FLUKA Scoring

FLUKA Advanced Course

Contents

- *fluence*
- the Monte Carlo way
- FLUKA built-in scoring
- user routines [in the dedicated lecture]
- (statistical) errors

Phase space

- Phase space: a concept of classical Statistical Mechanics
- Each Phase Space dimension corresponds to a particle degree of freedom
- 3 dimensions correspond to Position in (real) space: x, y, z
- 3 dimensions correspond to Momentum: p_x, p_y, p_z
  (or Energy and direction: E, θ, φ)
- More dimensions may be envisaged, corresponding to other possible degrees of freedom, such as quantum numbers: spin, etc.
- Another degree of freedom is the particle type itself (electron, proton...)
- Each particle is represented by a point in phase space
- Time can also be considered as a coordinate, or it can be considered as an independent variable: the variation of the other phase space coordinates as a function of time constitutes a particle "history"

The angular flux Ψ

The angular flux Ψ is the most general radiometric quantity:

particle phase space density × velocity

or also

derivative of fluence Φ(x,y,z) with respect to 3 phase space coordinates: time, energy and direction vector

$$\Psi = \frac{\partial\Phi}{\partial t\,\partial E\,\partial\vec{\Omega}} = \dot{\Phi}_{E\vec{\Omega}}$$

Ψ is fully differential, but most Monte Carlo estimators integrate it over one or more (or all) phase space dimensions: coordinates, time, energy, angle

Fluence Φ, on the opposite, is the most integral radiometric quantity:

$$\Phi = \iiint_{E\vec{\Omega}t} \dot{\Phi}_{E\vec{\Omega}}\, dE\, d\vec{\Omega}\, dt = nl$$

where *n* = particle density in normal space, *l* = tracklength

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 $n(\mathbf{r},v)=dN/dV$ $[cm^{-3}]$ 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 σ [barn=10^{-24}cm^2].

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

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 interactions can be obtained from both, by multiplying them 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 (NOT IN VACUUM!):

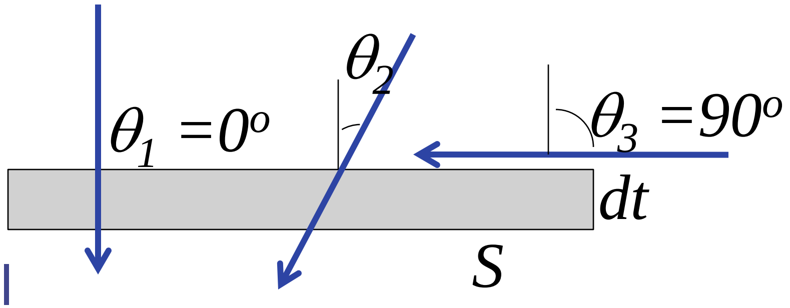
$$\dot{\Phi}(v) = \frac{d\dot{R}(v)}{dV}\lambda(v)$$

0 x ∞

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 $S\,dt$

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

- While the current *J* will be to count the number of particles crossing the surface divided by the surface

$$J = dN/dS$$

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

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

The Boltzmann Equation

- All particle transport calculations are (explicit or implicit) attempts to solve the Boltzmann Equation
- It is a balance equation in phase space: at any phase space point, the increment of angular flux Ψ in an infinitesimal phase space volume is equal to

sum of all "production terms"
minus
sum of all "destruction terms"

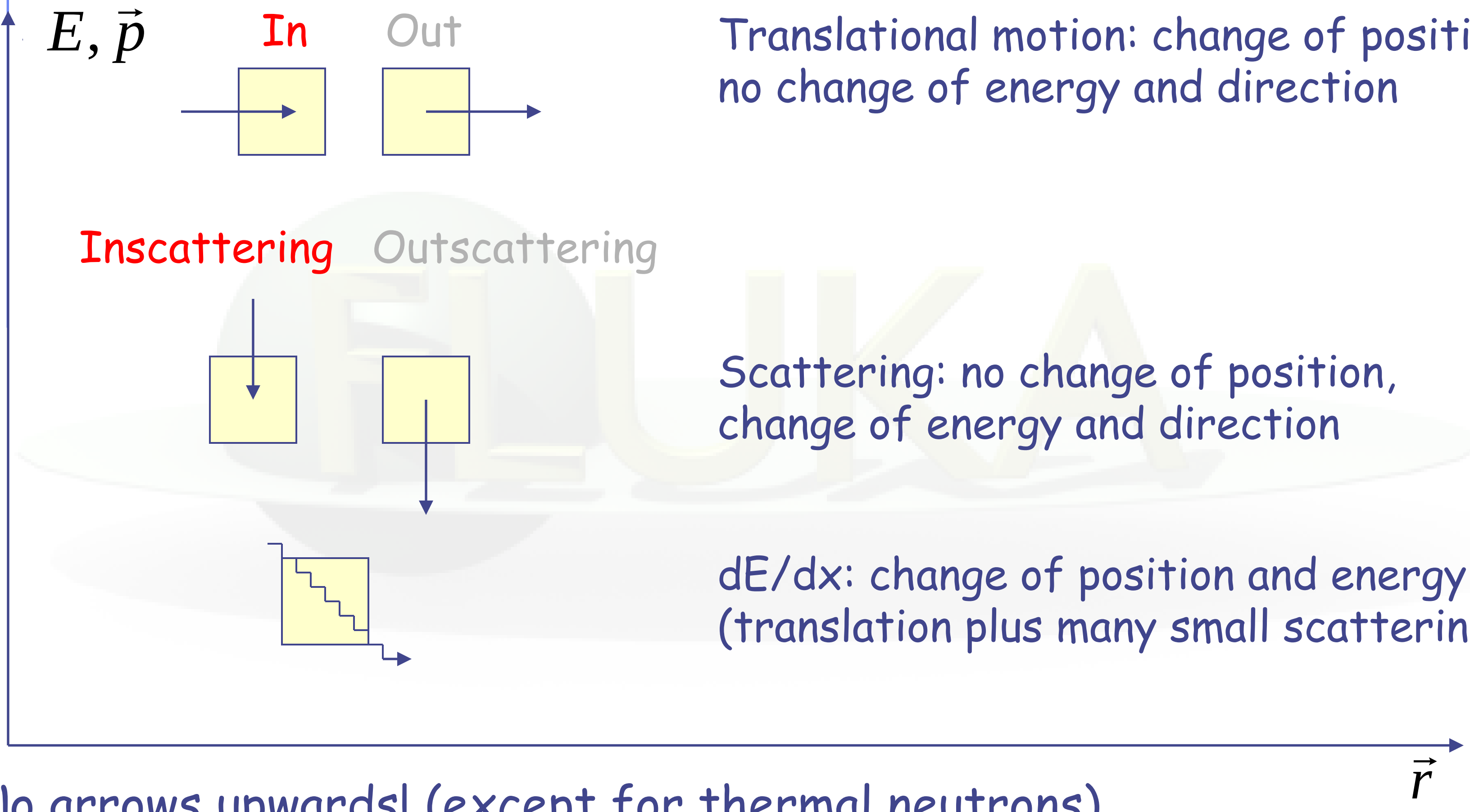
- Production:

  Sources, Translational motion "in", "Inscattering", Particle Production, Decay "in"
- Destruction:

  Absorption, Translational motion "out", "Outscattering", Decay "out"

A theorem of statistical mechanics, the Ergodic Theorem, says that the average of a function along the trajectories is equal to the average over all phase space.
The particle trajectories "fill" all the available phase space.

Visualizing a 2-D phase space...



Translational motion: change of position, no change of energy and direction

Scattering: no change of position, change of energy and direction

dE/dx: change of position and energy (translation plus many small scatterings)

No arrows upwards! (except for thermal neutrons)

The sources and the detectors

- To solve the Boltzmann Equation, we must define one or more source and one or more detectors
- A source is a region of phase space: one or more particle types, a range of space coordinates, a distribution in angle, energy and time (but often the source is simply a monoenergetic monodirectional point source — a "beam"!)
- Also a detector is a region of phase space, in which we want to find a solution of the Boltzmann equation
- We can look for solutions of different type:
  - □ at a number of (real or phase) space points
  - □ averages over (real or phase) space regions
  - □ projected on selected phase space hyperplanes
  - □ time-dependent or stationary
  - □........
- For each solution we must define a detector

Central Limit theorem

$$\lim_{N\to\infty} P(S_N) = \frac{1}{\sqrt{\frac{2\pi}{N}}\,\sigma_A}\, e^{-\frac{(S_N-\overline{A})^2}{2\sigma_A^2/N}}$$

For large values of N, the distribution of averages (normalized sums S_N) of N independent random variables identically distributed, according to any distribution with mean $\overline{A}$ and variance $\sigma_A^2 \neq \infty$, tends to a normal distribution with mean $\overline{A}$ and variance σ_A^2/N

$$\lim_{N\to\infty} S_N = \lim_{N\to\infty} \frac{\sum_1^N A(x, y, z,\ldots)\, f'(x)\; g'(y)\; h'(z) \ldots}{N} = \overline{A}$$

MC mathematical foundation

The Central Limit Theorem is the mathematical foundation of the Monte Carlo method. In words:

> Given any observable *A*, that can be expressed as the result of a convolution of random processes, the average value of *A* can be obtained by sampling many values of *A* according to the probability distributions of the random processes.

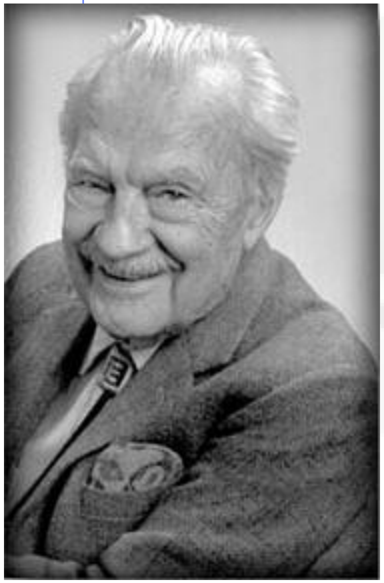
MC is indeed an integration method that allows to solve multi-dimensional integrals by sampling from a suitable stochastic distribution.

The accuracy of MC estimator depends on the number of samples:

$$\sigma \propto \frac{1}{\sqrt{N}}$$

The Monte Carlo method

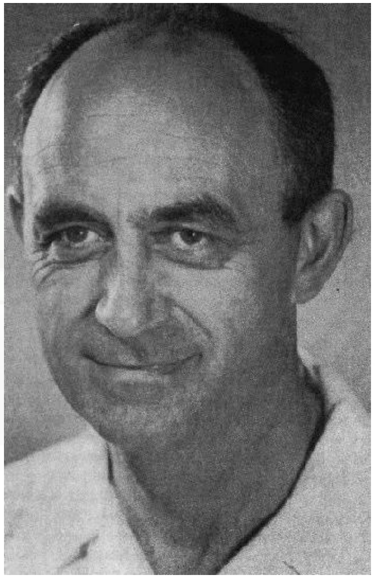
Invented by John von Neumann, Stanislaw Ulam and Nicholas Metropolis (who gave it its name), and independently by Enrico Fermi



N. Metropolis

S. Ulam

J. von Neumann



E. Fermi

Monte Carlo as Simulation

- It was soon realized, however, that when the method was applied to an equation describing a physical stochastic process, such as neutron diffusion, the model (in this case a random walk) could be identified with the process itself
- In these cases the method (analog Monte Carlo) has become known as a simulation technique, since every step of the model corresponds to an identical step in the simulated process

Particle transport

- Particle transport is a typical physical process described by probabilities (cross sections = interaction probabilities per unit distance)
- Therefore it lends itself naturally to be simulated by Monte Carlo
- Many applications, especially in high energy physics and medicine, are *based on simulations* where the history of each particle (trajectory, interactions) is reproduced in detail
- However in other types of application, typically shielding design, the user is interested only in the expectation values of some quantities (fluence and dose) at some space point or region, which are *calculated as solutions of a mathematical equation*
- This equation (the Boltzmann equation), describes the statistical distribution of particles in phase space and therefore does indeed represent a physical stochastic process
- But in order to estimate the desired expectation values it is not necessary that the Monte Carlo process be identical to it

Integration without simulation

- In many cases, it is more efficient to replace the actual process by a different one resulting in the same average values but built by sampling from modified distributions
- Such a *biased process*, if based on mathematically correct variance reduction techniques, converges to the same expectation values as the unbiased one
- But it cannot provide information about the higher moments of statistical distributions (fluctuations and correlations)
- In addition, the faster convergence in some user-privileged regions of phase space is compensated by a slower convergence elsewhere

Analog Monte Carlo

In an analog Monte Carlo calculation, not only the mean of the contributions converges to the mean of the actual distribution, but also the variance and all moments of higher order:

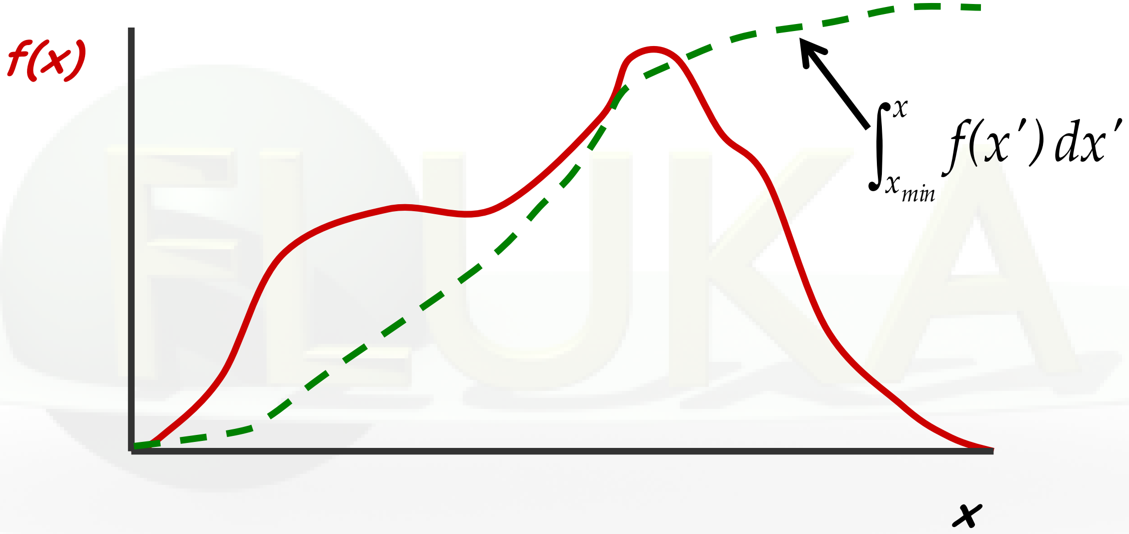
$$\lim_{N\to\infty}\left[\frac{\sum_1^N (x-\bar{x})^n}{N}\right]^{\frac{1}{n}} = \sigma_n$$

Then, partial distributions, fluctuations and correlations are all faithfully reproduced: in this case (and in this case only!) we have a real simulation

Random sampling: the key to MC

The central problem of the Monte Carlo method:

Given a Probability Density Function (pdf), ***f(x)***, generate a sample of ***x***'s distributed according to ***f(x)*** (***x*** can be multidimensional)



The use of random sampling techniques is the distinctive feature of Monte Carlo

Solving the integral Boltzmann transport equation by Monte Carlo consists of:

- Geometry and material description of the problem
- Random sampling from probability distributions of the outcome of physical events

Assumptions

- Static, homogeneous, isotropic, amorphous media and geometry

  Problems: e.g. moving targets*, atmosphere [must be represented by discrete layers of uniform density], radioactive decay taking place in a geometry different from that in which the radionuclides were produced*, crystal channeling*.

  * These restrictions have been (* are being) overcome in FLUKA

- Markovian process: the fate of a particle depends only on its actual present properties, not on previous events or histories

- Particles do not interact with each other

  Problems: e.g. the Chudakov effect (charges cancelling in e^+e^- pairs)

- Particles interact with individual electrons / atoms / nuclei / molecules

  Problems: invalid at low energies (X-ray mirrors)

- Material properties are not affected by particle reactions

  Problems: e.g. burnup

Built-in 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 (some of which mentioned afterwards), 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
- Geometry dependent and geometry independent scoring both available
- FLUKA can score particle fluence (tracklength), current, energy spectra, angular distributions, energy deposition, activity ...
- Either integrated over the "run", with proper normalization, OR event-by event
- Standard scoring can be weighted by means of simple user routines

Scoring cards [1]

see the Beginners' Course

- SCORE scores energy deposited (or star density) in each region *[table in the .out file, not automatically merged over cycles]*
- RESNUCLEi scores residual nuclei (or their activity) in a given region
- USRTRACK (USRCOLL) scores average differential fluence $d\Phi/dE$ of a given type or family of particles over a given region
- USRBDX scores average double differential fluence (or current) $d^2\Phi/dEd\Omega$ of a given type or family of particles over a given surface
- USRBIN scores the spatial distribution of deposited energy density, dose, integrated fluence, star density, *dose equivalent*, net charge, specific activity, ... in a regular mesh (cylindrical or Cartesian) described by the user
- USRYIELD scores a double differential yield *[do not ask for cross section, like by default]* of particles escaping from a surface. The distribution can be with respect to energy and angle (wrt the beam direction), but also many other more "exotic" quantities

Remember that *low energy (<20MeV) neutrons*
have a pre-defined energy binning

Warnings [I]

- USRBIN scoring algorithm:

  By selecting **WHAT(1)>=10**, *energy deposition, dose,* ... are distributed along the particle track (recommended!)

*** Activity/fission/neutron balance binnings cannot be track-length!!!

  Point-wise quantities have to be scored at a point (select **WHAT(1)<10**)

- Badly defined USRBIN limits

```
******* Fluka stopped in Usrbin: "usr/eventbin" n.    1 *******
 ******* with zero width   0.000     for axis  R  ******
```

- Never use unit numbers smaller than 20 or higher than 99

  <20 reserved by FLUKA   >99 FORTRAN limitation

- Never mix the output of different scoring cards in the same unit

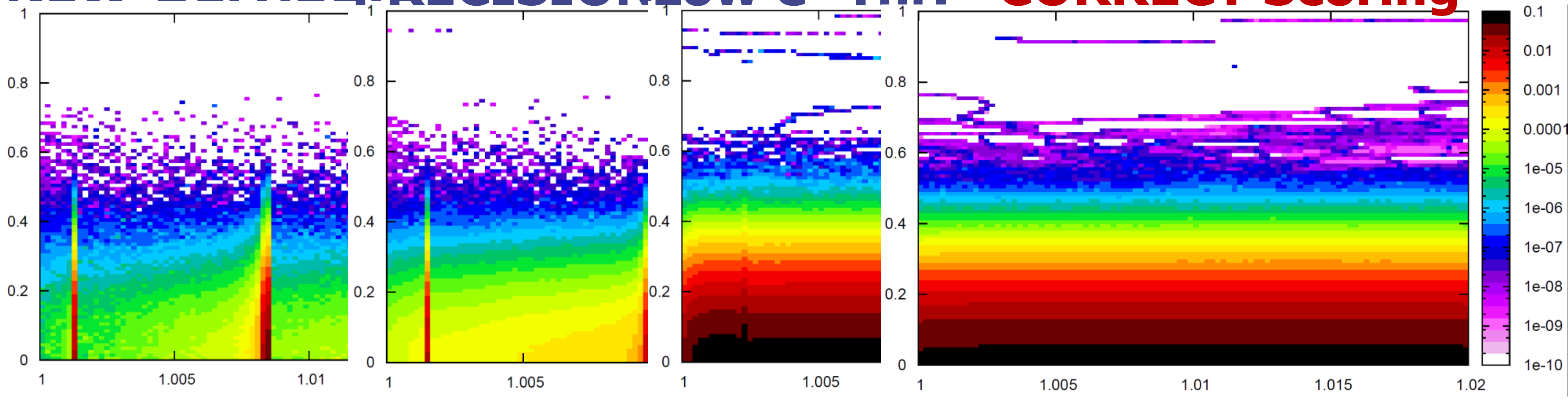
- Verify that you didn’t merge cycles referring to different input versions (change the name of the input file for every new problem!)

Example

- Thin window with low-E (5MeV) electron beam
- Energy deposition profile in the window (for radiation damage studies)
- Observation of 'strange peaks'

- **Trying to understand:** lower e^--thresholds help
- **Real-Problem:** point-wise scoring requested

NEW-DEFAULTS **PRECISION** **Low e- Thr.** **CORRECT Scoring**



Scoring cards [2]

- 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). Quite useful in case of LATTICE
- USERDUMP defines the events to be written onto a “collision tape” file Coupled to the **mgdraw** user routine
- AUXSCORE defines filters and conversion coefficients
- TCQUENCH sets scoring time cut-offs and/or Birks quenching parameters for binnings (USRBIN or EVENTBIN) indicated by the user
- DETECT scores energy deposition in coincidence or anti-coincidence with a trigger, separately for each “event" (primary history). Dedicated post-processing routine available

Dose-Equivalent (not Dose)

For some quantities, there is the possibility to get built-in conversions, without the need for user routines, rather through dedicated generalized particles. The most commonly used is *dose equivalent* (ambient dose equivalent or effective dose):

DOSE-EQ	Dose Equivalent [pSv]
DOSEQLET	Dose Equivalent via Q(LET) – unrestricted LET in water – according to ICRP60 [GeV/g]

!!!! Different to !!!

(DOSE	total absorbed dose in GeV/g
DOSE-EM	as above but electromagnetic contribution only)

DOSE-EQ is calculated by folding *particle fluences* with conversion coefficient sets, selected by the user among a list (see manual) through AUXSCORE. The default set (not requiring the AUXSCORE association) is “AMB74”.

WARNING : in case of DOSE-EQ no coefficients available for heavy ions (ok for DOSEQLET) !!!

“FILTER” : AUXSCORE

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

```
USRBIN     11.0     ENERGY    -40.0      10.0           15.0 TargEne
USRBIN      0.0                -5.0     100.0          200.0 &
AUXSCORE USRBIN      MUONS              TargEne
```

ionization (+NIEL) by the selected particle, critically depending on the delta rays threshold! [doubtful physical meaning]

Another example: score the yield [vs polar angle and kinetic energy] of 56-Iron ions (there is no separate name for each ion specie, except light ones. HEAVYION scores all isotopes heavier than alpha together!)

```
USRYIELD     124.0 HEAVYION   -87.    TARGS3     INAIR      1.0Fe56
USRYIELD     180.0      0.0    18.     10.0       0.0      3.0&
AUXSCORE USRYIELD -5602600.             Fe56
```

The requested ion is coded in WHAT(2)= - (100***Z** + 100000***A** + **m***100000000)
according to its **A**, **Z** and (optionally) isomeric state **m**
with 0==all, i.e. -2600 == all Iron isotopes

Routines associated to FLUKA scoring

comscw.f	weighting energy deposition and star production
fluscw.f	weighting fluence, current and yield
mgdraw.f	*general event interface*
usrrnc.f	intercepting produced residual nuclei (at the end of their path)
endscp.f	shifting energy deposition
fldscp.f	shifting fluence
musrbr.f	special USRBIN binning (lattice): returns region #
lusrbl.f	special USRBIN binning (lattice): returns lattice #
fusrbv.f	special USRBIN binning (lattice): returns zero
mdstck.f	intercepting particle stack
stuprf.f	
stupre.f	

Statistical Errors [1]

- Can be calculated for single histories (not in FLUKA), or for batches of several histories
- Distribution of scoring contributions by single histories can be very asymmetric (many histories contribute little or zero)
- Scoring distribution from batches tends to Gaussian for $N \rightarrow \infty$, 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”)
- How good is such an estimate depends on the type of estimator and on the particular problem (but it converges to the true value for $N \rightarrow \infty$)

Statistical Errors [2]

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

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

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

where:

n_i = number of histories in the i^{th} batch

$n = \Sigma n_i$ = 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

Statistical Errors [3]

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
- Be careful about the way convergence is achieved: often (particularly with biasing) apparent good statistics with few isolated spikes could point to a lack of sampling of the most relevant phase-space part
- Plot 2D and 3D distributions! Looking at them the eye is the best tool in judging the quality of the result

Statistical Errors [4]

from an old version of the MCNP Manual:

Relative error	Quality of Tally
50 to 100%	Garbage
20 to 50%	Factor of a few
10 to 20	Questionable
< 10%	Generally reliable

- Why does a 30% σ mean an uncertainty of a "factor of a few"?
  Because σ in fact corresponds to the sum (in quadrature) of two uncertainties: one associated to the fraction of histories which don't give a zero contribution and the other reflecting the spread of the non-zero contributions
- The MCNP guideline is empirically based on experience, not on a mathematical proof. But it has been generally confirmed as working also with other codes
- Small penetrations and cracks are very difficult to handle by MC, because the "detector" is too small and too few non-zero contributions can be sampled, even by biasing

Systematic Errors

- physics: different codes are based on different physics models. Some models are better than others. Some models are better in a certain energy range. Model quality is best shown by benchmarks at the microscopic level (e.g. thin targets)
- artifacts: due to imperfect algorithms, e.g., energy deposited in the middle of a step*, inaccurate path length correction for multiple scattering*, missing correction for cross section and *dE/dx* change over a step*, etc. Algorithm quality is best shown by benchmarks at the macroscopic level (thick targets, complex geometries)
- data uncertainty: results can never be better than allowed by available experimental data!
- material composition: not always well known. In particular *concrete/soil* composition (how much water content? Can be critical). *Air* contains humidity and pollutants, has a density variable with pressure
- beam losses: most of the time these can only be guessed
- presence of additional material, not well defined (cables, supports...)
- geometries cannot be reproduced exactly (or would require too much effort)

Is it worth doing a very detailed simulation when some parameters are unknown or badly known?

Mistakes

- mis-typing the input: Flair is good at checking, but the final responsibility is the user’s
- error in user code: use the built-in features as much as possible!
- wrong units
- wrong normalization: quite common
- unfair biasing: energy/space cuts cannot be avoided, but must be done with much care
- forgetting to check that gamma production is available in the low energy neutron library (e.g., Ba cross sections)
- …

Biasing Mean Free Paths

Multiplicity Tuning BIASING

- Multiplicity tuning is meant to be to hadrons what LPB is for electrons and photons.
- A hadronic nuclear interaction at LHC energies can end in hundreds of secondaries. Except for the leading particle, many secondaries are of the same type and have similar energies and other characteristics
- The user can tune the average multiplicity in different regions

Interaction Length LAM-BIAS

- Mean life / average decay length of unstable particles can be artificially shortened
- Can increase generation rate of decay products without discarding the parent
- For hadrons the mean free path for nuclear inelastic interactions can be artificially decreased. Useful for very thin targets, and also for photonuclear reactions where the cross section is relatively small