Estimators and Scorings

FLUKA Beginner’s Course

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 l times the probability per unit distance Σ: $R = l\Sigma$

- The reaction rate will be $\dot{R} = \mathrm{d}l/\mathrm{d}t \ \Sigma = v\Sigma$, 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 $[cm^{-3}\ cm\ t^{-1}]=[cm^{-2}\ t^{-1}]$.

- 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 both Σ and Φ are 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:

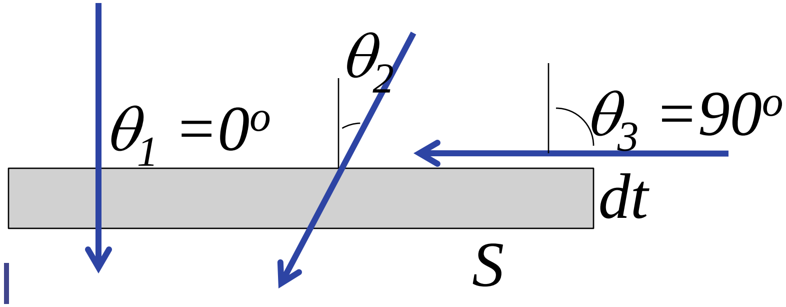
$$\dot{\Phi}(v)\,dt = n(v)\,v\,dt = \frac{dN(v)}{dV}\frac{dl(v)}{dt}dt = \lim_{\Delta V \to 0}\frac{\sum_i l_i(v)}{\Delta V}$$

- Collision density estimation:

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

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 count 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 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

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
- Scoring can be geometry dependent AND/OR geometry independent FLUKA can score particle fluences, current, track length, energy spectra, Z spectra, energy deposition...
- Either integrated over the “run”, with proper normalization, OR event-by event
- Standard scoring can be weighted by means of simple user routines

Related Scoring Commands

- USRTRACK, USRCOLL score average dΦ/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.

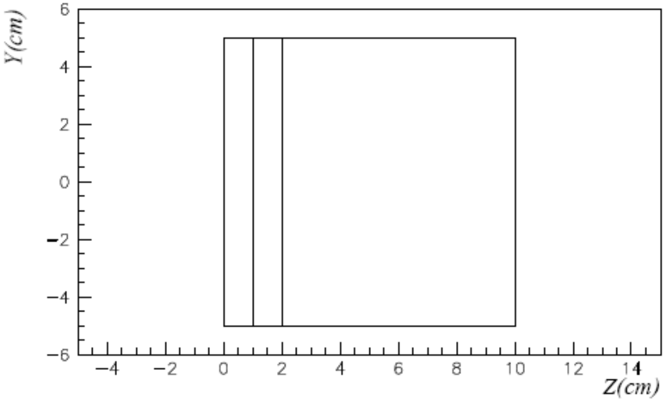
The FLUKA Output Files

The respective Fluka output consists of:

- A main (standard) output, written on logical output unit LUNOUT (predefined as 11 by default) [.out]
  - for details refer to the **lecture explaining the FLUKA output**
- 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)
- 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).

USRBIN

```
* Energy deposition
*   +    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 deposition (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.

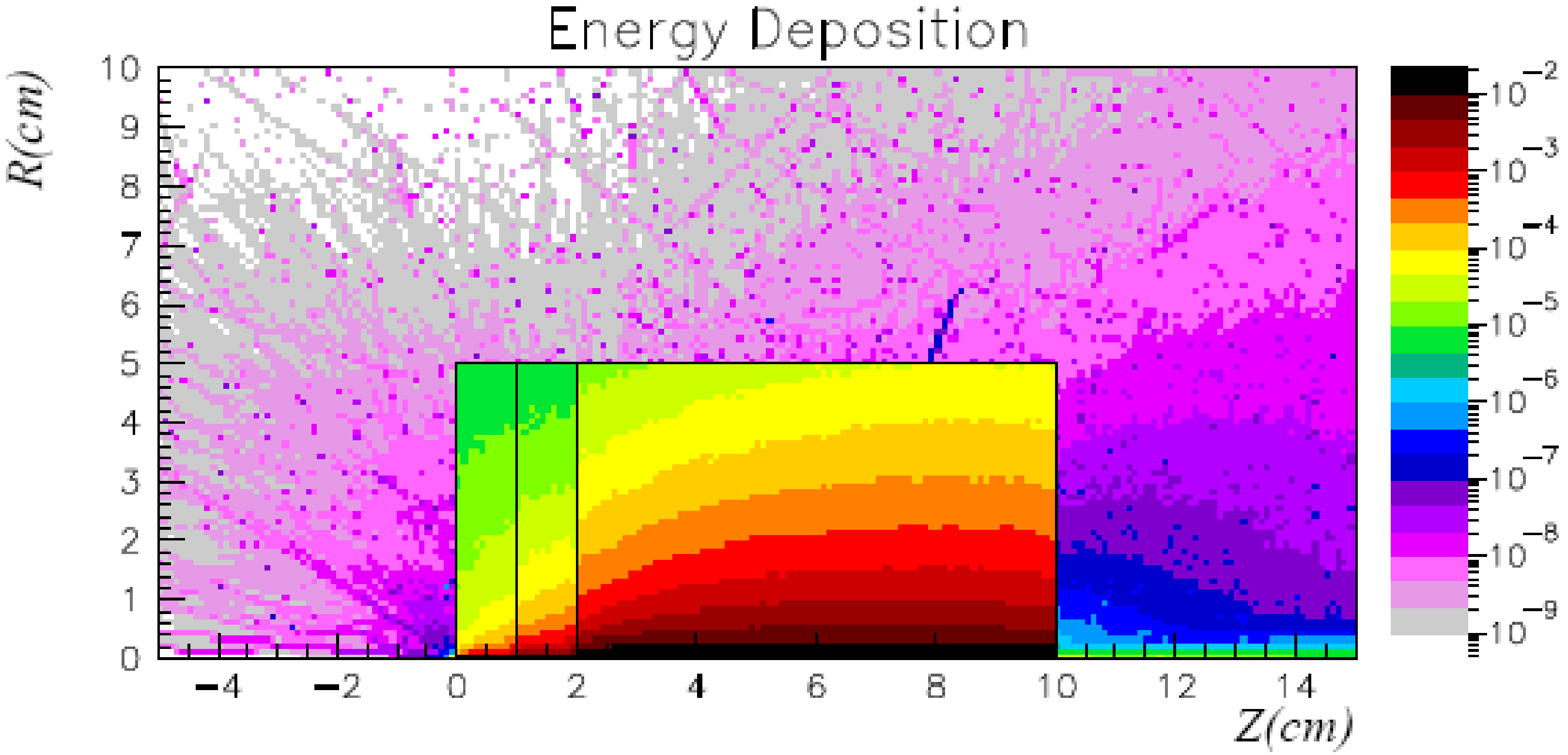
```
* Neutron fluence
*   +    1    +    2    +    3    +    4    +    5    +    6    +    7    +
*                R-Z  PartType Outp.Unit      Rmax     axisY      Zmax
USRBIN          11.0   NEUTRON     -40.0      10.0                15.0TargNeu
*               Rmin    axis X      Zmin    R-bins  Phi-bins    Z-bins
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.

USRBIN → The Result

WHAT(2) = ENERGY :Energy deposition from a 3.5 GeV proton beam hitting at [0.,0.,0.] directed along z

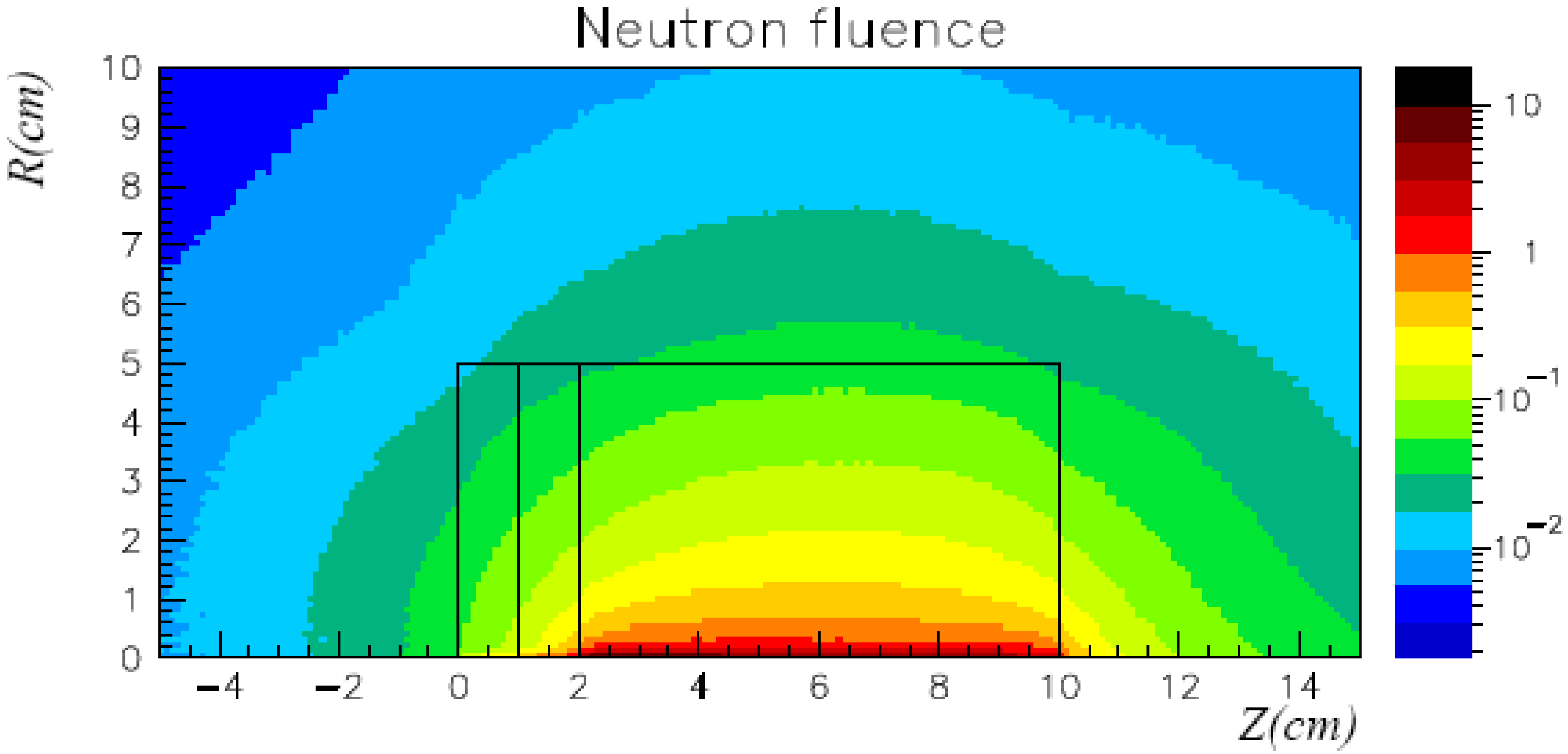
results are normalized to GeV/cm^3 per primary



USRBIN → The Result

Same, WHAT(2)= NEUTRON to get neutron fluence

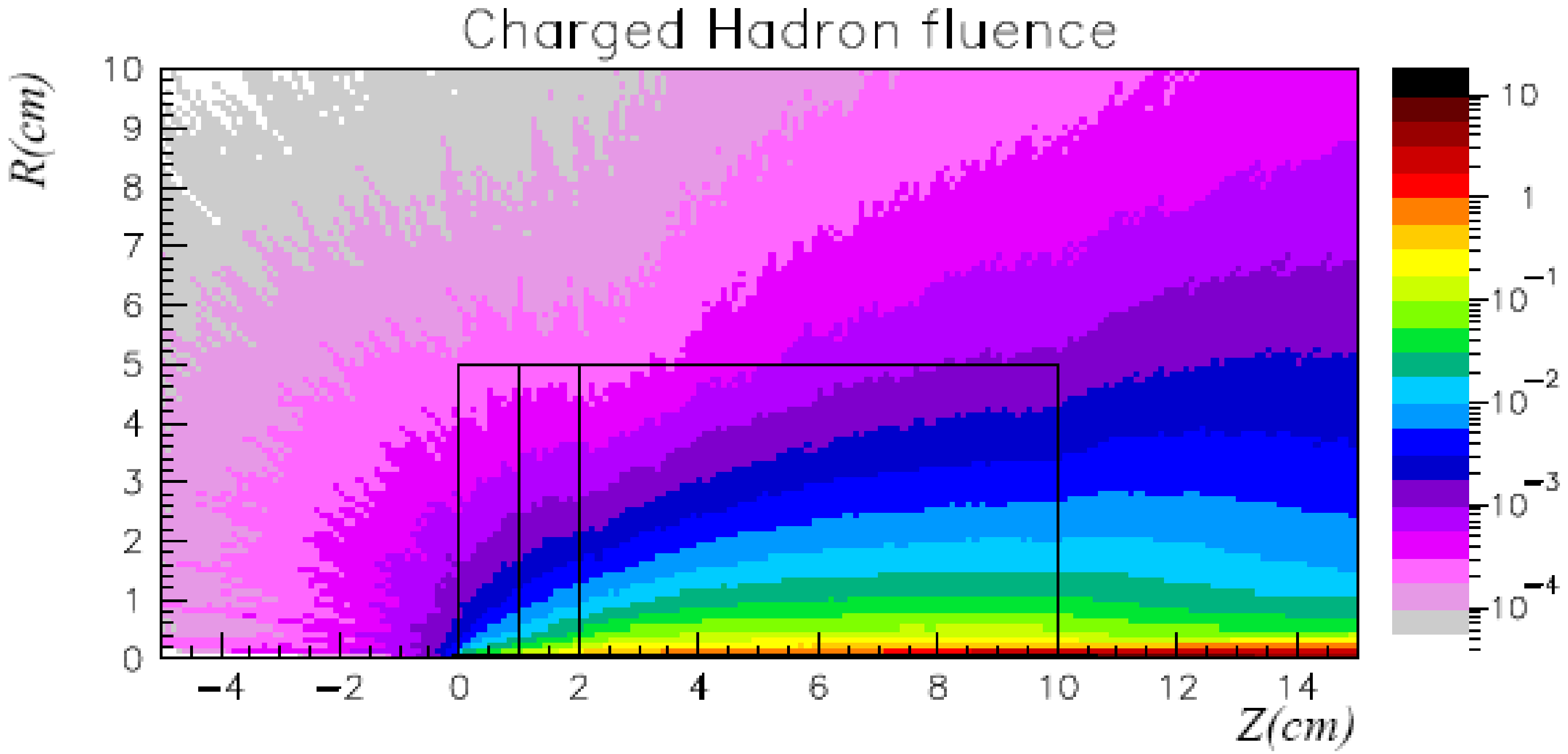
results are normalized to particles/cm^2 per primary



USRBIN → The Result

Same, WHAT(2)= HAD-CHAR to get charged hadron fluence

results are normalized to particles/cm^2 per primary



USRBIN – more quantities

USRBIN can score most of the "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
- ACTIVITY:activity per unit volume (see lecture on Activation)

- .. And more. Try to look for USRBIN on the manual;

USRBDX

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} 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.                                  &
```

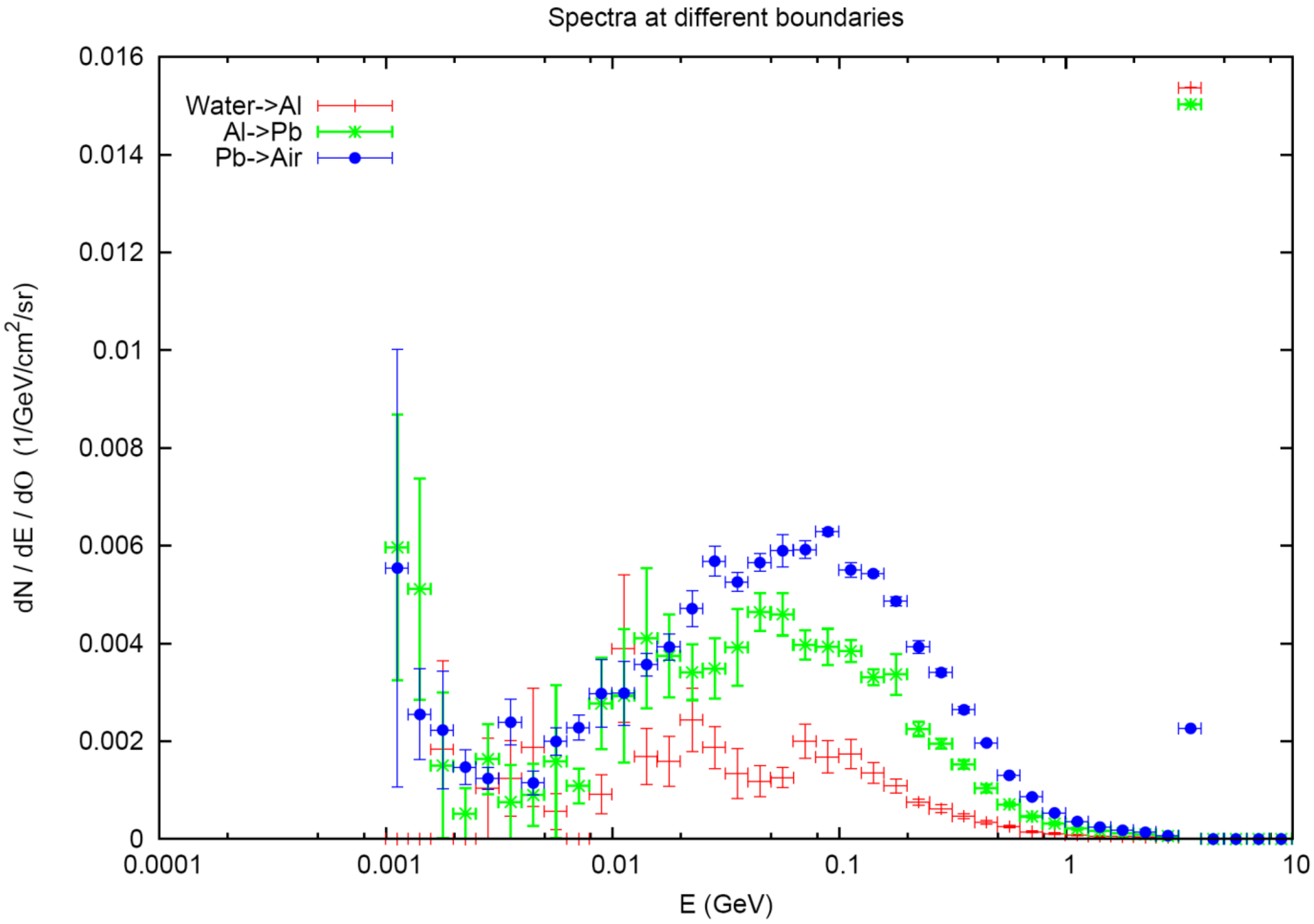
- Score at the surface between 2nd and 3rd target section, same as before but in 3 angular bins.

```
*    +    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 → The Result

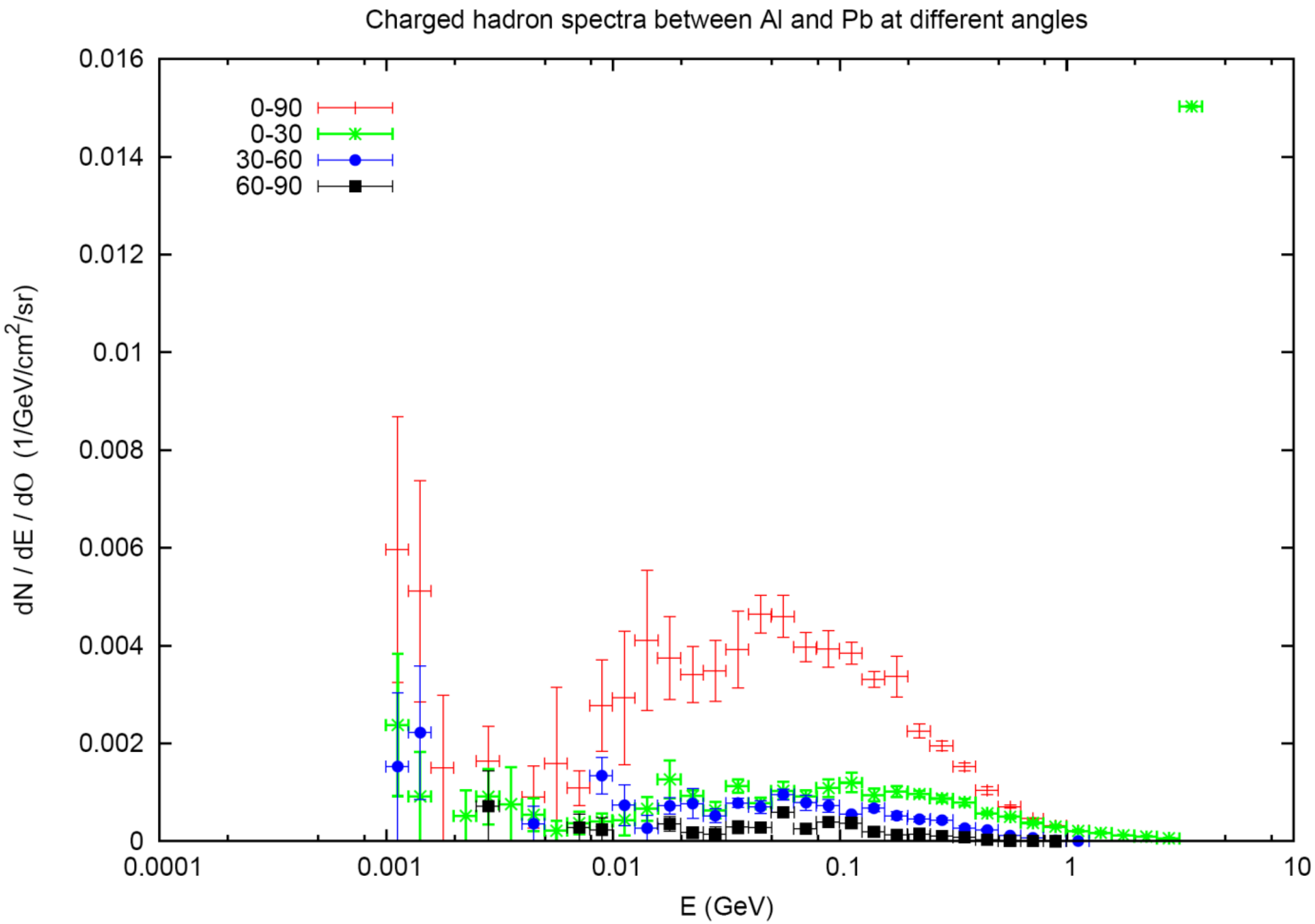
This is true only if the surface area is explicitly given

- Evolution of charged hadron spectra at the various surfaces results are normalized to /GeV/cm²/sr per primary



USRBDX → The Result

- Double differential charged hadron spectra for different angles; results are normalized to /GeV/cm^2/sr per primary



USRTRACK

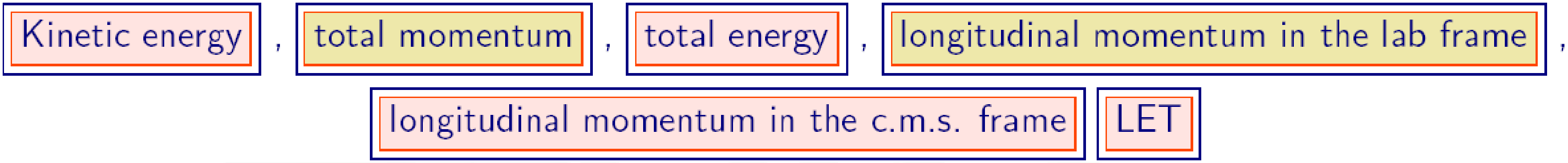
- Calculates fluence as a function of energy by scoring track-length in a given volume. Results are normalized to /GeV/cm^2/primary

```
*    +    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

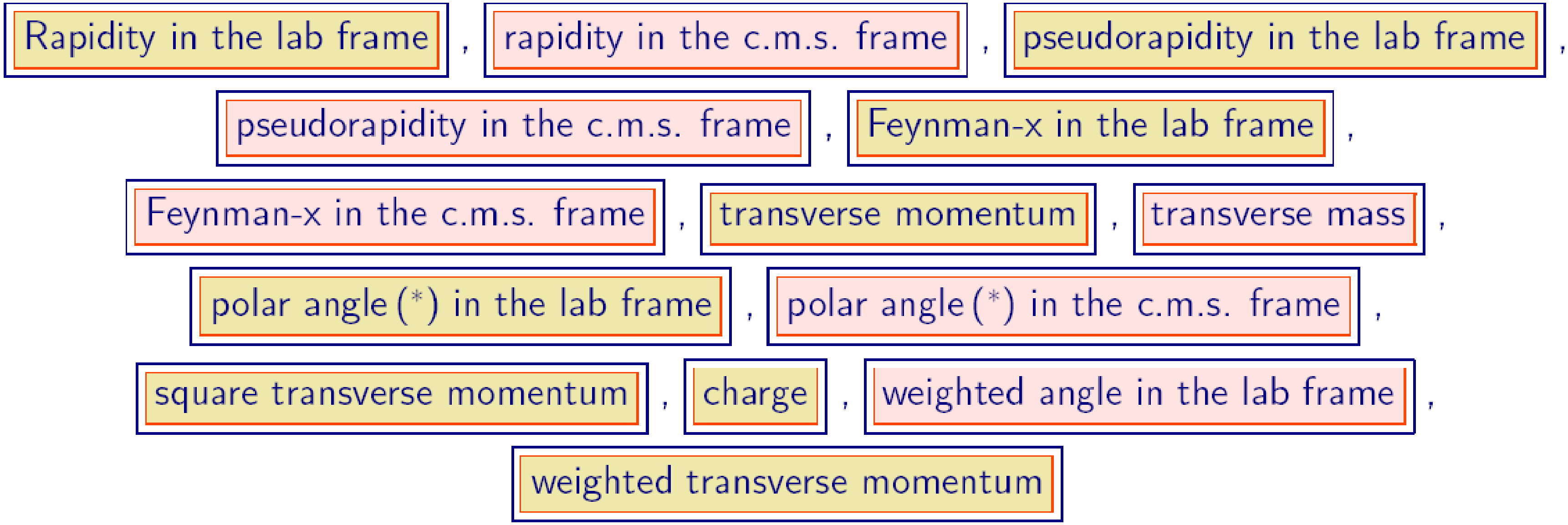
USRYIELD

- Scores a double-differential particle yield around an extended or a point target.
- "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 (in degrees or radians)



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

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 (or a different direction specified with SDUM=BEAMDEF).

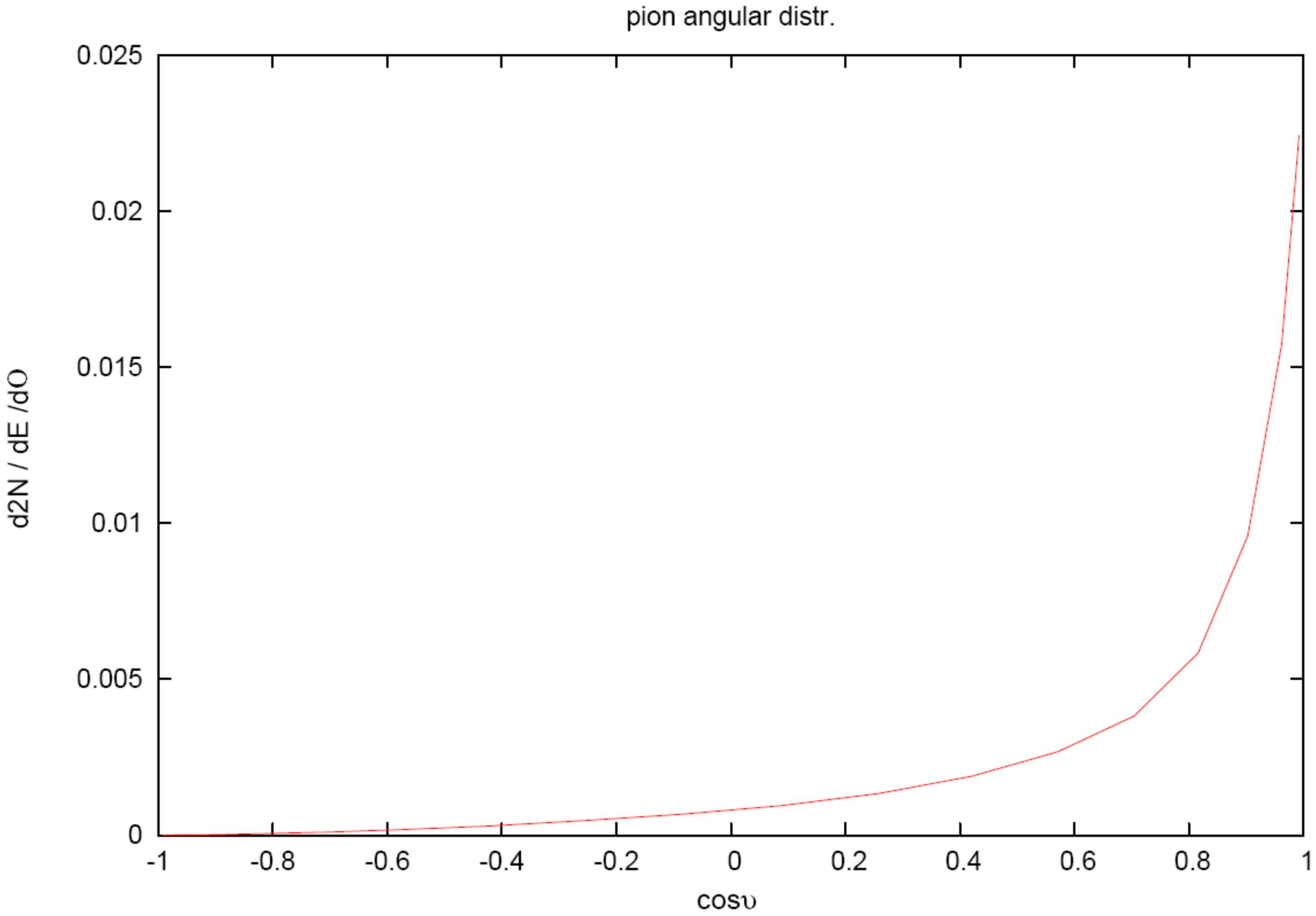
```
*    124 = 24 + 1 * 100 => polar angle (in degrees) and kinetic energy
*                              out.unit       Reg1      Reg2      Norm
*               Amax      Amin   #Abins       Emax      Emin dbl.differential
*   +    1    +    2    +    3    +    4    +    5    +    6    +    7    +
USRYIELD       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, particles/GeV/sr)

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

- To analyze the results of the different scoring options, several programs are available
- The most powerful ones are kept in $FLUPRO/flutil.
- They assume that the **estimator files are unformatted**, and can calculate standard deviations and integral 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 averages, standard deviations, **cumulative (integral)** quantities
  - an unformatted file which can replace the *N* unformatted estimator files and can be used for further calculations
  - a text file with extension *_tab.lis* to be easily readout by graphic codes

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

Standard Postprocessing Programs

- Example of sum.lis for 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

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

WARNING!! The Tot. response is NOT integrate over the second quantity !!
(in this case particles/GeV)

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
...
```

FILTERS : AUXSCORE

WARNING!!
ONLY energy deposition in the ionization by the selected particle

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

For instance: USRBIN energy deposition by muons only:

```
*   +    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&
AUXSCORE      USRBIN     MUONS             TargEne   TargEne
```

Assign the “muons” filter to the USRBIN estimator named TargEne

Another example: 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**:

$$\text{what}(2) = -\,(100*\mathbf{Z} + 100000*\mathbf{A} + \mathbf{m}*100000000)$$

with 0==all , i.e. 2600 == all Iron isotopes

Built-in Conversions and AUXSCORE

For some quantities, there is the possibility to get built-in conversions, without the need for user routines: done with generalized particles:

For example:

SI1MEVNE   Silicon 1 MeV-neutron equivalent fluence

DOSE-EQ   Dose Equivalent (pSv)

The set of conversion coefficients used to calculate DOSE-EQ can be selected by the user among a list (see manual) with AUXSCORE:

```
*    +    1    +    2    +    3    +    4    +    5    +    6    +    7    +
USRBIN          11.0   DOSE-EQ     -40.0      10.0                15.0TargDEQ
USRBIN           0.0                -5.0     100.0               200.0&
AUXSCORE      USRBIN                       TargDEQ   TargDEQ          AMB74
```

Scores equivalent dose by folding the particle fluences with the “AMB74” conversion coefficients

→ see lecture on activation