinetic energy , total momentum , total energy , longitudinal momentum in the lab frame , longitudinal momentum in the c.m.s. frame , LET

- "Angle-like" quantities

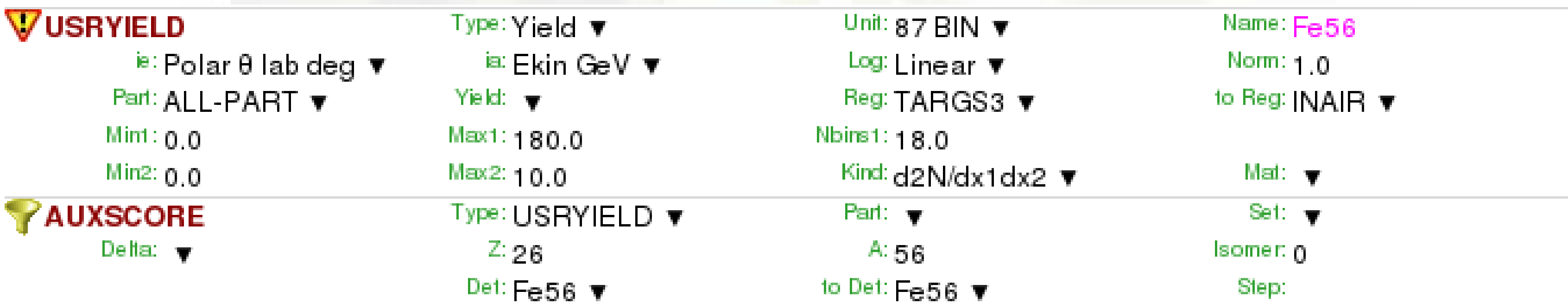
Rapidity in the lab frame , rapidity in the c.m.s. frame , pseudorapidity in the lab frame , pseudorapidity in the c.m.s. frame , Feynman-x in the lab frame , Feynman-x in the c.m.s. frame , transverse momentum , transverse mass , polar angle (*) in the lab frame , polar angle (*) in the c.m.s. frame , square transverse momentum , charge , weighted angle in the lab frame , weighted transverse momentum

FILTERS : AUXSCORE

There is the possibility to filter the estimators, restricting the scoring to a selected subset of particles.

For instance: : score the yield of 56-Iron ions (very useful: there is no separate name for each ion specie, except light ones. HEAVYION score all isotopes heavier than alpha's together!)

```
*    +    1    +    2    +    3    +    4    +    5    +    6    +    7    +
USRYIELD       124.0  ALL-PART       -87.    TARGS3     INAIR       1.0Fe56
USRYIELD       180.0       0.0        18.      10.0       0.0       3.0&
AUXSCORE    USRYIELD  -5602600.                Fe56      Fe56
```



The requested ion is coded in what(2) according to its **A**, **Z** and (optionally) isomeric state **m**:

what(2) = - (100***Z** + 100000***A** + **m***100000000)

Z,A,m=0 means all , e.g. -2600 == all Iron isotopes

Warning on AUXSCORE

- To be used with care, or NOT used at all for energy deposition scoring.
- Why? Because **In real world, energy is eventually deposited mostly by electrons only.**
- In MonteCarlo, part of the energy is deposited “by other particles” as continuous energy deposition or point energy deposition, depending on the delta-ray threshold, production threshold, transport threshold (see lecture on transport)
- Any filtering done with AUXSCORE on energy deposition will depend on the adopted thresholds and settings. (see lecture on transport)

Summary of commands follows,

FOR YOUR DOCUMENTATION

Summary of Scoring Commands

- USRTRACK, USRCOLL score 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;
- USRBIN scores the spatial distribution of energy deposited, or total fluence (or star density, or momentum transfer) in a regular mesh (cylindrical, Cartesian o per region) described by the user;
- USRYIELD scores a double differential yield of particles escaping from a surface. The distribution can be with respect to energy and angle, but also other more "exotic" quantities;
- SCORE scores energy deposited (or star density) in all regions;
- The output of SCORE will be printed in the main (standard) output, written on logical output unit LUNOUT (pre-defined as **11** by default)
- All other detectors write their results into logical output units assigned by the user (the unit numbers must be **>20**)

More “Special” Scoring

- RESNUCLEi scores residual 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)

Lethargy plots

- What is Lethargy? (in FLAIR : y*<x> plot option) , why to use it?
  - Using log bins, metrics on x-axis is log(E) : a constant length *dx* on this axis corresponds to a constant *d(log(E))* interval
  - the visual rendering of spectra can be misleading, because the same length on the x axis corresponds to dramatically different E intervals, thus the relative importance of different energy ranges is difficult to visualize:

Over *dx*= *d(log(E))*, $\Delta N = \frac{dN}{dE} * \Delta E = \frac{dN}{dE} * E_1 * (10^{dx} - 1)$ proportional to E

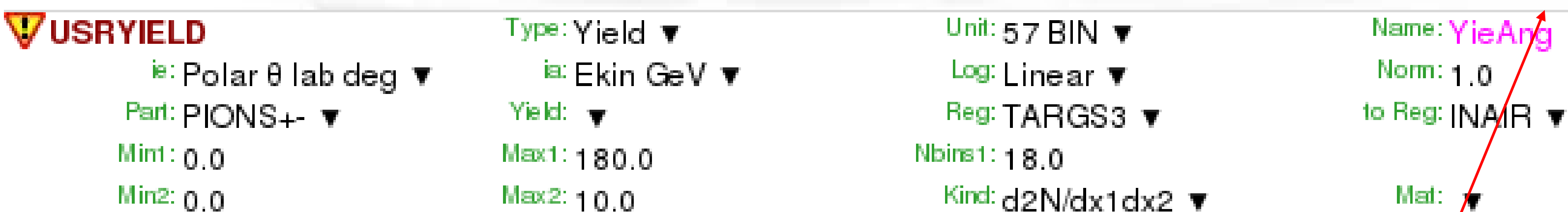
  - Metrics on x-axis is log(E), bin width is d(log(E)) → plot dN/d(log(E)) == dN/(1/E) == dN*E
  - In this way, the AREA corresponding to dx on the plot is proportional to the INTEGRAL of the spectrum over dx
  - A plot in dN*E is called Lethargy plot
  - Standard representation for Low Energy Neutrons spectra, also because of the 1/E dependence of moderated n spectra

WARNING!! calculating a cross section has no meaning in case of a thick target.

USRYIELD

- While option USRBDX calculates angular distributions WITH RESPECT TO THE NORMAL to the boundary at the point of crossing, USRYIELD's distributions are calculated WITH RESPECT TO THE BEAM DIRECTION, as defined by BEAMPOS (or a different direction specified with SDUM=BEAMDEF).

```
*    124 = 24 + 1 * 100 => polar angle (in degrees) and kinetic energy
*                                  out.unit           Reg1      Reg2      Norm
*               Amax      Amin    #Abins           Emax      Emin dbl.differential
*   +    1    +    2    +    3    +    4    +    5    +    6    +    7    +
USRYIELD       124.0   PIONS+-       -57.    TARGS3     INAIR       1.0YieAng
USRYIELD       180.0       0.0        18.      10.0       0.0       3.0&
```

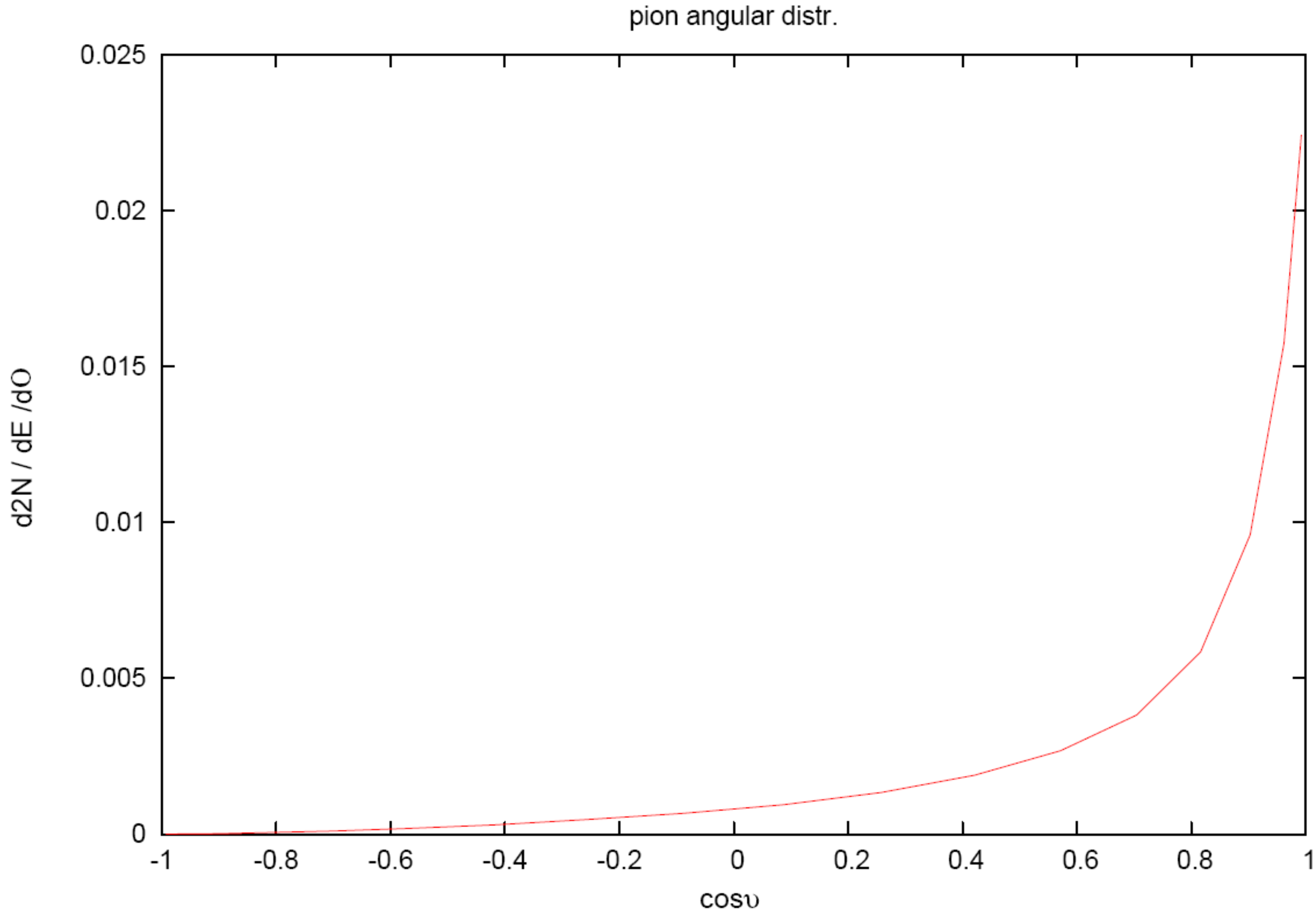


- Only one interval is possible for the second variable, BUT results are normalized as Double Differential (in this case, charged pions GeV^{-1} sr^{-1} per primary)

WARNING!!
Use WHAT(6) = 3 for plain double differential yield, the DEFAULT is plain double-differential cross section !!

USRYIELD -> The Result

- pion angular distribution



Standard Postprocessing Programs

- Example of sum.lis from USRYIELD

Detector n: 1 YieAng
(User norm: 1.
sigma: 1. mb
distr. scored: 209 , PIONS+-
from reg. 4 to reg. 5)
linear 1st variable (x1) binning from 0.0000E+00 to 3.1416E+00 18 bins
(1.7453E-01 wide)
2nd variable (x2) ranges from 0.0000E+00 to 1.0000E+01
1st variable (x1) is: Laboratory Angle (radians)
2nd variable (x2) is: Laboratory Kinetic Energy
The scored double differential yield is (normalized per primary particle):
plain d2 N / dx1 dx2 where x1, x2 are the first and second variables

automatic conversion from degrees

Tot. response (integrated over x1) 2.6339998E-02 +/- 3.883959 %

WARNING!! The Tot. response is NOT integrate over the second quantity !!
in this case it turns out to be *particles/GeV per primary*
(to be multiplied by the energy width interval of 10GeV)

Standard Postprocessing Programs

- Example of tab.lis for USRYIELD

Detector n: 1 YieAng
N. of x1 intervals 18

Thetamin	Thetamax	Result	Error (%)
0.000000	0.1745329	2.0742605E-02	10.87912
0.1745329	0.3490658	1.4463779E-02	10.65940
0.3490658	0.5235988	9.8084798E-03	7.649231
0.5235988	0.6981317	5.8580171E-03	4.966214
0.6981317	0.8726646	3.8220894E-03	10.60832
0.8726646	1.047198	2.6973977E-03	5.450788

...