Estimators and Scoring

23rd FLUKA Beginner's Course
Lanzhou University
Lanzhou, China

June 2–7, 2024

FLUKA Scoring & Results - Estimators

- It is often said that **Monte Carlo** (MC) is a **"mathematical experiment"**, so 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.

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).

FLUKA Scoring & Results

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**

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: 21-512
- Generally, the user can choose between formatted (ASCII) and unformatted (binary) scoring (positive or negative 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

Reaction Rate and Cross Section

Definitions:

- λ[cm]: mean free path, the average distance travelled by a particle in a material before an interaction. It depends on the material, particle type and energy.
- $\Sigma[\mathrm{cm}^{-1}] = 1/\lambda$[cm]: macroscopic cross section, the probability of interaction per unit distance. It depends on the material, particle type and energy.
- N_0: number of atoms in volume [atoms/(cm barn)]
- $\sigma = \frac{\Sigma}{N_0}$ [barn = $10^{-24}\mathrm{cm}^2$]: microscopic cross section (atom effective area)
    i) the area of an atom weighted with the probability of interaction (hence the name "cross section")
    ii) 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 base of 1cm^2)

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 interaction can be obtained by multiplying both by the corresponding particle track-length.

Reaction Rate and Cross Section

More definitions:

- $R = N\, l\, \Sigma$: The number of reactions for N identical particles along a total distance l.
- $\dot{R} = N\, \frac{dl}{dt}\, \Sigma\ = N\, v\, \Sigma$ [1/s]: Reaction rate, where v is the average particle velocity.
- $\frac{d\dot{R}}{dV} = \frac{dN}{dV}\, v\, \Sigma = n(\mathbf{r}, v)\, v\, \Sigma$ [1/(cm^{3}s)]: Reaction rate inside a volume element dV, with $n(\mathbf{r}, v)$ the density of particles with velocity v at position $\mathbf{r}$.
- $\dot{\Phi}(\mathbf{r}, v) = n(\mathbf{r}, v)\, v$ [1/(cm^{2}s)]: Fluence rate or Flux density.
- $\Phi = \int \dot{\Phi}(\mathbf{r}, v) dt = n(\mathbf{r}, v)\, l$ [1/(cm^2)]: Fluence, the time integral of the flux density.

Fluence is measured in particles per cm^2 but in reality it describes the density of particle tracks.

Fluence estimation

There are several ways to estimate the fluence:

- Track length estimation

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

Sum the length of all particle tracks inside a volume **USRTRACK**

- Collision density estimation

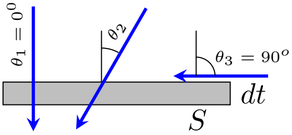
$$\dot{\Phi}(v) == \frac{\dot{R}(v)}{\sigma(v)\, N_0\, \Delta V} = \frac{\dot{R}(v)}{\Sigma(v)\, \Delta V} = \frac{\dot{R}(v)\lambda(v)}{\Delta V}$$

Count the collisions (weighted by the mean free path) inside a volume

USRCOLL

Fluence estimation

- 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:

$$J = \frac{dN}{dS}$$

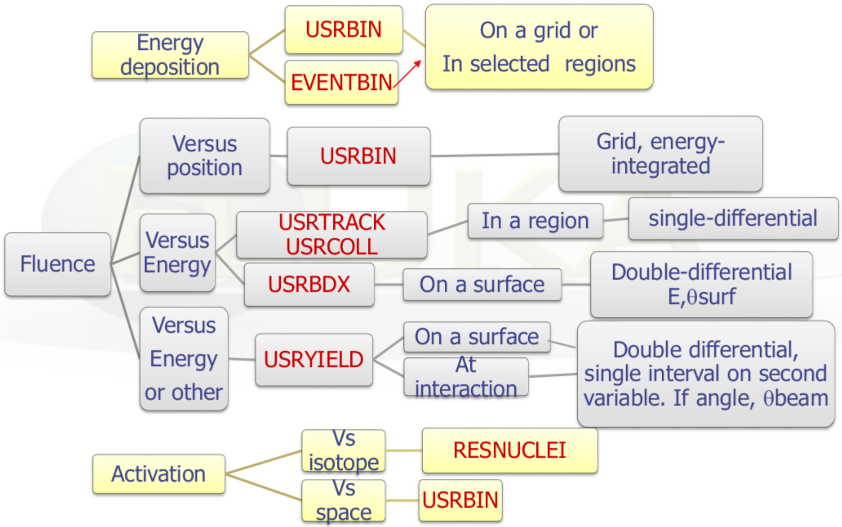
Fluence vs. current

The fluence is independent from the orientation of surface S
while the current is NOT!
In an isotropic radiation field it can be easily seen that on a flat surface

$$J = \Phi/2$$

Current is *meaningful* if particles are only counted (e.g. signal trigger).
But if someone wants to estimate dose, activation, radiation damage or instrument response the quantity to be used is fluence.

FLUKA estimators



FLUKA estimators

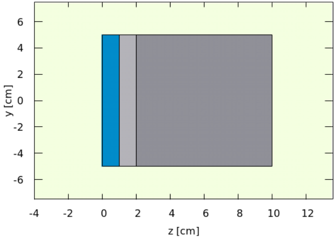
- USRTRACK, USRCOLL score average $d\Phi/dE$ (differential fluence) of a given type or family of particles in a given region;
- USRBDX scores average $d^2\Phi/dEd\Omega$ (double-differential fluence or current) of a given type or family of particles on a given surface;
- USRBIN scores the spatial distribution of deposited energy, total fluence, DPAs, activation, star density, momentum transfer or dosimetric quantities related to cell survival in a regular mesh (cylindrical, Cartesian oder per region) described by the user
- USRYIELD scores double differential yield of particles escaping from a surface. Distribution can be with respect to energy and angle (but also other quantities are possible);
- SCORE scores energy deposited (or star density) in all regions;
  - The output of SCORE will be printed in the main (standard) output, writen on main (standard) output, written o logical output unit LUNOUT (pre-defined as **11** by default)
- Scoring detectors write their result into logical output units assigned by the user (unit numbers must be **>20**)

Additional FLUKA estimators and configuration cards

- RESNUCLEi scores residual nuclei in a given region (more details in lecture on activation);
- DETECT scores energy deposition in one or more regions in coincidence or anti-coincidence with a trigger separately for each "event" (primary history) to logical output unit **17**;
- EVENTBIN works like USRBIN, but prints the binning output after each event (instead of averaging over event histories);
- USERDUMP defines the events to be written to a "collision tape" file;
- AUXSCORE defines filters (for particle types) and conversion coefficients (e.g. fluence-to-dose conversion coefficients);
- DCYSCORE assigns irradiation and cooling times (see lecture on activation);
- ROTPRBIN sets storage precision to single or double precision and assigns rotations/translations to USRBIN or EVENTBIN scorers (useful in case of LATTICEs);
- TCQUENCH sets scoring time cut-offs and/or Birks quenching parameters for USRBIN or EVENTBIN binnings.

Extending the Geometry example with scoring

- Result of Geometry exercise: Cylinder along Z, filled by water-aluminum-lead and surrounded by CO_2



- The USRBIN command allows to define a 3-D grid, either cartesian (X-Y-Z), 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!).

USRBIN-Card

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

Example: WHAT(1) = 11.0 → Mesh: R-Z or R-Φ-Z, no symmetry

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

Examples: NEUTRON for neutrons, HAD-CHAR for charged hadrons

WHAT(3): logical output unit, (|WHAT(3)| >21 recommended)

>0 formatted (human readable) output,

<0 unformatted (machine readable) output

The first three fields have similar meanings for all estimators. The other WHATs contain the limits for regions or boundaries, and the number of intervals for cells/histograms (see manual).

For USRBIN with R-Φ-Z binning:

WHAT(4): R_{max}

WHAT(5): y coordinate of binning axis

WHAT(6): Z_{max}

SDUM: estimator name (optional)

Continuation card:

WHAT(1,3): R_{min}, Z_{min}

WHAT(2): x coordinate of binning axis

WHAT(4-6): Number of bins in R, Φ, Z

SDUM: & (continuation character)

Exercise: USRBIN

- Start from the solution of Geometry_exercise (either copy your `.inp` and .flair files and rename them to `example_score`, or download/copy the file `Geometry_exercise/geometry_solution.inp` and rename it):

```
mkdir example_scoring
cp Geometry_exercise/geometry_solution.inp example_score/ex_Score.inp
cd example_scoring
```

- Open in FLAIR or with your preferred editor
- Add USRBIN scoring to:
  1) score Energy on a CYLINDRICAL GRID (R, Φ, Z covering the target and surroundings: $0 < r < 10$ cm, $-5 < z < 15$ cm, with cells having $\Delta r = \Delta z = 1$ mm, $\Delta\Phi = 2\pi$, Output unit = 40 BIN
  2) score Neutron Fluence on the same grid, Output unit = 41 BIN
  3) score Charged Hadron Fluence on the same grid, Output unit = 42 BIN
- Run 5 cycles, 1000 primaries each

Exercise: USRBIN

1) Add USRBIN to score ENERGY on a CYLINDRICAL GRID (R, Φ, Z) covering the target and the surroundings: $0 < r < 10$ cm, $-5 < z < 15$ cm, with cells having $\Delta r = \Delta z = 1$ mm, $\Delta\Phi = 2\pi$

```
* Energy deposition [GeV/cm^3]
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7...
USRBIN           11.    ENERGY      -40.      10.0                15.0TargEne
USRBIN           0.0                -5.0      100.                200. &
```

USRBIN

		Unit: 40 BIN ▾	Name: TargEne
Type: R-Φ-Z ▾	Rmin: 0.0	Rmax: 10.0	NR: 100.
Part: ENERGY ▾	X:	Y:	NΦ:
	Zmin: -5.0	Zmax: 15.0	NZ: 200.

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

Exercise: USRBIN

2) Add USRBIN to score Neutron Fluence on the same grid as before:

```
* Neutron fluence [1/cm^2]
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7...
USRBIN           11.    NEUTRON      -41.       10.0                15.0TargNeu
USRBIN           0.0                 -5.0       100.                200. &
```

USRBIN			
		Unit: 41 BIN ▾	Name: TargNeu
Type: R-Φ-Z ▾	Rmin: 0.0	Rmax: 10.0	NR: 100.
Part: NEUTRON ▾	X:	Y:	NΦ:
	Zmin: -5.0	Zmax: 15.0	NZ: 200.

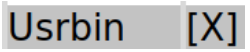
- This is an R-Φ-Z binning (WHAT(1) = 11.), scoring neutron fluence (particle NEUTRON, or FLUKA PID 8), writing unformatted output to unit 41, spanning $0 < R < 10$ cm in 100 bins, $0 < \Phi < 2\pi$ in 1 bin (default), $-5 < Z < 15$ cm in 200 bins.

3) Add USRBIN to score Charged Hadron Fluence (`HAD-CHAR`) on the same grid as before, writing to output unit = 42 BIN.
   - You can use the Clone button in FLAIR to clone the previous scorer, then modify Unit: and Part: accordingly
   - Or simply the copy the two lines in your favorite text editor...

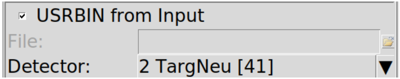
Exercise: USRBIN - visualisation in FLAIR

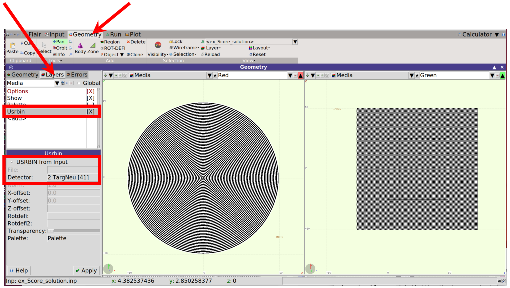
You can display the USRBIN-scorer you just defined in FLAIR's "Geometry" menu:

- In the "Geometry"-menu of FLAIR, chose the "Layers"-tab and select the desired layer (e.g. "media")
- Click "<add>", then mark "USRBIN"



- Mark "USRBIN from Input" and select the desired scorer from the drop-down menu

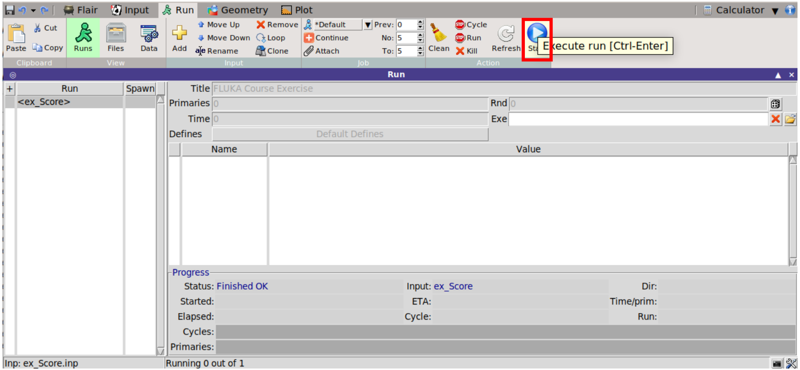




After you ran the example (next slide), deselect "USRBIN from Input", and instead select the *.bnn-file and plot the results inside the FLAIR Geometry-Menu.

Exercise: USRBIN

Run the input file by clicking the Start button in the "Run"-menu of FLAIR:



Or execute FLUKA on the commandline:

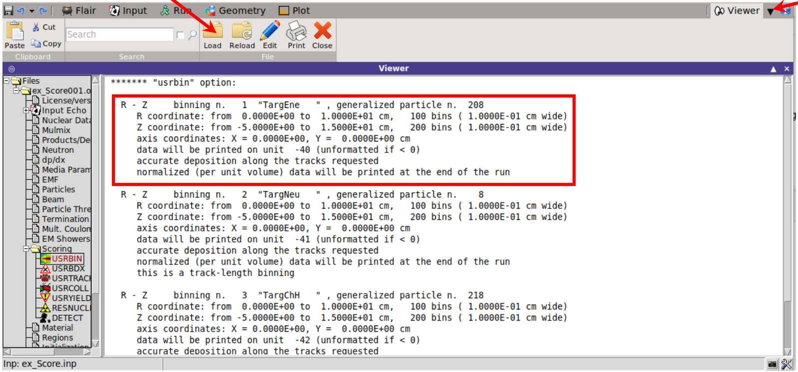
```
$FLUPRO/flutil/rfluka  -e  $FLUPRO/flukahp  -N0 -M5  ./ex_Score
```

Exercise: USRBIN

Load the `.out`-file in the "Viewer" of FLAIR - it gives a complete description of each requested estimator:

Load file

Switch to "Viewer"

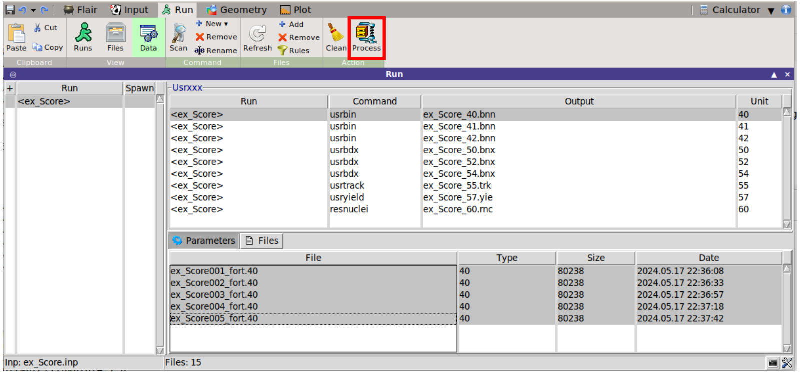


Alternatively, check the `.out`-files on the command line by doing

```
less ex_Score001.out
```

Exercise: USRBIN - Processing of the data

Select the desired scorer in the “Run” → “Data” tab, add the corresponding `*_fort.xx`-files and click the Process button:

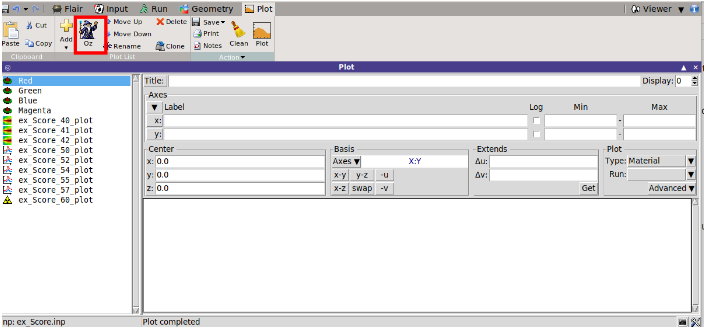


Or execute the post-processing script on the commandline:

`$FLUPRO/flutil/usbsuw` (then follow the instructions)

Exercise: USRBIN - Plotting of the data

Plots can be created in the "Plot" menu - add new plots or clone from existing ones:



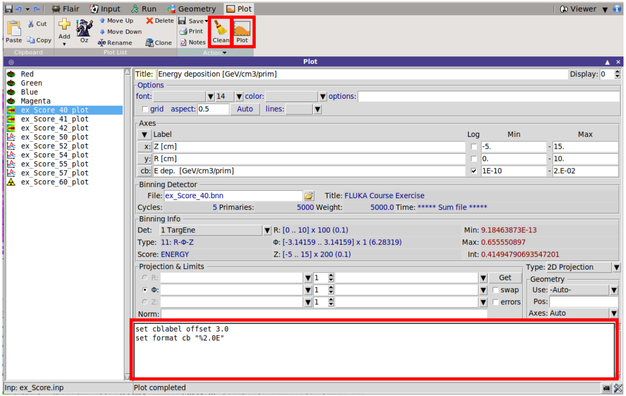
- The "Wizard" button scans the input and creates automatically a plot for each processed unit
- Set a unique filename for each plot
  - This filename will be used for auxiliary files that the plot needs

Plot types:

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

Exercise: USRBIN - Plotting of the data

FLAIR uses **GNUPLOT** to create the plots:

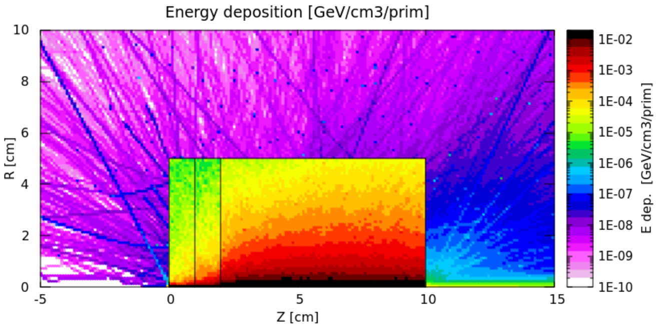


- For all plots one can specify: Title + options, filename, axis labels, legends, Gnuplot commands
- The button (Ctrl-Enter) will generate all the necessary files (if they don't exist yet) and produce the plot
- The button will remove al files generated by FLAIR during plotting (useful when the plot name was changed)

- Additional GNUPLOT commands can be specified in the white field, e.g.:
  - Change colorband label offset
  - Change format of colorband (cb) palette values to "`%2.0E`"

Exercise: USRBIN → The result

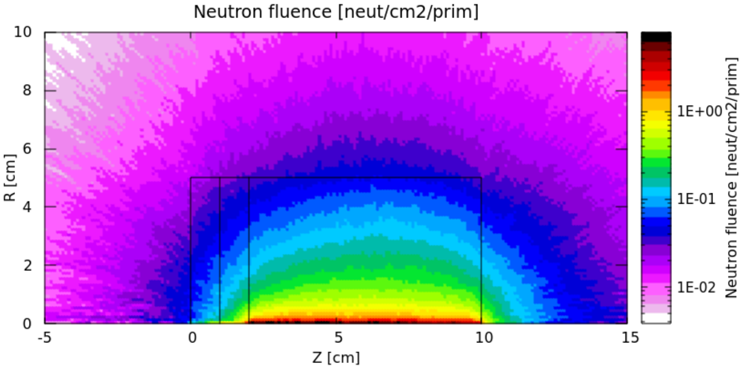
WHAT(2) = ENERGY: Energy deposition from a 3.5 GeV proton beam hitting at [0., 0., 0.] directed along z. Results are normalized to **GeV/cm^3 per primary**



This plot is a 2D projection of a 3D structure → the result is averaged over the 3rd coordinate. Projection limits can be set in FLAIR.

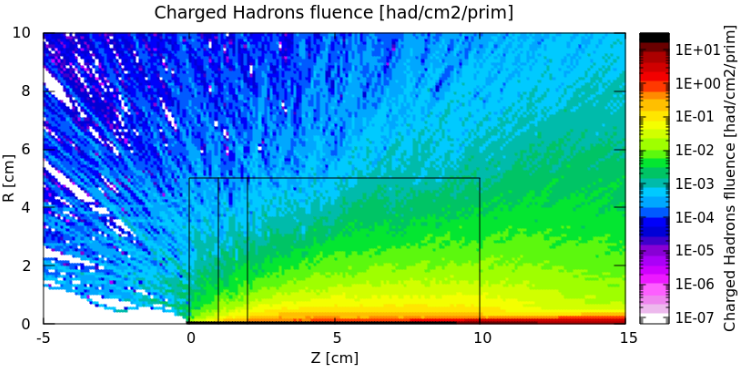
Exercise: USRBIN $\rightarrow$ The result

Same for WHAT(2) = NEUTRON: Neutron fluence from a 3.5 GeV proton beam hitting at [0., 0., 0.] directed along z. Results are normalized to **neutrons/cm^2 per primary**



Exercise: USRBIN $\rightarrow$ The result

Same for WHAT(2) = HAD-CHAR: Charged hadron fluence from a 3.5 GeV proton beam hitting at [0., 0., 0.] directed along z. Results are normalized to **charged hadrons/cm^2 per primary**



USRBIN can score particle fluence as well as "generalized particles", either fluence-like or energy-like, for instance with WHAT(2) given as

- DOSE — Energy/uni mass [GeV/g]
- DPA-SCO — Displacements per atom
- DOSE-EQ — Dose equcalent [pSv] (needs AUXSCORE card to set fluence-to-dose conversio coefficients
- X-MOMENT — x-component of momentum transfer [GeV/c]
- ACTIVITY — activity per unit volume [Bq/cm^3] (needs DCYTIMES and DCYSCORE cards to specify decay times)
- ...and more (see FLUKA manual)

Exercise: USRBDX

USRBDX ("User Boundary Crossing" scoring) scores double differential (energy and angle) particle distributions across a boundary surface. The angle is with respect to the normal of the surface. The angular distributions must be intended as distributions in solid angle $\Omega = 2\pi(1 - \cos\theta)$, where θ is the angle between the particle trajectory and the normal to the boundary at the point of crossing.

Input card:

- WHAT(1): distribution can be fluence or current, one- or two-way, according to the value of WHAT(1) (see manual)
- WHAT(2): particle type to be scored
- WHAT(3): logical output unit
- WHAT(4): first region defining the boundary
- WHAT(5): second region defining the boundary
- WHAT(6): area of the detector in cm^2
- SDUM: detector identifier

Continuation card:

- WHAT(1): maximum kinetic energy for scoring [GeV]
- WHAT(2): minimum kinetic energy for scoring [GeV]
- WHAT(3): number of energy intervals
- WHAT(4): maximum solid angle in steradians
- WHAT(5): minimum solid angle (linear angular binning) or solid angle of first bin (logarithmic anglular binning)
- WHAT(6): number of angular bins
- SDUM: & (continuation character)

Exercise: USRBDX

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

```
*  charged hadron fluence exiting lead target
USRBDX           99.  HAD-CHAR      -50.    TARGS3     INAIR    329.87Sp3ChH
USRBDX           10.     0.001       40.                                  &
```

charged hadron fluence exiting lead target

USRBDX

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

Repeat the same between TARGS1 and TARGS2, and between TARGS2 and TARGS3 (take care to use the correct normalization factor!).

Exercise: USRBDX

2) Score CHARGED HADRONS at the surface between 2nd and 3rd section, but in 3 angular bins:

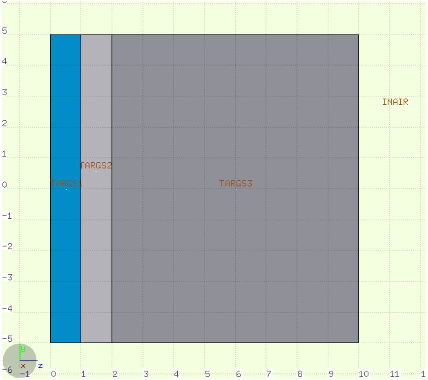
```
*  double-differential charged hadron fluence entering lead target
USRBDX           99.   HAD-CHAR      -54.     TARGS2    TARGS3   78.5398Sp2ChHA
USRBDX           10.      0.001       40.                               3.  &
```

double-differential charged hadron fluence entering lead target

USRBDX

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

Exercise: USRBDX - Normalization



R_{TARG} = 5 cm
Z_{TARGS1} = Z_{TARGS2} = 1 cm
Z_{TARGS1} = 8 cm

Area between TARGS2 and TARGS3:
$\pi\ R_{TARG}^{\ 2}$ = **78.5398 cm²**

Area between TARGS3 and INAIR:
$\pi\ R_{TARG}\ Z_{TARGS3} + \pi\ R_{TARG}^{\ 2}$ = **329.87 cm²**

Exercise: USRBDX - Postprocessing

The FLUKA postprocessing program for USRBDX, $FLUPRO/flutil/usxsuw generates files with extension **_sum.lis** (human readable) and **_tab.lis** (slightly more machine readable). Invoke it by clicking the Process button in FLAIR, or execute the post-processing script on the commandline: `$FLUPRO/flutil/usxsuw` (then follow the instructions)

Example of _sum.lis from USBRDX:

```
 Total primaries run:              5000
  Total weight of the primaries run:   5000.00000
 Detector n:              1 (            1 )  Sp1ChH
    (Area:             78.5398026        cmq,
     distr. scored:          218      ,  Charged Hadrons
     from reg.            2  to           3 ,
     one way scoring,
     fluence scoring scoring)

    Tot. resp. (Part/cmq/pr)  1.3232824E-02  +/-   0.3747608     %
    ( -->      (Part/pr)       1.039303      +/-   0.3747608     % )

 **** Different. Fluxes as a function of energy **** ****      (integrated over solid angle)        ****
 (...list of data points...)
 **** Cumulative Fluxes as a function of energy **** ****      (integrated over solid angle)        ****
(...list of data points...)
 **** Double diff. Fluxes as a function of energy ****
(...list of data points...)
```

User-provided area in WHAT(6), thus normalization in cm^{-2}

Exercise: USRBDX - Postprocessing

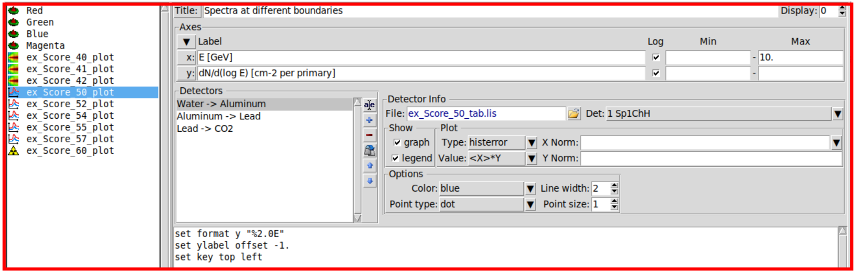
- Example of _tab.lis for USRBDX:

```
# Detector n:            1    Sp1ChH    (integrated over solid angle)
 # N. of energy intervals         40
     Emin         Emax        Result     Error (%)
  1.000E-02  1.259E-02  1.781E-03  9.900E+01
  1.259E-02  1.585E-02  0.000E+00  0.000E+00
  1.585E-02  1.995E-02  6.536E-04  9.900E+01
  1.995E-02  2.512E-02  7.069E-03  7.818E+01
  2.512E-02  3.162E-02  1.320E-03  6.309E+01

 # double differential distributions
 # number of solid angle intervals          1
 # Block n:           1      0.00000000      :    6.28318548
  1.000E-01  1.259E-01  3.060E-04  2.570E+01
  1.259E-01  1.585E-01  3.144E-04  2.570E+01
  1.585E-01  1.995E-01  7.765E-05  2.369E+01
  1.995E-01  2.512E-01  1.959E-04  2.605E+01
  2.512E-01  3.162E-01  1.078E-04  3.540E+01
```

Exercise: USRBDX - USR-1D Plotting

Plot → Add → USR-1D (or using the "Wizard" button)



- USR-1D plots the 1D single differential information from USRBDX, USRCOLL, USRTRACK and USRYIELD cards
- File type should be _tab.lis (generated by FLUKA data merging tool `usxsuw`)
- Many scoring outputs can be superimposed on the same plot
- Plot error bars using "histerror" or "xyerrorbars"
- Use a **lethargy** plot Value: <X>*Y to plot the spectrum

Exercise: USRBDX - Results

The result shows the evolution of charged hadron spectra at the different surfaces.

post-processed results are normalized to GeV^{-1} cm^{-2} per primary (only if surface area is explicitely given).

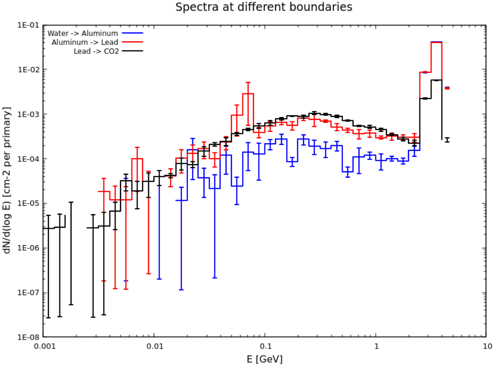
- **Lethargy plots** are used to display spectra where the energy extends over may orders of magnitude

$$y = \frac{dn}{d(logE)} = E\frac{dN}{dE}$$

Value: <X>*Y

- In this way, the area of each bin is proportional to the corresponding integral flux, giving an immediate feeling which energy bin or region contributes more or less particles.

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



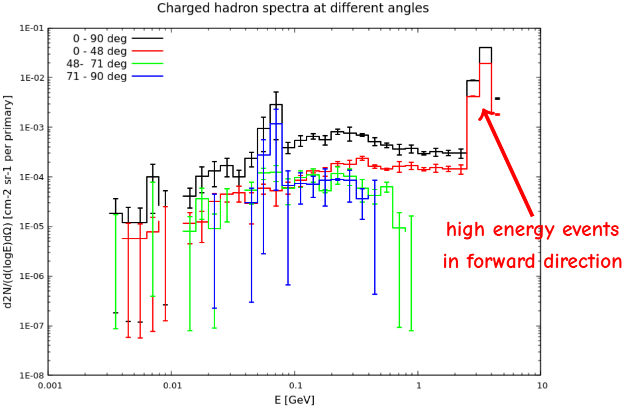
Exercise: USRBDX - Results

Double-differential charged hadron spectra for 3 consecutive solid angle bins results are normalized to **GeV^{-1} sr^{-1} cm^{-2} per primary** (only if surface area is explicitely given)

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



USRTRACK

USRTRK calculates differential fluence as a function of energy bz scoring track-length in a given region. Results are normalized to GeV^{-1} cm^{-2} per primary particle if the region volume is provided (otherwise should be intended as GeV^{-1} cm per primary particle, i.e. differential track-length). For all SDUMs except EN-NUCL and ENERGY:

WHAT(1): linear (=1) or logarithmic (=-1) energy binning
WHAT(2): particle type to be scored
WHAT(3): logical output unit
WHAT(4): region defining the detector (if >0) or all regions (if =-1)
WHAT(5): volume of the detector in cm^3
WHAT(6): number of energy bins
SDUM: track-length detector identifier

Continuation card:

WHAT(1): maximum kinetic energy for scoring [GeV]
WHAT(2): minimum kinetic energy for scoring [GeV]
SDUM: & (continuation character)

For SDUM=EN-NUCL: The energy scale will be changed from energy to energy per nucleon.
For SDUM= ENERGY: the energy scale will be changed to the default, that is total kinetic energy.

Exercise: USRTRACK

1) Score the track-length of CHARGED HADRONS in TARGS3, with logarithmic binning in energy (WHAT(1)=-1) using 40 bins between 1 MeV and 10 GeV. Normalize with the region volume in order to have the results in GeV^{-1} cm^{-2} per primary particle. Write the output to unformatted unit 55:

```
*  charged hadron fluence in lead target
* ..+....1....+....2....+....3....+....4....+....5....+....6....+....7...
USRTRACK         -1.   HAD-CHAR      -55.     TARGS3    628.31        40.TrChH
USRTRACK         10.      0.001                                          &
```

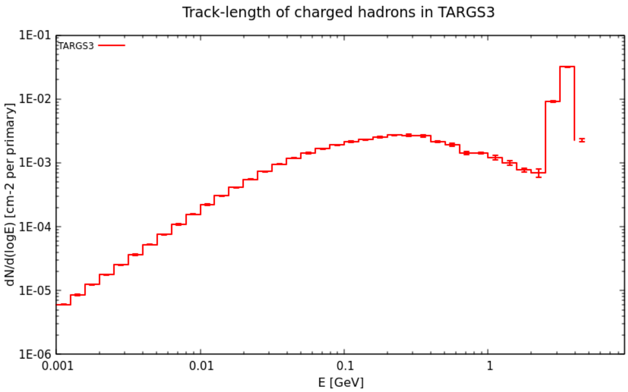
charged hadron fluence in lead target

USRTRACK		Unit: 55 BIN ▾	Name: TrChH
Type: LogE,Groupwise ▾	Reg: TARGS3 ▾		Vol: 628.31
Part: HAD-CHAR ▾	Emin: 0.001	Emax: 10.	Bins: 40.

Remember: USRTRACK scores differential fluence in a region, USRBDX scores fluence or current on a surface, and USRBIN scores e.g. fluence in volumes and gives no differential information.

Exercise: USRTRACK - Result

Track-length based fluence of charged hadrons in region TARGS3, plotted as a lethargy plot:



USRYIELD

- Scores a double differential particle yield across a boundary surface or at interaction points. The angles are defined respect to the beam direction. Only 1 interval is used in the second quantity (use several USRYIELD cards for more intervals in 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

USRYIELD

- While USRBDX scoring calculates angular distributions with respect to the normal to the boundary at the point of crossing, USRYIELD's distribution are calculated with respect to the beam direction, as defined by the BEAMPOS card (use an additional USRYIELD card with SDUM=BEAMDEF to define a different direction).

Input card:

WHAT(1): definition of two physical quantities
WHAT(2): particle type to be scored
WHAT(3): logical output unit
WHAT(4): first region defining the boundary
WHAT(5): second region defining the boundary
WHAT(6): normalization factor
SDUM: detector name

Continuation card:

WHAT(1): upper limit for the first quantity
WHAT(2): lower limit for the first quantity
WHAT(3): number of scoring intervals (bins) for first quantity
WHAT(4): upper limit for the second quantity
WHAT(5): lower limit for the second quantity
WHAT(6): kind of yield or cross section, and/or target material (see manual)
SDUM: & (continuation character)

Exercise: USRYIELD

1) Score plain double-differential yield (continuation card WHAT(6)=3) for pions, with the first quantity polar angle (degree) and second quantity kinetic energy (WHAT(1)=124), between TARGS3 and INAIR, between 0 and 180 degrees in 18 bins and between 0 and 10 GeV:

```
*  charged pion angular distribution exiting lead target
USRYIELD        124.   PIONS+-      -57.    TARGS3     INAIR        1.YieAng
USRYIELD        180.       0.0       18.       10.       0.0        3.  &
```

Remember: Only one interval is possible for the second variable, but results are normalized as double-differential quantities (in this case, charged pions yield in GeV^{-1} sr^{-1} per primary).

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

USRYIELD - Postprocessing

The FLUKA postprocessing program for USRYIELD scorers, **$FLUPRO/flutil/usysuw**, generates generates files with extension **_sum.lis** and **_tab.lis**.

Example of _sum.lis from USRYIELD:

automatic conversion from degrees

```
 Detector n:            1   YieAng
    (User norm:          1.00000000
     sigma:              1.00000000       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 / d x1 d x2 where x1, x2 are
       the first and second variables

     1st variable (x1) differential is a solid angle one

    Tot. response (integrated over x1)   2.6879998E-02  +/-   2.035619     %
    ( --> Tot. response without normaliz.   2.6879998E-02  +/-   2.035619     % )
```

Warning! The total response is not integrated 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 10 GeV)

USRYIELD - Postprocessing

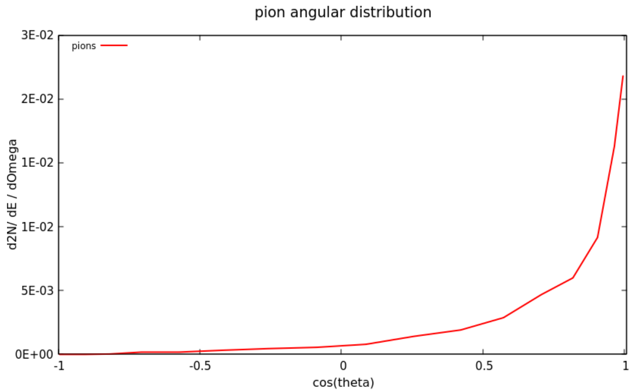
Example of _tab.lis from USRYIELD:

```
# Detector n:            1   YieAng
 # N. of x1 intervals          18
```

$Theta_{min}$	$Theta_{max}$	Result	Error (%)
0.000000	0.1745329	2.1790212E-02	5.133210
0.1745329	0.3490658	1.6298208E-02	5.450077
0.3490658	0.5235988	9.1603426E-03	5.757809
0.5235988	0.6981317	5.9853652E-03	1.356122
0.6981317	0.8726646	4.6743122E-03	7.381429
0.8726646	1.047198	2.8757381E-03	3.952729
1.047198	1.221730	1.9141326E-03	12.11670
1.221730	1.396263	1.4178869E-03	8.692270
1.396263	1.570796	7.8822160E-04	18.23766
1.570796	1.745329	5.4992206E-04	20.41241
1.745329	1.919862	4.5372383E-04	23.19902
1.919862	2.094395	3.2238022E-04	18.22172
2.094395	2.268928	1.7834034E-04	54.48624
2.268928	2.443461	1.8077450E-04	28.57143
2.443461	2.617994	3.1837051E-05	99.00000
2.617994	2.792527	0.000000	0.000000
2.792527	2.967060	0.000000	0.000000
2.967060	3.141593	0.000000	0.000000

USRYIELD - Result

Pion angular distribution:



Use **gnuplot** commands to plot with FLAIR:

```
set key top left
set format y "%2.0E"
set ylabel offset -1.

plot 'ex_Score_57_tab.lis' ind 0 us (cos(($1+$2)/2.0)):3 w lines lw 2 lc rgb 'red' t ' pions'
```

Plot completed

Bins, units, normalization

- 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
  1) results are independent on the chosen binning
  2) to obtain integral results (fluence in cm^{-2} per primary for that energy interval) one needs to multiply each value by the width of the energy interval or bin (even for logarithmic binning):

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

- When scoring neutrons with groupwise treatment, energy bin limits below 20 MeV are automatically set to the neutron transport group limits (see lecture on low-energy neutrons), overwriting the user-defined binning
- Normalization is per cm^{-2} only if the user provides the surface area or region volume - FLUKA in general is not able to calculate areas or vlumes by itself
- The same holds if USRBIN scoring is used with region binning
- Only for USRBIN scoring on grids is the normalization automatically in cm^{-2} (or cm^{-3} for deposited energy etc.)

FILTERS: AUXSCORE

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

For instance, to score the yield of 56-Iron ions (there is no separate name for each ion specie, except the light ones like 4-HELIUM, 3-HELIUM, TRITON, DEUTERON. HEAVYION scores all isotopes heavier than alpha-particles together)

```
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....
USRYIELD        124.  ALL-PART      -58.    TARGS3     INAIR         1.Fe56
USRYIELD        180.       0.0       18.       10.       0.0         3.  &
AUXSCORE    USRYIELD -5602600.                Fe56      Fe56
```

USRYIELD	Type: Yield ▾	Unit: 58 BIN ▾	Name: Fe56
ie: Polar θ lab deg ▾	ia: Ekin GeV ▾	ip: Groupwise ▾	Log: Linear ▾
Norm: 1.	Part: ALL-PART ▾	Yield: ▾	Reg: TARGS3 ▾
to Reg: INAIR ▾	Min1: 0.0	Max1: 180.	Nbins1: 18.
	Min2: 0.0	Max2: 10.	Kind: d2N/dx1dx2 ▾
AUXSCORE	Type: USRYIELD ▾	Part: ▾	Set: ▾
Delta Ray: ▾	Z: 26	A: 56	Isomer: 0
	Det: Fe56 ▾	to Det: Fe56 ▾	Step:

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

WHAT(2) = - (100***Z** + 100 000***A** + 100 000 000 * **m**)

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

Warning on AUXSCORE

- To be used with care, or better NOT used at all for energy deposition
- **In real world, energy is eventually deposited by electrons only!**
- In Monte Carlo calculations, part of the energy is deposited "by other particles" as continuous energy deposition or point energy deposition, depending on 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!

Normalization of plots

- The Monte Carlo simulation does NOT know the intensity of your beam. It only knows how many event histories were run
- Normalization in all FLUKA intrinsic scoring is “per primary event”, or better, “per unit primary weight” if the source particles are biased
- The normalization to experimental conditions has to be done by the user
- Exception : Activation scoring, because there the beam intensity is provided by the user in the `IRRPROFI` card (see lecture on activation).
- For scoring which involves volumes (`USRTRACK`) or areas (`USRBDX`), quantities are normalized per unit volume (1/cm^3) or unit area (1/textsuperscript2) only for regular meshes (e.g. `USRBIN`) - for all other scorers, correct area or volume values have to be given by the user in the scoring card.
- Normalization can also be done at the plotting stage using the 

field in FLAIR’s `PLOT` menu.

Statistical uncertainties

- Can be calculated for single histories (not in FLUKA), or for batches (cycles or runs) of several histories
- Distribution of scoring contributions by single histories can be very asymmetric (many event histories contribute little or zero)
- Scoring distribution from batches tends to Gaussian for $N \to \infty$, provided $\sigma^2 \neq \infty$ (thanks to **Central Limit Theorem!**)
- The standard deviation of an estimator calculated from batches or from single histories is an estimate of the standard deviation of the actual distribution ("error of the mean")
- The goodness of such an estimate depends on the type of estimator and on the particular problem (but it converges to the true value for $N \to \infty$)

Statistical uncertainties

- The variance of the mean of an estimated quantity x (e.g. *fluence*), calculated in N batches, is:

$$\sigma^2_{<x>} = \frac{1}{N-1}\left[\frac{\sum_1^N n_i x_i^2}{n} - \left(\frac{\sum_1^N n_i x_i}{n}\right)^2\right]$$

$$\frac{\text{mean of squares - square of means}}{\text{N - 1}}$$

where:

n_i = number of histories in the N batches

$n = \sum n_i$ = total number of histories in the N batches

x_i = average of x in the i^{th} batch: $x_i = \sum_{j=1}^{n_i} \frac{x_{ij}}{n_i}$

x_{ij} is the contribution to x of the j^{th} history in the i^{th} batch

In the limit $N = n$, $n_i = 1$, the formula applies to single history statistics

The FLUKA user routines combining several scoring files will use **this** formula!

Statistical uncertainties

Practical tips:

- Use always at least 5-10 batches of comparable size (it is not at all mandatory that they be of equal size)
- Never forget that the variance itself is a stochastic variable subject to fluctuations
- Plot 2D and 3D distributions! Looking at them the eye is the best tool in judging the quality of the result
- Use FLAIR's option to plot errors in the PLOT menu
- As a rule of thumb, the statistical uncertainty of your estimators in the regions you are interested in should be better than 5% (use variance reduction methods, see lecture on biasing)

FLUKA post-processing

- Post-processing utilities are provided in the FLUKA distribution. They
  - Sum the results from different cycles
  - Calculate statistical uncertainties
  - Provide a summed file, that contains averages and statistics, and can be re-used to sum other cycles
  - Provide human-readable as well as gnuplot-readable output
- These utilities are used by flair in the "Data process"-button Process
- Can be (also) used directly from command line
- All of them are in the `$FLUPRO/flutil/`-directory
- Compiled automatically during installation of FLUKA

List of post-processing codes

- `flutil/detsuw.f` - DETECT: sum and provide tab_lis-file
- `flutil/gplevbin.f` - USRBIN: prepare 2D or 1D for plot
- `flutil/usbsuw.f` - USRBIN: sum and .bnn binary file
- `flutil/usbrea.f` - USRBIN: convert binary output to ASCII
- `flutil/usrsuwev.f` - RESNUC: offline evolution
- `flutil/usrsuw.f` - RESNUC: sum and tab_lis-files
- `flutil/ustsuw.f` - USRTRACK: sum and tab_lis-files
- `flutil/usxrea.f` - USRBDX: read summed file
- `flutil/usxsuw.f` - USRBDX: sum and provide tab_lis-files
- `flutil/usysuw.f` - USRYIELD: sum and tab_lis-files

www.fluka.org