FLUKA

Introduction to the Monte Carlo simulation of radiation transport

- The transport equation
- The Monte Carlo method
- Statistical uncertainties
- Advantages and limitations of the method

The radiation transport problem

Photons,
Leptons ($e^{\pm}$, $\mu^{\pm}$, $\tau^{\pm}$, ν),
Hadrons (n, p, π, Σ,…),
Ions (Z,A),
Radioactive sources
Cosmic rays,

Colliding particle beams,
Synchrotron radiation,
…

“Monoenergetic”/Spectral

Energies:
- keV-PeV,
- down to thermal energies for neutrons.

Arbitrary geometry,

Various bodies,
materials, compounds.

Radiation-matter interaction,

Secondary particles,
Particle shower,

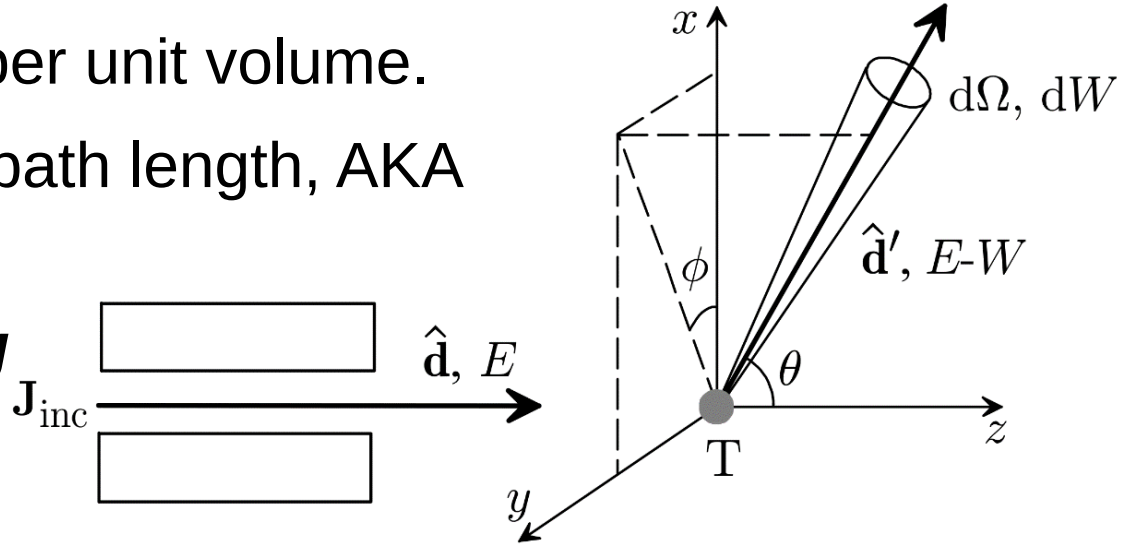
Material activation,
Magnetic and electric fields…

Measure/estimate/score:

- Energy-angle particle spectra,
- Deposited energy,
- Material damage,
- Biological effects,
- Radioactive inventories…

Terminology

- ***Radiation field***: an ensemble of particles, possibly of different species (**γ**,e±,p,n,…), each at a position **r** moving with energy E along a direction Ω=(θ,ϕ) with polar and azimuthal angles θ and ϕ.
- Every particle species can undergo a series of interaction mechanisms, each characterised by a ***differential cross section:*** $$\frac{\mathrm{d}^2\sigma}{\mathrm{d}\Omega\,\mathrm{d}W} \equiv \frac{\dot{N}_{\mathrm{count}}}{|\mathbf{J}_{\mathrm{inc}}|\,\mathrm{d}\Omega\,\mathrm{d}W}$$
- The ***integrated cross section*** σ (area) measures the likelihood of the interaction.
- Consider a medium with *N* scattering centers per unit volume.
- *N*σ gives the probability of interaction per unit path length, AKA ***macroscopic cross section***.
- 1/(Nσ) gives the ***mean free path*** or ***scattering length*** between consecutive interactions.



The transport equation

- Let $n_0(\mathbf{r},E,\Omega,t=0)$ be the particle density at the radiation source with energy E, moving in a direction Ω.
- The **transport equation** determines the radiation field (consisting of several particle species *i*, with different energies E, and different directions Ω) at another position **r** at a later time *t* by looking at the particle balance in a small volume V (with surface S)

$$\int_V \mathrm{d}\boldsymbol{r}\,\frac{\partial n_i(\boldsymbol{r},E,\Omega,t)}{\partial t} = -\oint_S \mathrm{d}A\,\mathbf{j}(\boldsymbol{r},E,\Omega,t)\cdot\hat{\boldsymbol{a}}$$ (unscattered particles)

$$-N\int_V \mathrm{d}\boldsymbol{r}\,n_i(\boldsymbol{r},E,\Omega,t)v(E)\sigma(E)$$ (particles scattered out)

$$+N\int_V \mathrm{d}\boldsymbol{r}\int \mathrm{d}E'\int \mathrm{d}\Omega'\;n_i(\boldsymbol{r},E',\Omega',t)v(E')\frac{\mathrm{d}\sigma}{\mathrm{d}\Omega''\mathrm{d}W''}$$ (particles scattered in)

$$+N\int_V \mathrm{d}\boldsymbol{r}\int \mathrm{d}E'\int \mathrm{d}\Omega'\;\sum_j n_j(\boldsymbol{r},E',\Omega',t)v(E')\frac{\mathrm{d}\sigma_{\mathrm{sec},i}}{\mathrm{d}\Omega''\mathrm{d}W''}$$ (production of secondaries)

$$+\int_V \mathrm{d}\boldsymbol{r}\,Q_{\mathrm{source}}(\boldsymbol{r},E,\Omega,t)$$ (source)

Notation: Ω'' is a direction such that scattering angles Ω' bring it to Ω.

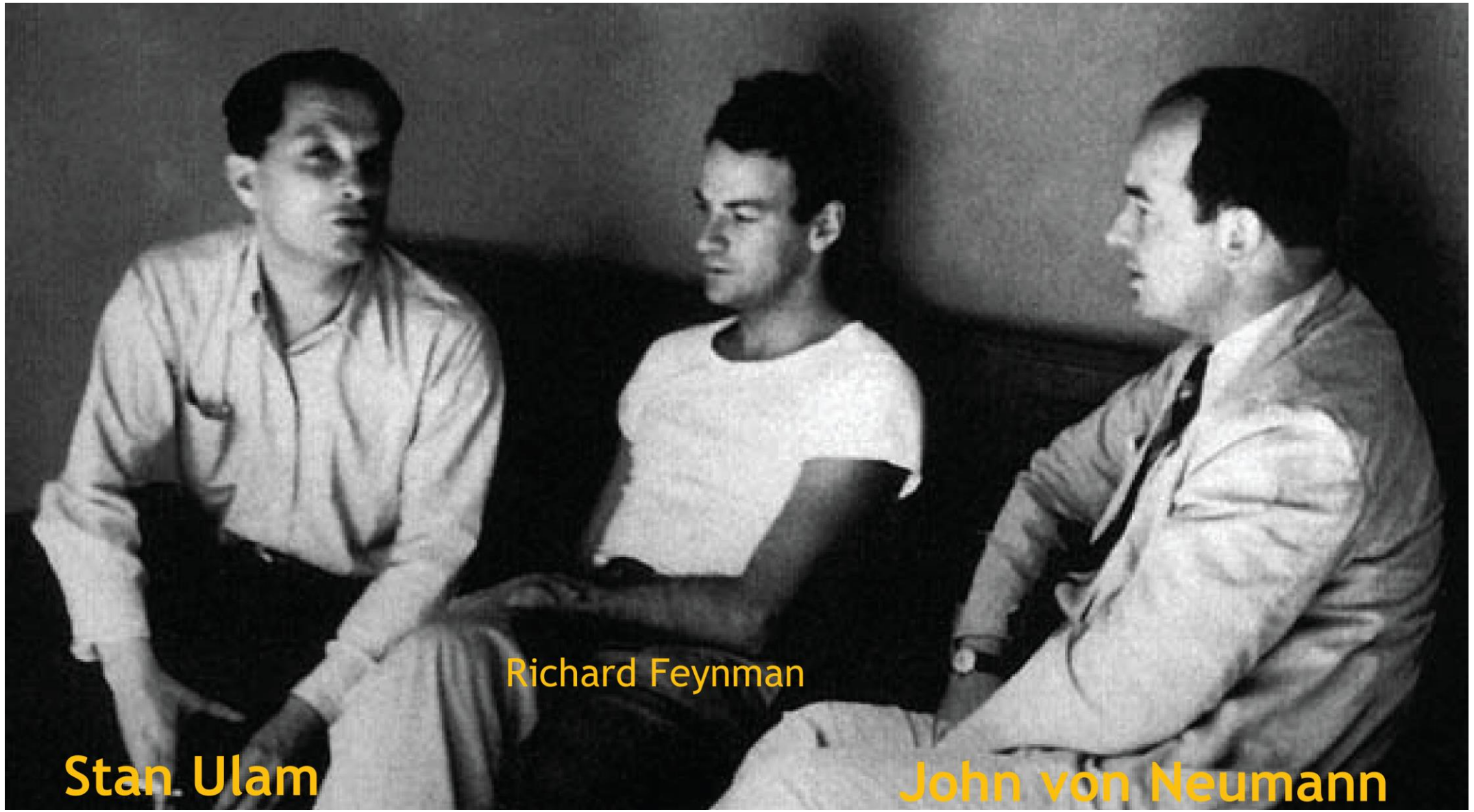
Solution strategies

- Transport equation to be solved for an arbitrary source density $n_0(\mathbf{r},E,\Omega,t)$, an arbitrary geometry, and realistic interaction cross sections.

- Solution strategies:
  - Analytical: only for restricted geometries and restricted interaction models.
  - Spectral: exploit symmetries and expand in appropriate basis functions. Only for restricted cases.
  - Numerical quadrature integration: general, but inefficient for high-dimensional integrals.
  - **Monte Carlo method:** general, efficient, can treat arbitrary radiation fields and geometries.

- Monte Carlo is a stochastic method, exploiting random numbers to:
  - **Simulate** an ensemble of particle histories governed by known interaction cross sections.
  - Track them in **arbitrary geometries**.
  - Accumulate contribution of each track to **statistical estimator** of the desired physical observables.

The origins



Probability and statistics toolkit for Monte Carlo simulations of radiation transport

Random variables

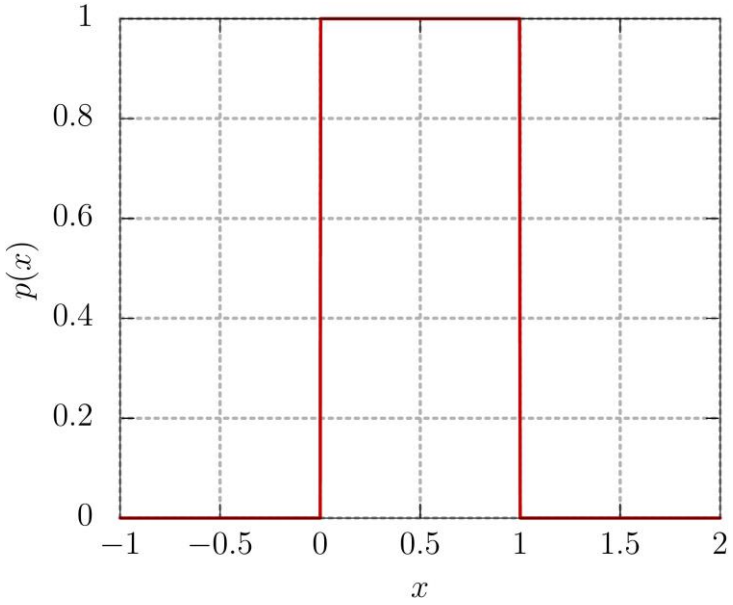
- A **random variable** X describes the outcome of a process whose value we cannot predict with certainty, but nevertheless we know:
  - Its possible values.
  - How likely each value is, governed by the **probability density function (PDF), p(x)**.
- Properties of p(x):
  - Positive defined: p(x)>=0 for all x
  - Unit-normalized: $\int dx\, p(x) = 1$
  - Integral gives probability: $\int_a^b dx\, p(x) = P(a<x<b)$
- The **expectation value** $\langle X \rangle = \int_{-\infty}^{\infty} \mathrm{d}x\; xp(x)$ measures the average value of X.
- The **variance** σ^2 measures the square deviation from <X>.

$$\sigma^2 = \langle (X - \langle X \rangle)^2 \rangle = \ldots = \langle X \rangle^2 - \langle X^2 \rangle$$

- The **standard deviation** σ measures the average deviation from <X>.

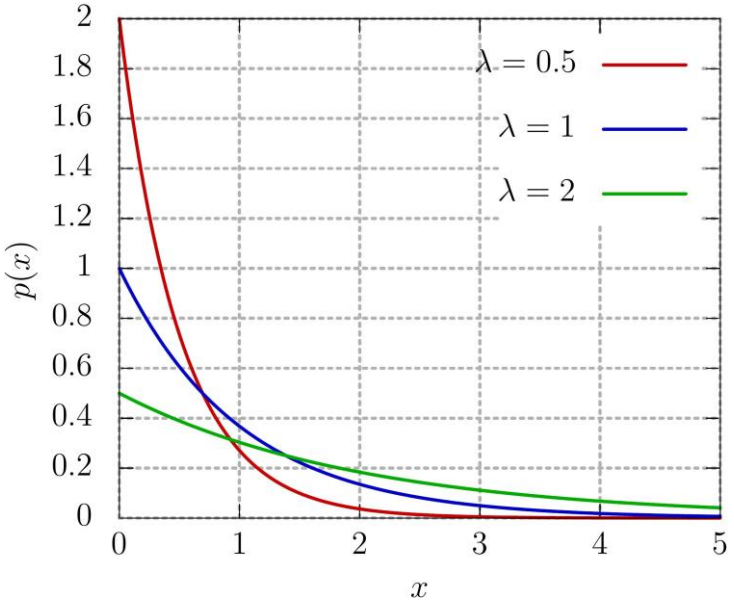
Relevant examples

Uniform distribution



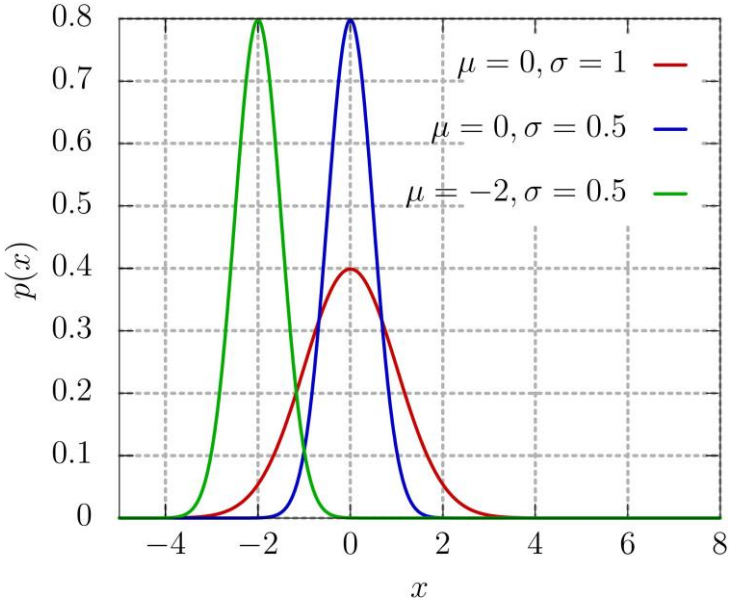
Basic PDF for sampling

Exponential



PDF from which particle steps are sampled (derivation in additional slides)

Gaussian



Appears in Central Limit Theorem below

Generation of homogeneously distributed random numbers

- For reasons of reproducibility, we use **pseudo-random numbers:** *uniformly* distributed numbers between 0 and 1 obtained from a deterministic algorithm (*not random*!) which pass all tests of randomness.

- Needs one/several **seed values**, X_1, from which the sequence starts: $X_2,X_3,X_4,\ldots$
- Different seed values yield different random number sequences.

- E.g.: linear congruence $X_{n+1} = \mathrm{mod}(aX_n+c,m)$, with carefully chosen a,c,m
- The random number generator used in FLUKA is **RM64**, based on an algorithm by G. Marsaglia et al. *Stat. Probabil. Lett*. **66** 183-187 (2004) and **8** 35-39 (1990).
- Based on a lagged Fibonacci generator: $X_{n+1}=\mathrm{mod}(X_{n-p}@X_{n-q},m)$, where @ is +,-,…, p=97, q=33.
- The state of the random-number generator requires **97 values**.

The state of the random-number generator in FLUKA

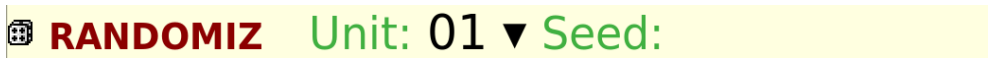
- At the end of each FLUKA cycle you will see files called `ran*`

```
 DF75DC       0   181CD    3039B6698493
     45       5B24BEE7E3FE70591
E14FD41E3FD9F96219018D003F631A592282EC723FDABC5431E61BA23FD52AA5
7616746F3FE9D8DDE961FB573FE0AF4A76A64FDA3FDA9A3CF23CECB23FD674B4
FD5594003F55F406927DD5003F857D81A39916283FBFC742F7E490E43FEBB1E6
259F67B83FBB498EE57697063FDB4EF07B77284E3FE947FB2A1395983FBC478B
5A700E083FC2BC89BC1415643FCF651D44DE10683FE774B4B05EC8203FD6EB8C
506530043FD2F20440C197943FDE6E2DF61740483FED57CA17F0CAF23FE071D8
F7A086823FD2179A6266CD843FCC5BC0D122239C3FDD1D0CB2B9D8E53FE0CCB8
CD0D0ABC3FCB35F570DCF1C43FD85E2BF75878B73FE7E37375FC2C033FE363F6
978EA48C3FDECE0E4CC18DFC3FDFBE33EF6390033FE828BE159746663FE03204
69F8C8603FE57BA36241A1443FECD93B6B0465603F90928EED43E1083FDB7A81
87BAFC6F3FEFCD9852BE02233FEE1427AA841B4C3FEE7ED6DE8F38F13FE07474
37BFFF503FB360BDE437AA703FB714FB27612303FD353CC39B8C9DF3FE01A01
CAD649303FD5067075EA51A23FE9C060C0BE59DB3FE2EA4D6EC986B03FC02303
D36EAA263FE7A2ECCE670FE73FE76D264C7667AC3FE27DE882628D193FED3B4B
27DA6F083FEBC43C28E2F8063FED9EF7287387163FD13F78E68DFCA63FE027E8
B9626D253FEC7BC3 73A7D443FE8595C6C9826173FEFEDED770937783FBAB23C
144DCB3E3FD4F440A34B92183FC7B43DB68C50D43FD0DE1510A26DC43FCC4C4A
E04497293FE3CCFC3D4D61D23FD691F6C50DE8BC3FD5FF61642614EC3FD2EC2C
858D96473FEC5DEFA4F1ED703FD2CD011925A4883FD5CBF76B0C4BBD3FE18179
5C3902B03FC1C1A5 FE047CC3FC61342AC78FEEB3FEA4A14A75EED3A3FDF2F7B
 3B2C0A03FE43F09 5ECCF4C3FC76797FBBA40D33FE9A1119FAA856D3FE6BA53
69874A403FB6157F5AD63DC13FEC389DB95D9FE53FE280A110A453703FA3C34E
6D56C5A43FD71B005B72FB5C3FE74BE8 AB682DD3FE73982 9F3D4E03FE533A3
20505BB43FD2FD159B2AF2E43FDCEC89F2F9B1A33FE3AB47 AE185F03FCBE666
14842CCE3FD618F9
```

- These files contain the values of the 97 seeds of RM64 in hexadecimal:

- The **initial seed** is controlled by the **`RANDOMIZe`** card

- In Flair (more in the next lecture):

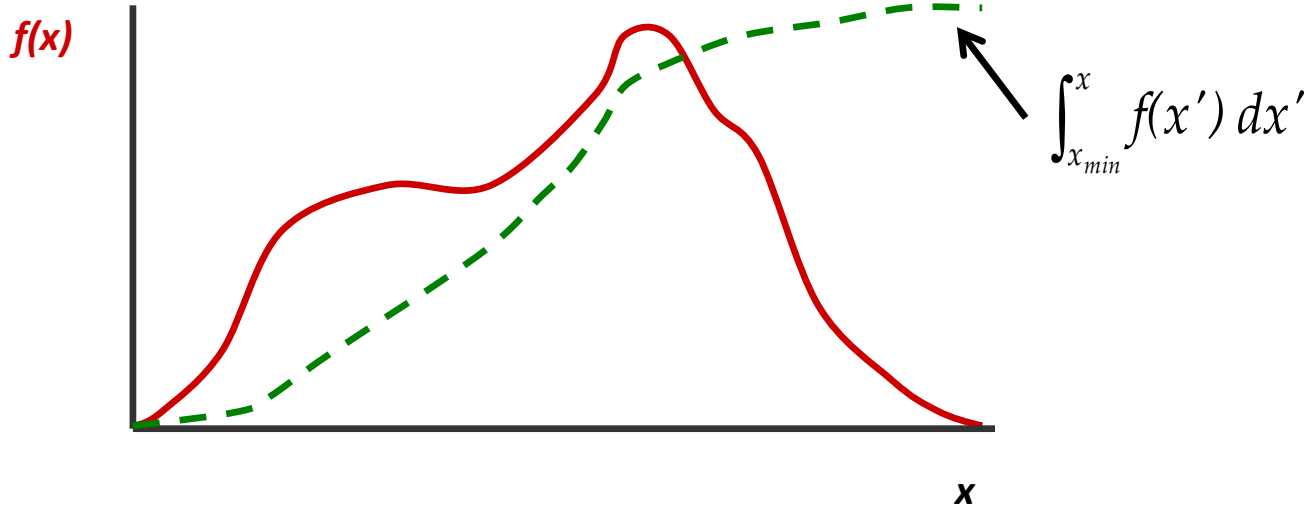


Sampling from arbitrary distributions

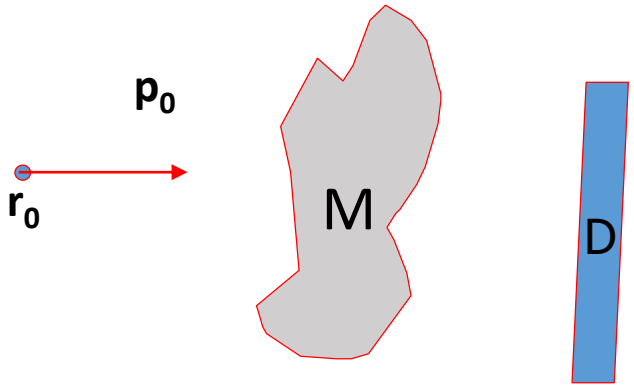
- In Monte Carlo we sample: step lengths, event type, energy losses, deflections…
- **Sampling**: generation of random values according to a given distribution.

- Fundamental problem: we know how to sample uniformly distributed values, but how do we sample from arbitrary distributions?

- There's a whole array of **sampling techniques**:
  - Inverse sampling
  - Rejection sampling
  - …



Basic Monte Carlo simulation algorithm

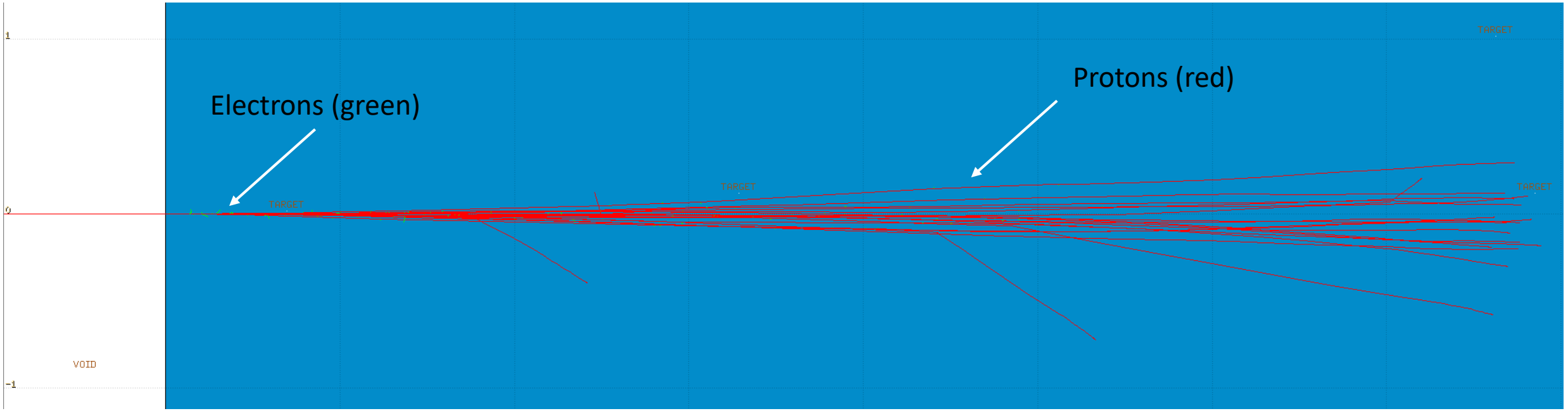


Loop over n_p primary events:

1. Initialize source particle position and momentum.
2. If particle is in vacuum, advance it to next material boundary.
3. Determine total interaction cross section at present energy and material: σ
4. Evaluate the mean free path to the next interaction: $\lambda = 1/(N\sigma)$
5. Sample step length to next interaction from $p(s) = (1/\lambda)\, e^{-s/\lambda}$
6. Decide nature of interaction: $P_i = \sigma_i / \sigma, \quad i=1,2,\ldots,n$
7. Sample energy loss (and/or change of direction) from differential cross section for the selected interaction mechanism i. Update energy and direction of motion.
8. Add generated secondary particles to the stack if any.
9. Score contribution of the track/event to the desired physical observables.
10. Go to 2 unless:
    - Particle energy drops below user preset threshold (see lecture on Friday)
    - Particle exits the geometry

Example: 100-MeV proton beam on water

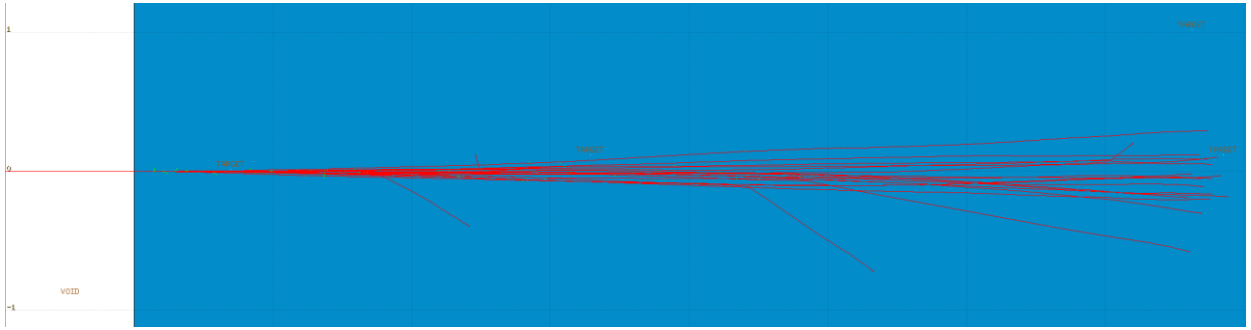
10 simulated proton trajectories in water (E_p=100 MeV):

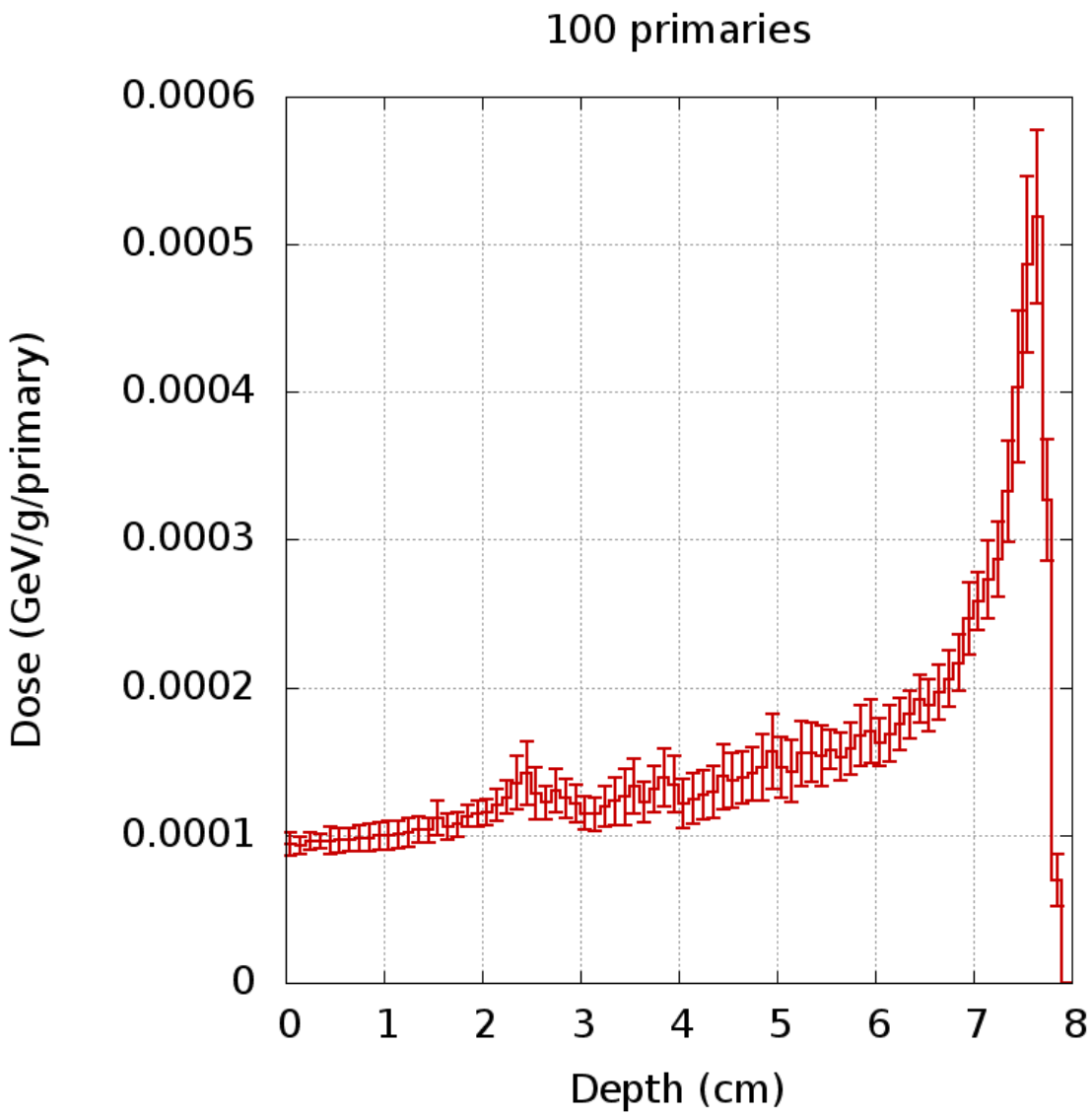


Statistical uncertainties

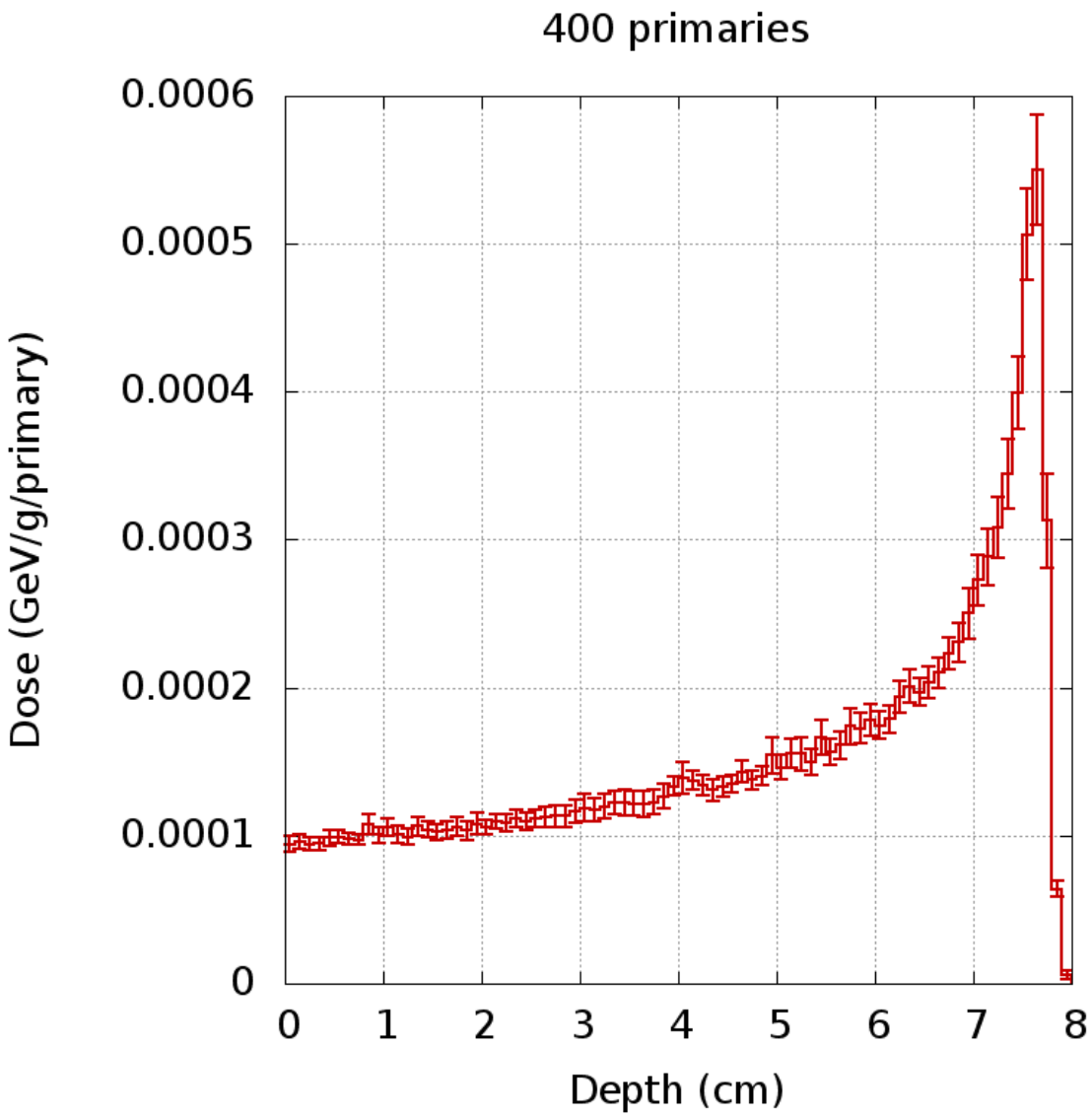
- Results from Monte Carlo simulations are affected by statistical uncertainty
- How does it depend on the number of simulated primary particles?

Example: 100-MeV proton beam on water





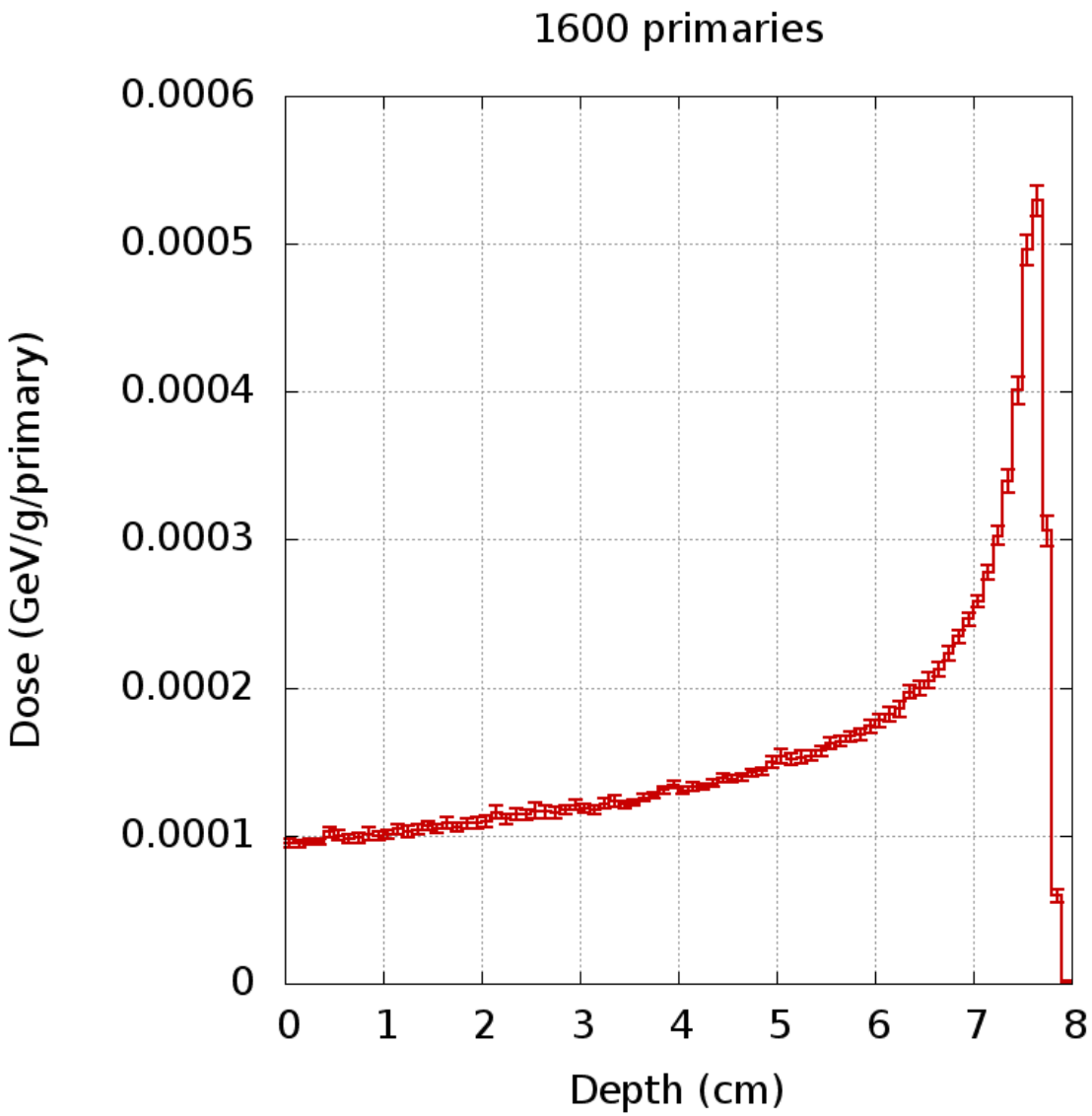
Example: 100-MeV proton beam on water



Example: 100-MeV proton beam on water

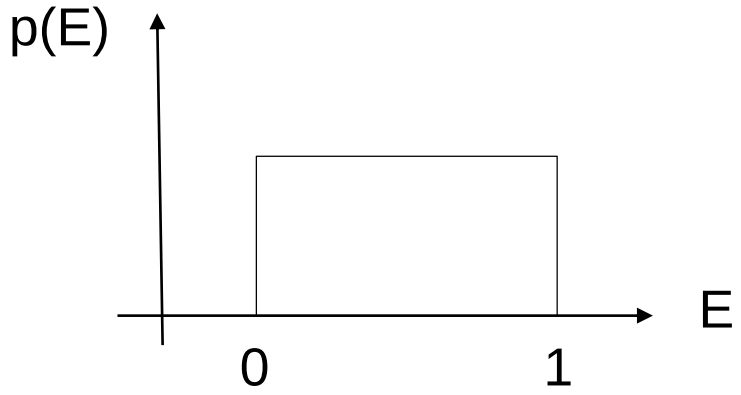
- Results from MC simulations are affected by statistical uncertainty

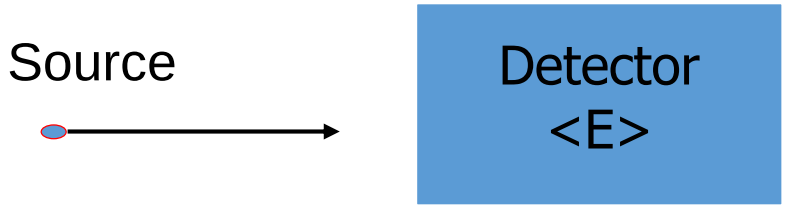
- The larger the number of primaries, the smaller the error bars.



A numerical experiment

- Imagine a source emitting particles with a flat energy distribution which deposit all their energy in a detector.
- Let the detector/estimator measure the average deposited energy:

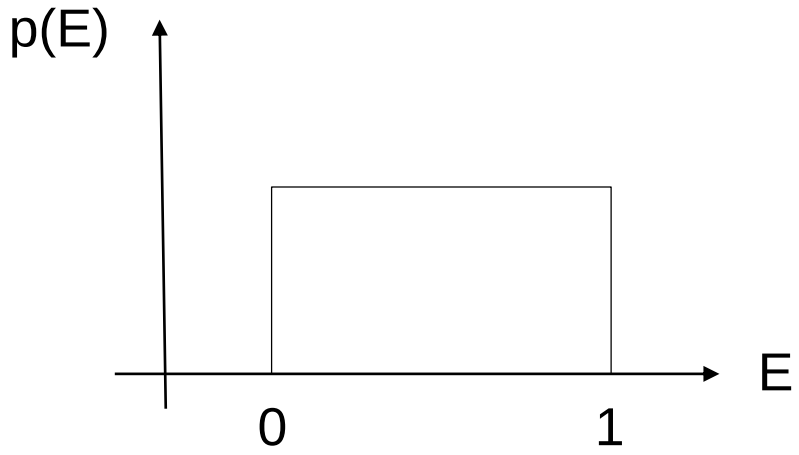


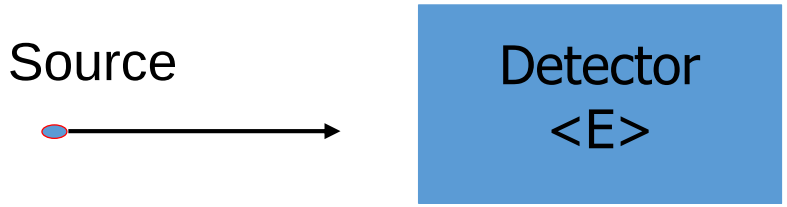


$$\langle E \rangle = \frac{1}{N} \sum_{i=1}^{N} E_i$$

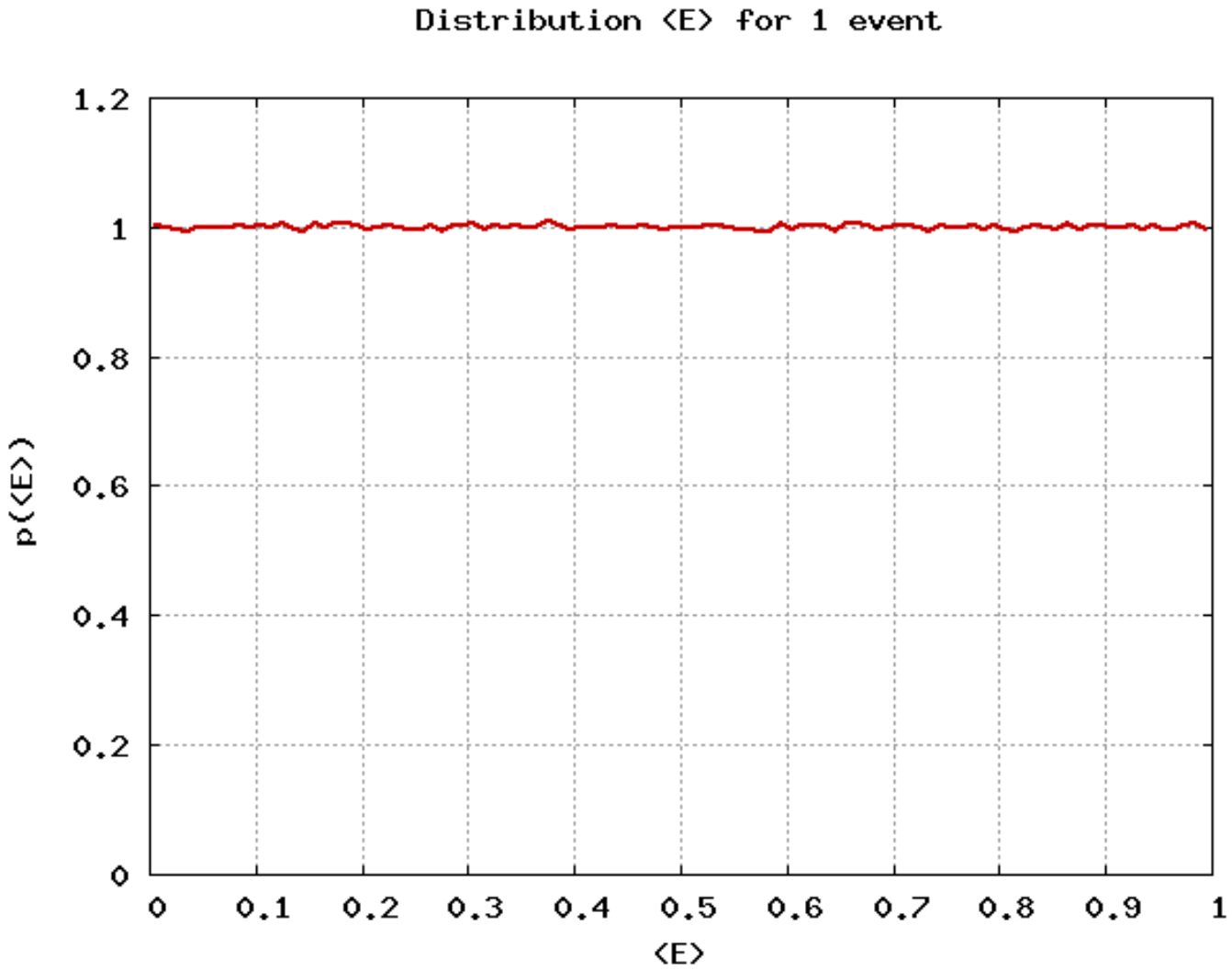
- What can one say about the estimated <E>?
  - It is a random variable
  - As such, it follows a certain distribution.
  - Which one? It depends on the number of events.

Distribution of <E> if source emits N=1 particle

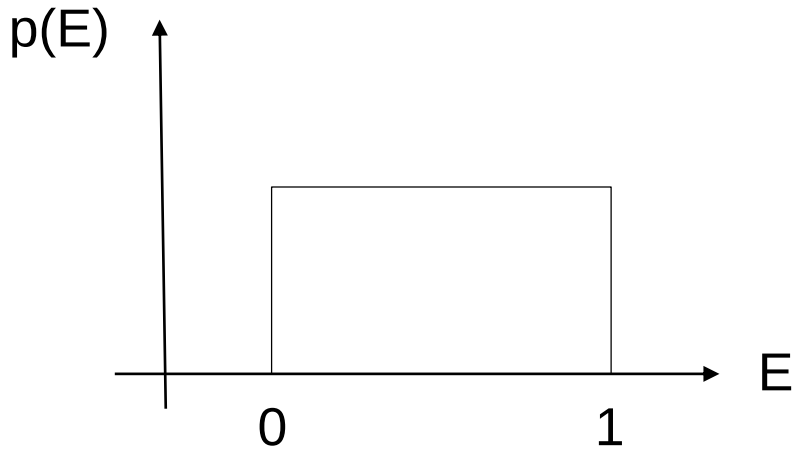


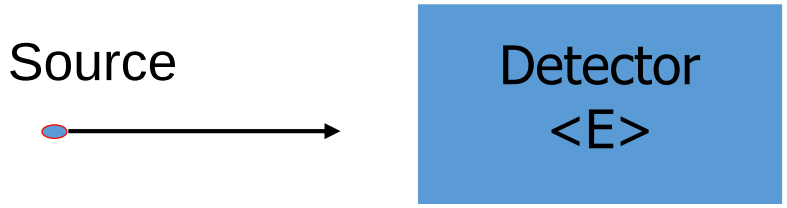


$$\langle E \rangle = \frac{1}{N} \sum_{i=1}^{N} E_i$$

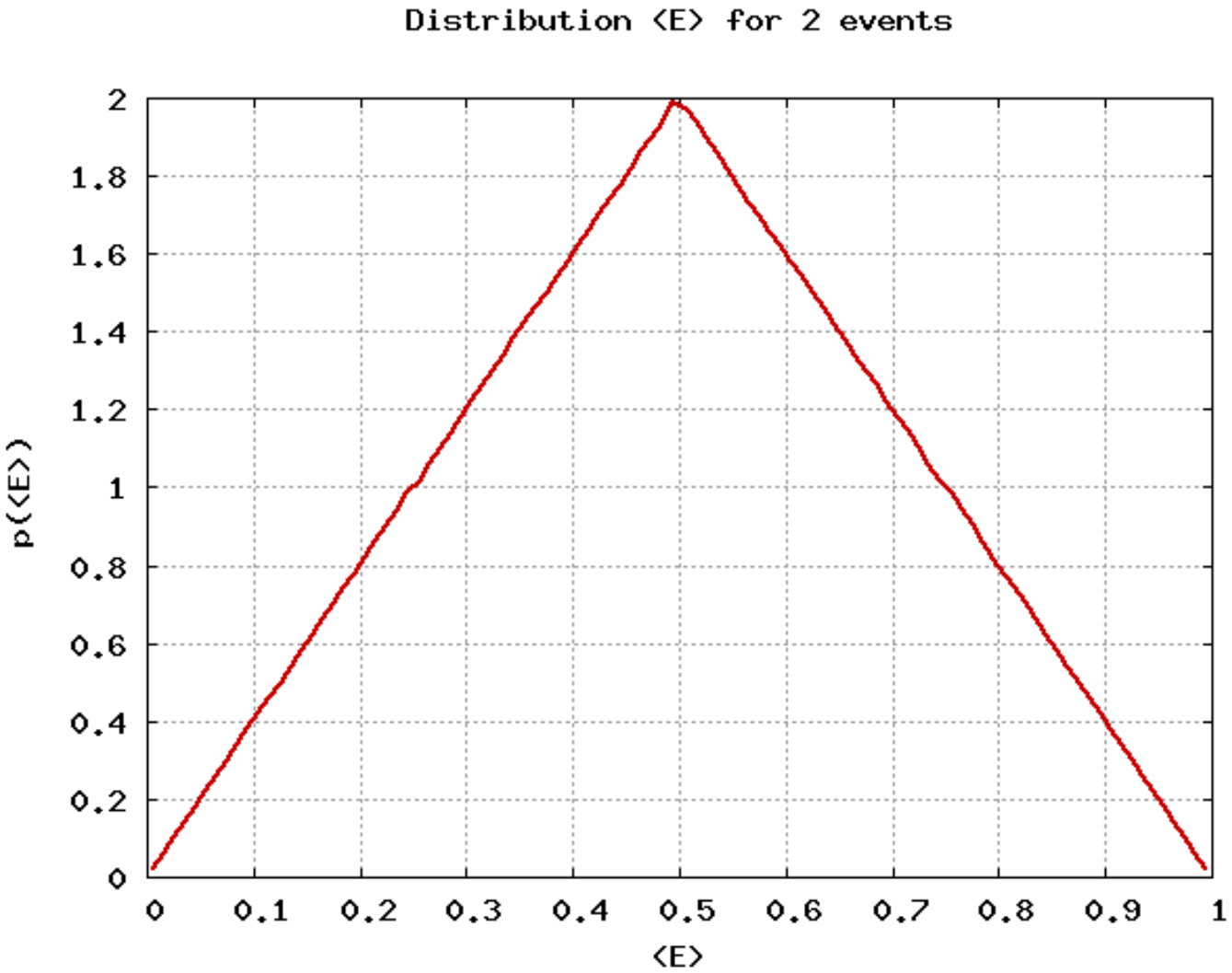


Distribution of <E> if source emits N=2 particles

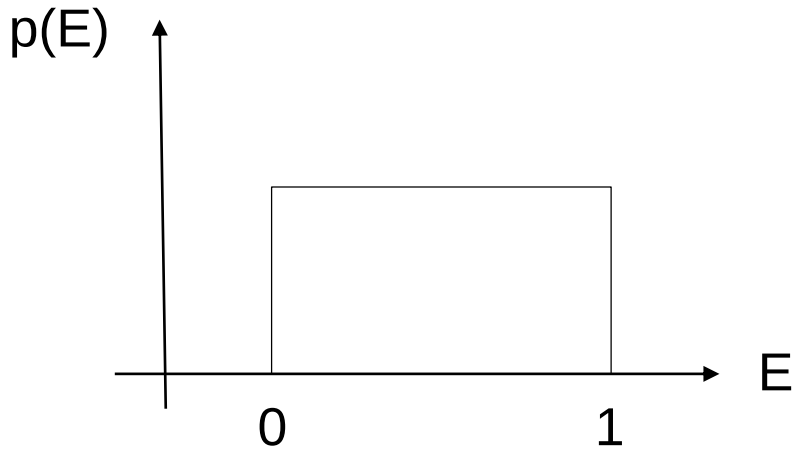


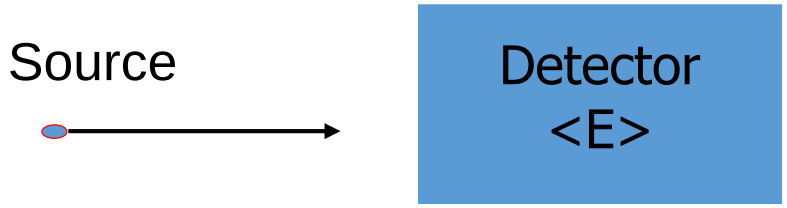


$$\langle E \rangle = \frac{1}{N} \sum_{i=1}^{N} E_i$$

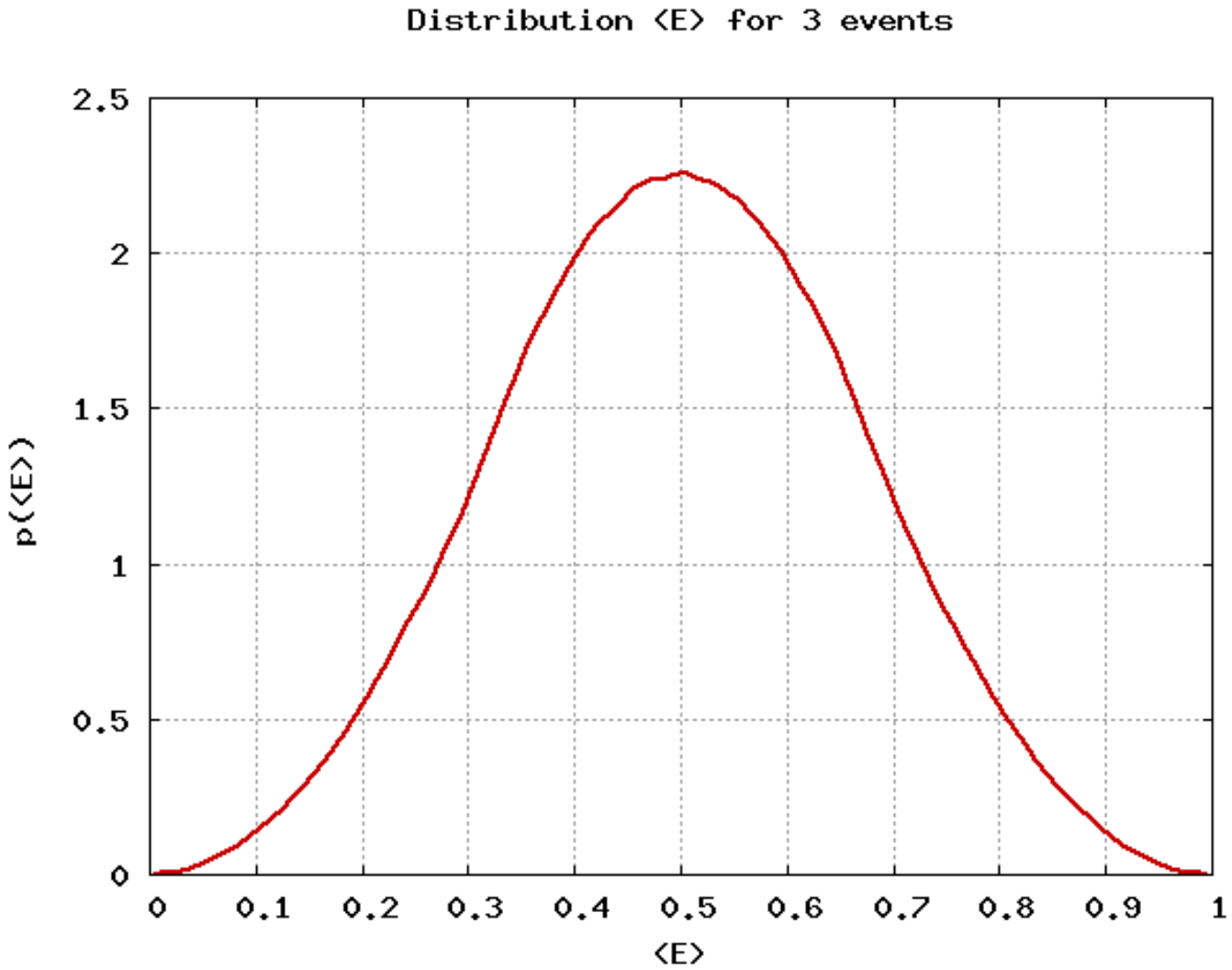


Distribution of <E> if source emits N=3 particles

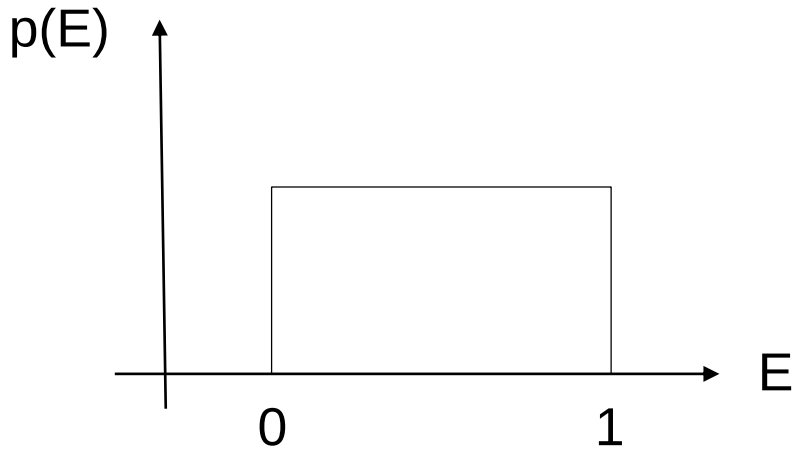


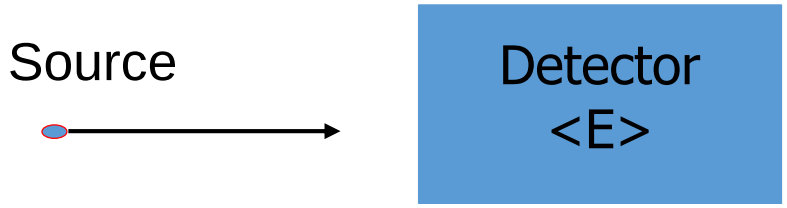


$$\langle E \rangle = \frac{1}{N} \sum_{i=1}^{N} E_i$$

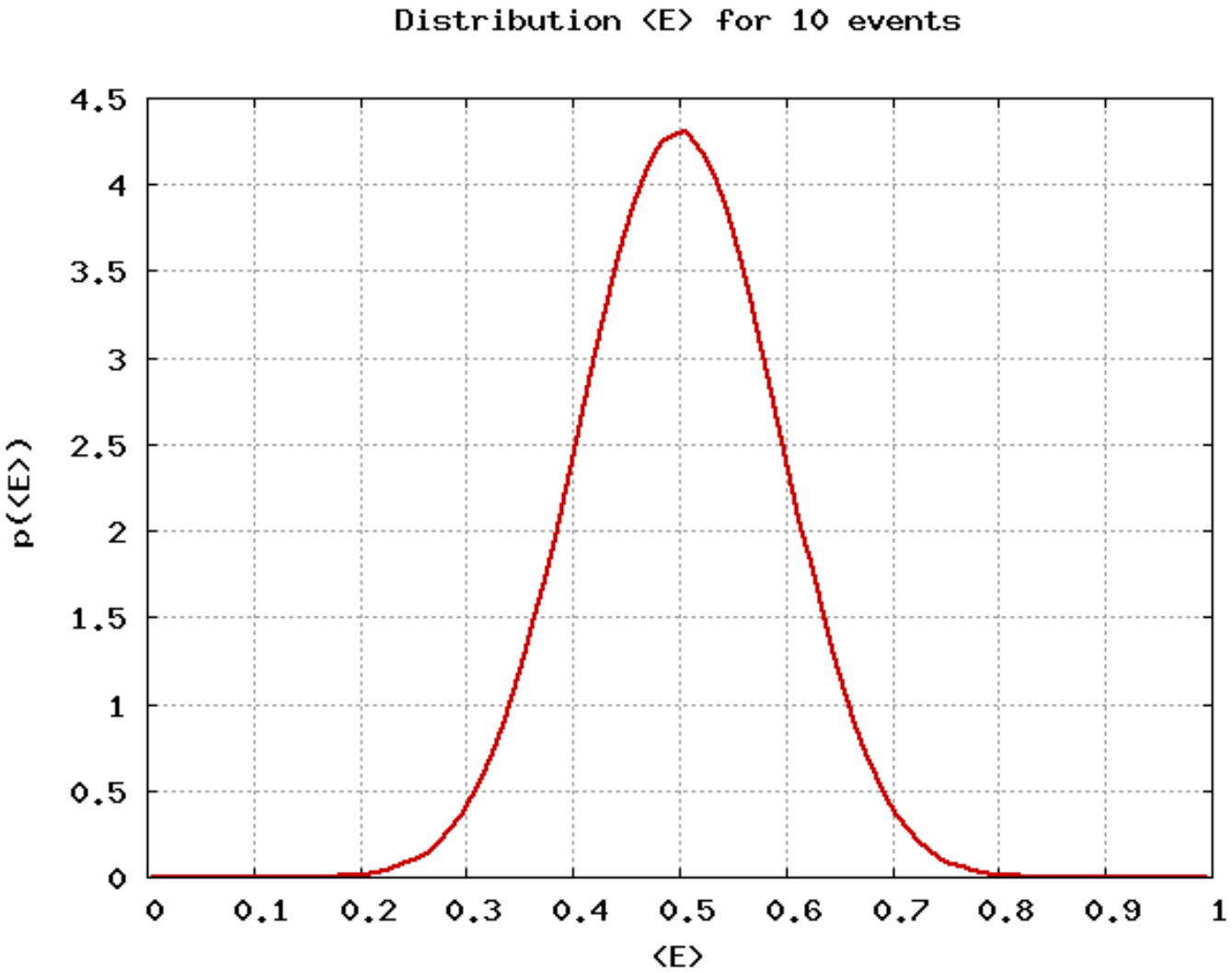


Distribution of <E> if source emits N=10 particles



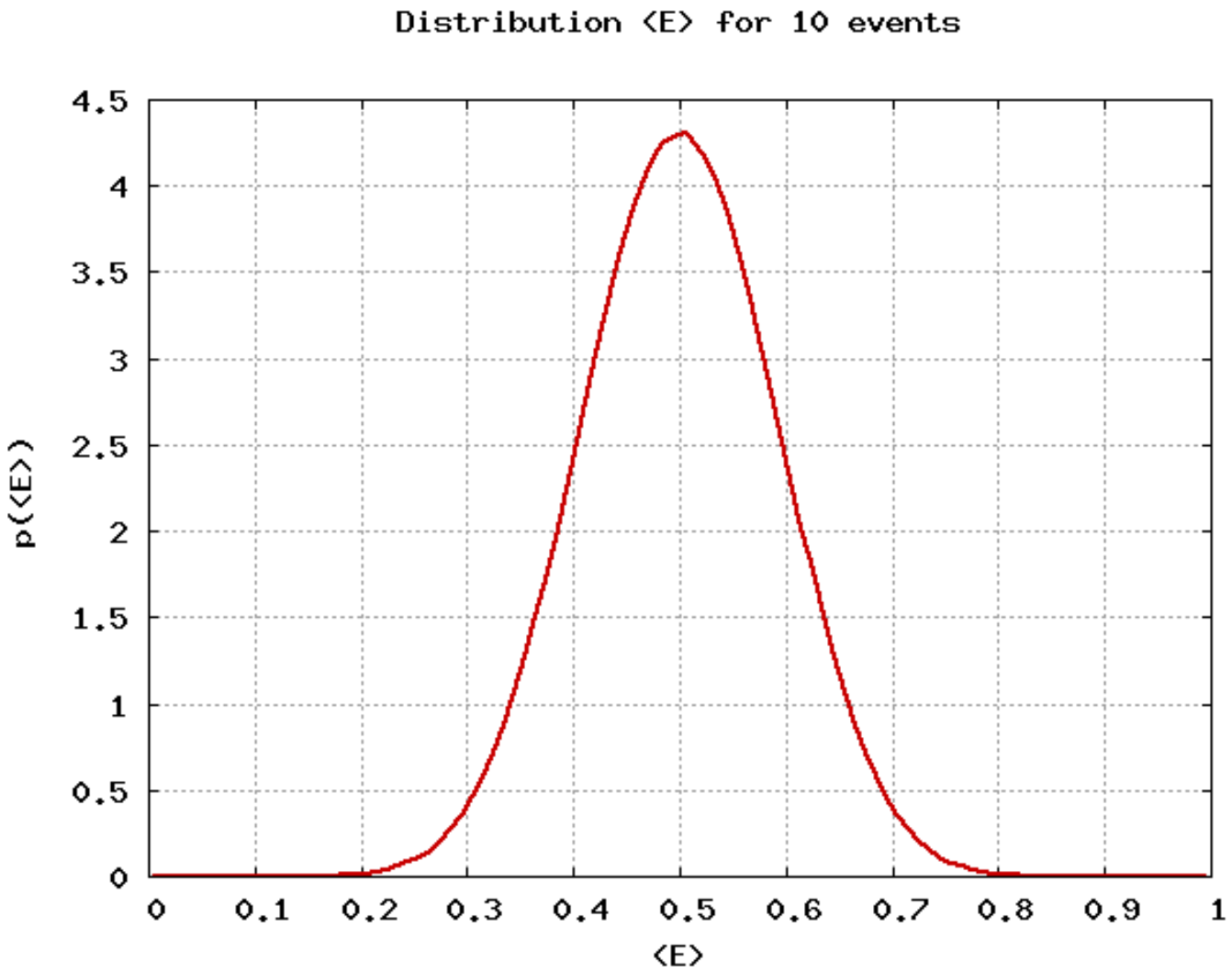


$$\langle E \rangle = \frac{1}{N} \sum_{i=1}^{N} E_i$$

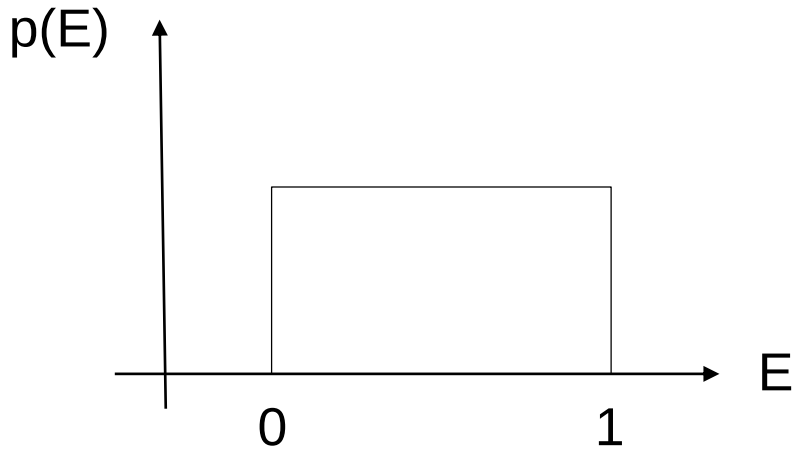


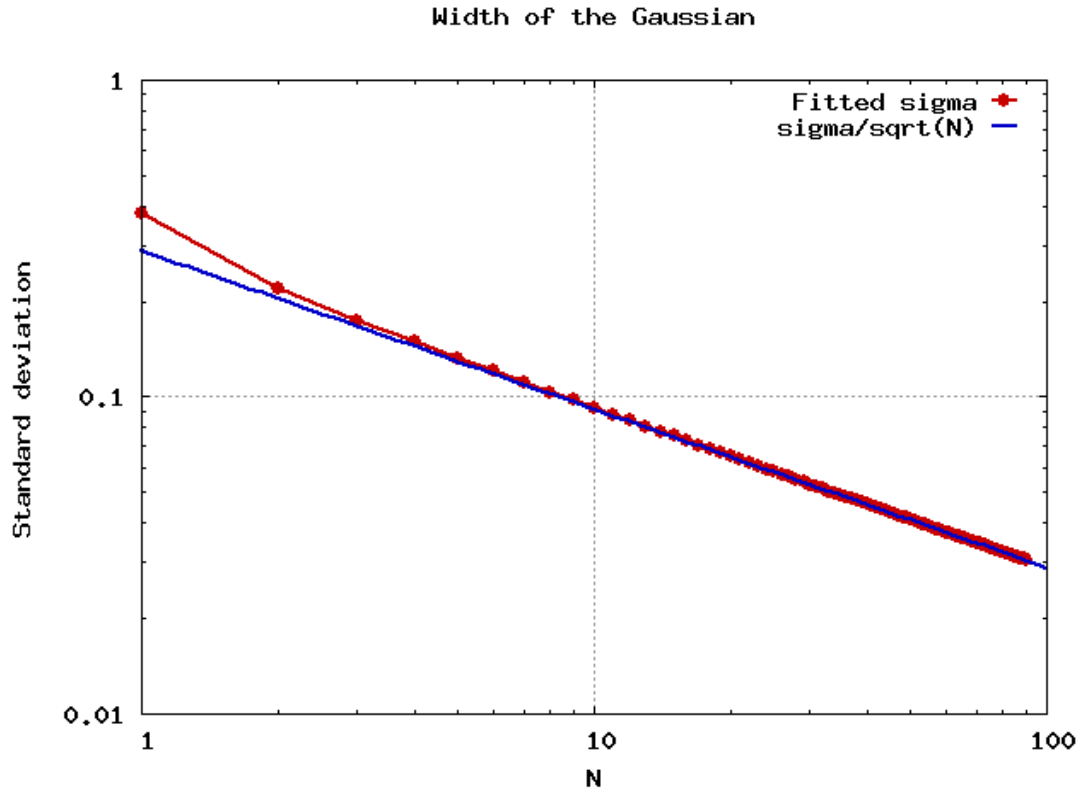
Distribution of <E>

- For sufficiently large number of contributions, the estimate mean <E> follows a Gaussian!
- The standard deviation (~width) of this Gaussian is a measure of the statistical uncertainty when estimating <E>.
- The standard deviation (statistical uncertainty) decreases with the number of contributions N
- We now check how the statistical uncertainty drops with N



Standard deviation





Source

Detector
<E>

$$\langle E \rangle = \frac{1}{N} \sum_{i=1}^{N} E_i$$

- For a very large number of samples the detector would yield the estimated mean <E>=1/2 with sigma=0.
- Statistical uncertainty decreases with the number of contributions N as 1/sqrt(N).

Central Limit Theorem

- In the numerical experiment above, the “detector” plays the role of the expectation value of any physical observable estimated à la Monte Carlo.

- If the estimator receives a sufficiently large number of contributions N, the distribution of its expectation value tends to a Gaussian centered around the true expectation value, with standard deviation that goes like 1/sqrt(N). That is, **the statistical uncertainty of a MC estimate reduces as 1/sqrt(N) with the number of primary events**.

- This is essentially the Central Limit Theorem.

- Note that:
  - When doing a Monte Carlo simulation, quoting a result without a measure of the statistical uncertainty is meaningless.
  - Quoting a result obtained with a low number of contributions is dangerous: the distribution of the mean may still be far from the Gaussian centered around the actual expectation value!

Single-history vs batch statistics

- In FLUKA, primaries are grouped in **cycles.**
- When you initiate a FLUKA run, you will indicate:
  - The number N of cycles.
  - The number n_i of primaries per cycle
- The variance of a scored observable X is evaluated via

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

  where:
  - n is the total number of primaries,
  - x_i is the average of the *i*-th cycle: $x_i = \sum_{j=1}^{n_i} \frac{x_{ij}}{n_i}$

- In the limit $N=n$ and $n_i=1$ the expression applies to single-history statistics.
- NOTE: if you run just one cycle (N=1), the above expression cannot be evaluated and FLUKA will return 100% uncertainty.

Recommendations

- Ensure you have 5-10 cycles of comparable size

- Remember that the variance itself is a random variable. E.g., runs with the same number of primaries but different random seeds will give different values of the variance. The larger the number of primaries, the smaller the difference (MC results for different seeds converge to the same value).

- It is wise to examine how convergence is attained: verify that error bars drop with 1/sqrt(N). Sudden/isolated spikes indicate poor sampling in some corner of phase space (see Biasing lecture).

- It is often a good idea to plot 2D and 3D distributions. The human eye is a good tool for judging statistical convergence of 2D/3D estimators!

Statistical uncertainty

Statistical errors, due to sampling (in)efficiency

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

(from an old version of the MCNP Manual)

- The MCNP guideline is based on experience, not on a mathematical proof. But it has been generally confirmed also working with other codes.

- Small penetrations and cracks in a geometry 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.

Assumptions, limitations, and sources of uncertainty

Assumptions and limitations

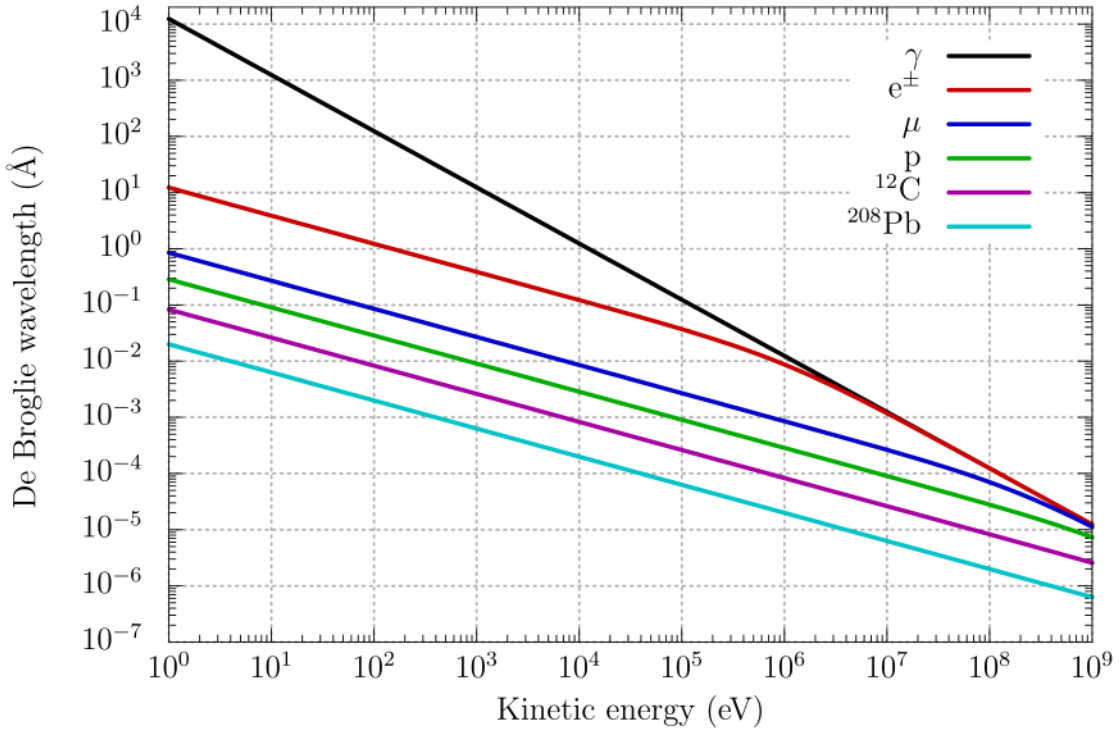
- Materials are static, homogeneous, and isotropic.
- Radiation transport is treated as a Markovian process: the fate of a particle depends only on its actual state, and not on its history.
- Material properties are not affected by previous histories.
- Particles follow trajectories and interact with individual atoms/electrons/nuclei.
  - A general order-of-magnitude measure: the particle's de Broglie wavelength must be small compared to typical interatomic distances (Angstroem).

Validity of the trajectory picture

- De Broglie wavelength:

$$\lambda_{\mathrm{dB}} = \frac{hc}{\sqrt{E_K(E_K + 2m_0c^2)}}$$

where E_K is the particle's kinetic energy, m_0 is its rest mass, h is the Planck constant, and c is the speed of light.



- Typical interatomic distances are in the order of ~Angstroem.
- E.g.: MC simulation of electron transport at energies much below 100 eV is questionable.
- The assumption of scattering on single target puts a lower energy bound on applicability of MC

Systematic uncertainties

- We have discussed **statistical uncertainties** above.
- That's only part of the uncertainty in the results of any MC simulation. The rest are **systematic uncertainties**, due to:

  - Adopted physics models: 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)
  - Transport algorithm: 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)
  - Cross-section data uncertainty: an error of 10% in the absorption cross section can lead to an error of a factor 2.8 in the effectiveness of a thick shielding wall (10 attenuation lengths). Results can never be better than allowed by available experimental data

* Not in FLUKA!

Systematic errors due to incomplete knowledge

- Systematic errors due to incomplete knowledge:
  - material composition not always well known. E.g. concrete/soil composition (how much water content? Can be critical)
  - beam losses: most of the time these can only be guessed. Close interaction with engineers and designers is needed.
  - presence of additional material, not well defined (cables, supports...)
  - Is it worth to do a very detailed simulation when some parameters are unknown or badly known?

- Systematic errors due to simplification:
  - Geometries that cannot be reproduced exactly (or would require too much effort)
  - Air contains humidity and pollutants, has a density variable with pressure

Errors, bugs, mistakes

- Monte Carlo codes can contain bugs:
  - Physics bugs
  - Programming bugs (as in any other software, of course)

- User mistakes:
  - mistyping the input: Flair is excellent at checking, but the final responsibility is on the user
  - 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

Summary

- Radiation transport is governed by the transport equation, a non-trivial integro-differential equation.
- The Monte Carlo method (exploiting pseudo-random numbers to simulate stochastic processes) is an expedient way to solve it for arbitrary radiation sources and material geometries.
- The pseudo-random number generator is at the core of the algorithm.
- Basic flow of a MC simulation.
- Results of MC simulations are affected by statistical uncertainty.
- The statistical uncertainty scales with the number of primary particles N as 1/sqrt(N)
- FLUKA expects a number of cycles and a number of primaries/cycle
- Basic assumptions and limitations


FLUKA

Additional material

The mean free path and its distribution

- Let n particles per unit time and surface impinge normally on a thin material slab of width ds with a density of N scattering centers per unit volume, each having a cross sectional area σ.
- Number of particles that interacted: $dn = n N\sigma\, ds$.
- The interaction probability in ds: $dn/n = N\,\sigma\, ds$
- Let *p(s)* be the distribution of path lengths to the next interaction.
- The probability that the next interaction is within *ds* of *s* is
  $p(s) = [\,1 - \int_0^s ds'\, p(s')\,]\ (N\,\sigma) = \int_s^{inf} p(s')\,(N\,\sigma)\, ds'$
- The solution of this diff eq is
  $p(s) = (N\,\sigma)\, e^{-s(N\,\sigma)}$
- Thus, the path length to the next interaction follows an exponential distribution. The average distance to the next interaction is:
  $\langle s \rangle = 1/(N\,\sigma) = \lambda$,
  i.e., we recover the expression of the mean free path given above.