LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
FACULTY OF PHYSICS
CHAIR OF EXPERIMENTAL PHYSICS
MEDICAL PHYSICS

Computational methods for Medical Physics

Lecture 3
Monte Carlo Simulations

Dr. George DEDES

WS 2016-2017

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Past and next lectures' content

- Previous lecture:
  - Finalized our discussion about more advanced MC integration methods
  - Presented basic examples and principles of random numbers (generators)
  - Showed the connection between MC simulation and non-uniform sampling
  - Described the Inverse Transform method for non-uniform sampling
- This lecture:
  - Continue on non-uniform sampling with the Acceptance-Rejection technique
  - More on MC (particle) simulation:
    - Phase Space
    - Boltzmann equation
    - Particle transport
    - A last exercise on MC

External material

- Material form this lecture was taken from the following lectures or presentations:

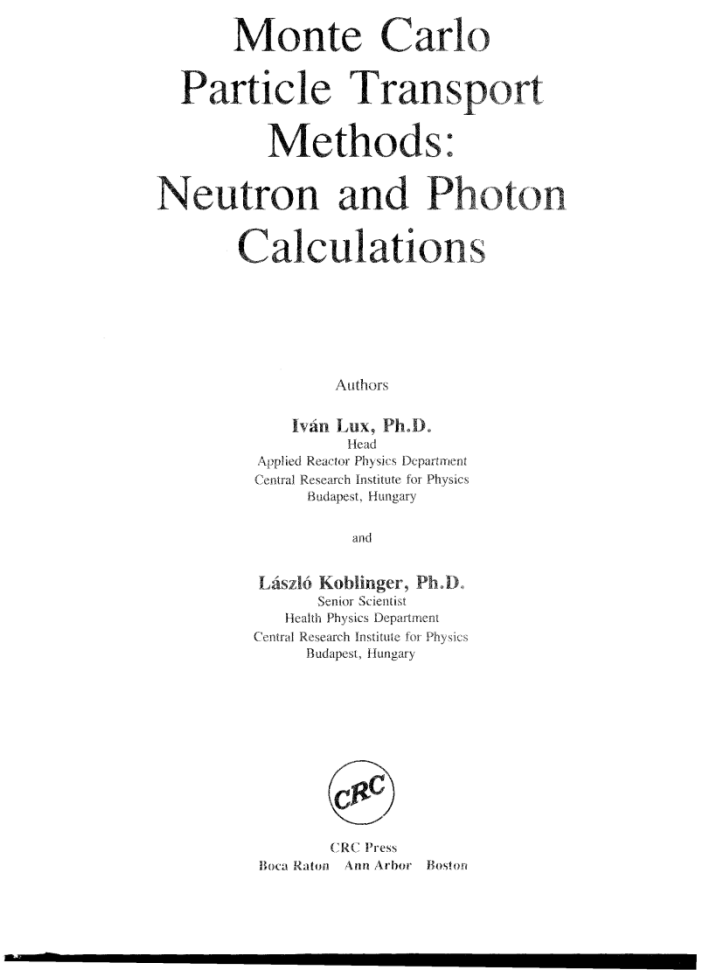
  - The FLUKA Collaboration
    14th FLUKA Course, Dresden, Germany 2013
    Statistics and Sampling
    Dr. Alberto Fassò
    https://www.fluka.org/free_download/course/dresden2013/Lectures/09_Statistics_and_sampling_1013.pdf

  - University of Sydney
    Institute of Medical Physics, School of Physics
    Monte Carlo Radiation Transport: Current Capabilities and Future Directions
    Dr. Zdenka Kuncic
    http://www.anzsnm.org.au/cms/assets/Uploads/Documents/Specialisations/Physics/PhysicsSIG2008-02-Kuncic.pdf

  - University of California, Berkeley
    Nuclear Engineering Department
    Monte Carlo Sampling Methods
    Dr. Jasmina Vujic
    http://web.ist.utl.pt/~ist11038/acad/theo/simul/Vujic.pdf

External material

- Books:
  - Monte Carlo Particle Transport Methods: Neutron and Photon Calculations
    I. Lux, L. Koblinger


Monte Carlo
Particle Transport
Methods:
Neutron and Photon
Calculations

Authors

Iván Lux, Ph.D.
Head
Applied Reactor Physics Department
Central Research Institute for Physics
Budapest, Hungary

and

László Koblinger, Ph.D.
Senior Scientist
Health Physics Department
Central Research Institute for Physics
Budapest, Hungary

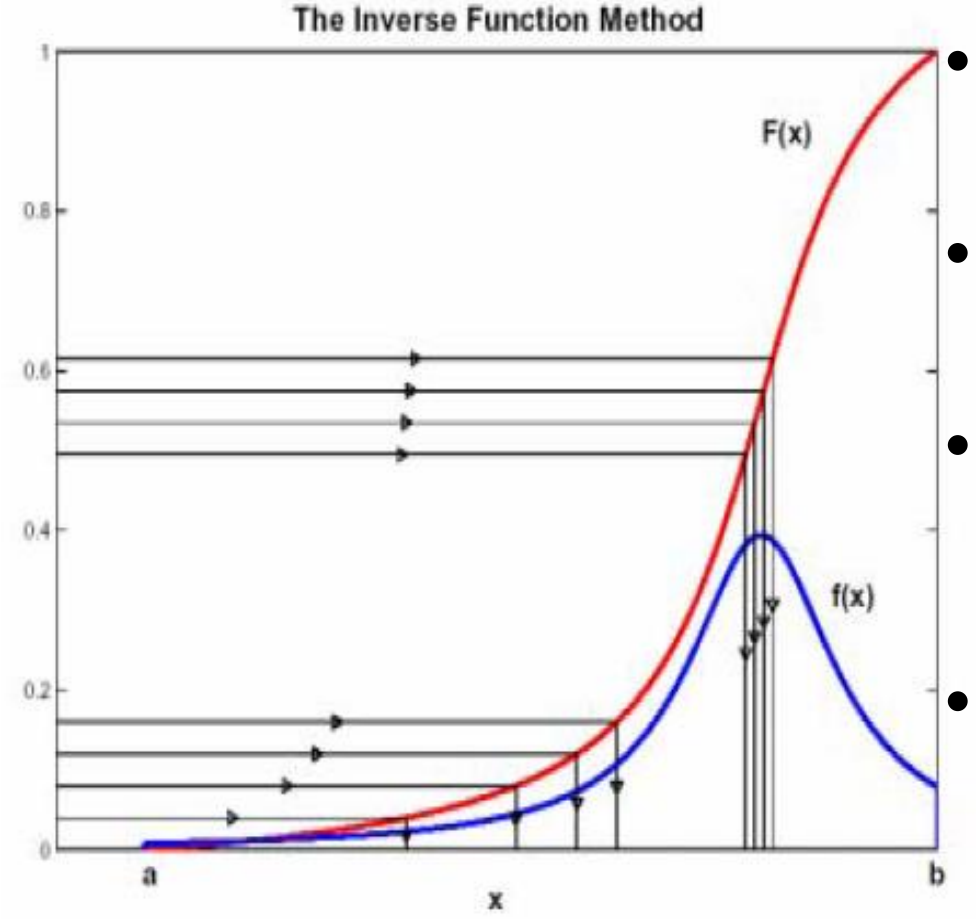
CRC

CRC Press
Boca Raton Ann Arbor Boston

https://gnssn.iaea.org/NSNI/Shared%20Documents/OPEN%20Shared%20Files/MonteCarloParticleTransportMethodsNeutronAndPhotonCalculations.pdf

Monte Carlo
Inverse Transform method

- With F(x) being the cumulative of f(x), the slope of F(x) is related to the shape of f(x)



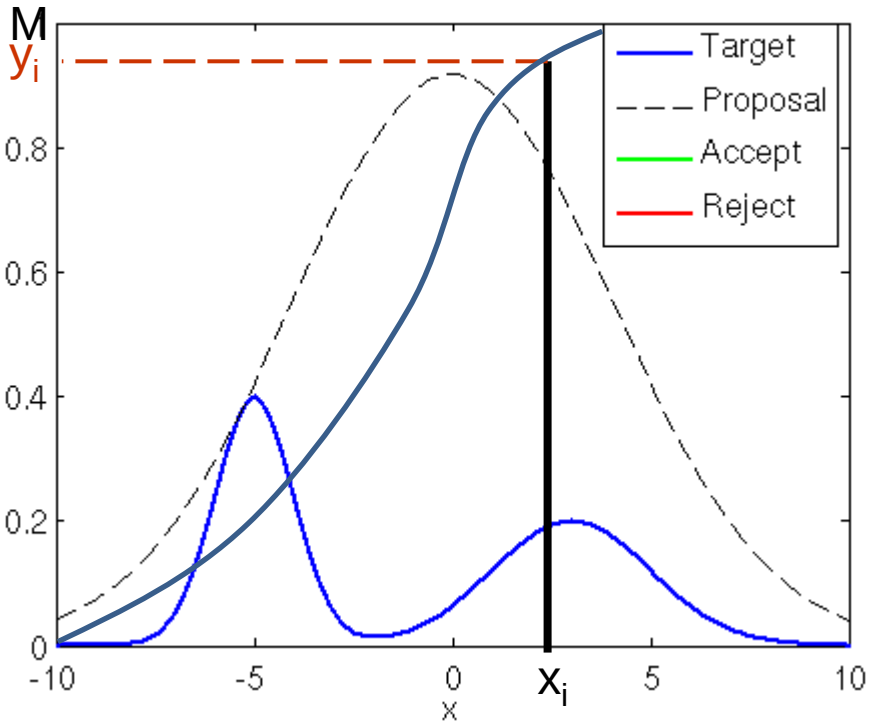
- By transforming a uniform y via $x=F^{-1}(y)$
- Achieve sampling in a non-uniform x
- Denser (more probable sampling) where F(x) has a steeper slope
- Higher probability to get a random x in the region where the pdf f(x) is higher

Monte Carlo
Inverse Transform method

- There is always a “but”:
- In order to use the Inverse Transform method, cdf F(x) has to be known and be simple enough to be inverted
- There is a good number of cases that the application of the Inverse Transform is impossible or too complicated
- In the next few slides we will present a flexible alternative called the Acceptance – Rejection method or simply Rejection method

Monte Carlo
Acceptance-Rejection method

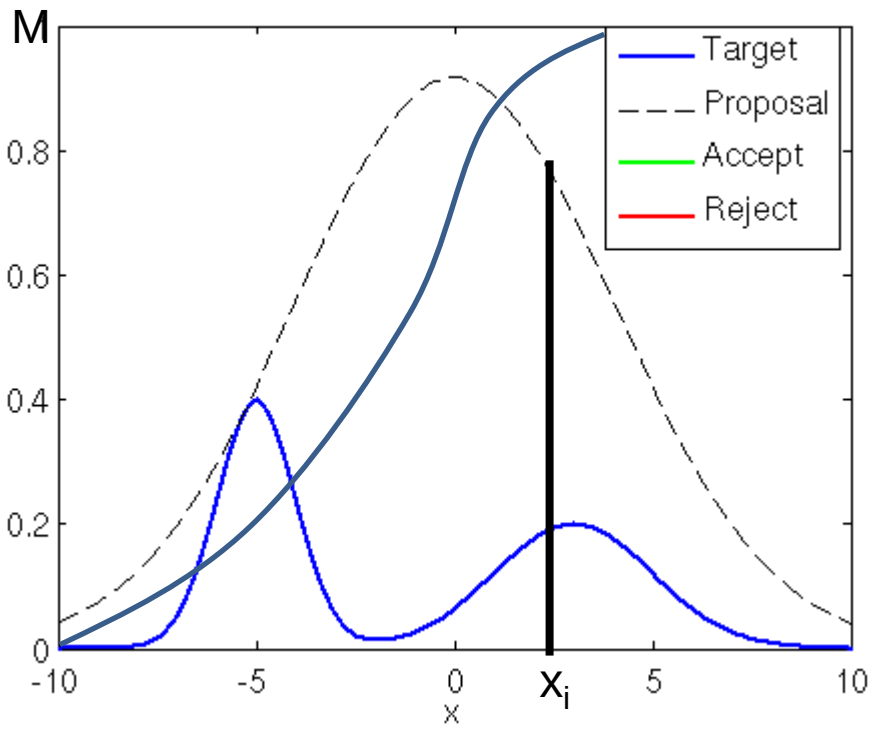
- Assume that we want to sample x from the pdf f(x), but F(x) quite complicated to cope with



- Find a proposal pdf $g(x)$ normalized for which cdf G(x) is invertible
- So that $Mg(x) \geq f(x)$ for any x and $M \geq 1$
- Generate uniform y_i U[0,M] this step is exactly an Inverse Transform for $Mg(x)$
- Transform to $x_i = G^{-1}(y_i)$

Monte Carlo
Acceptance-Rejection method

- Assume that we want to sample x from the pdf f(x), but F(x) quite complicated to cope with

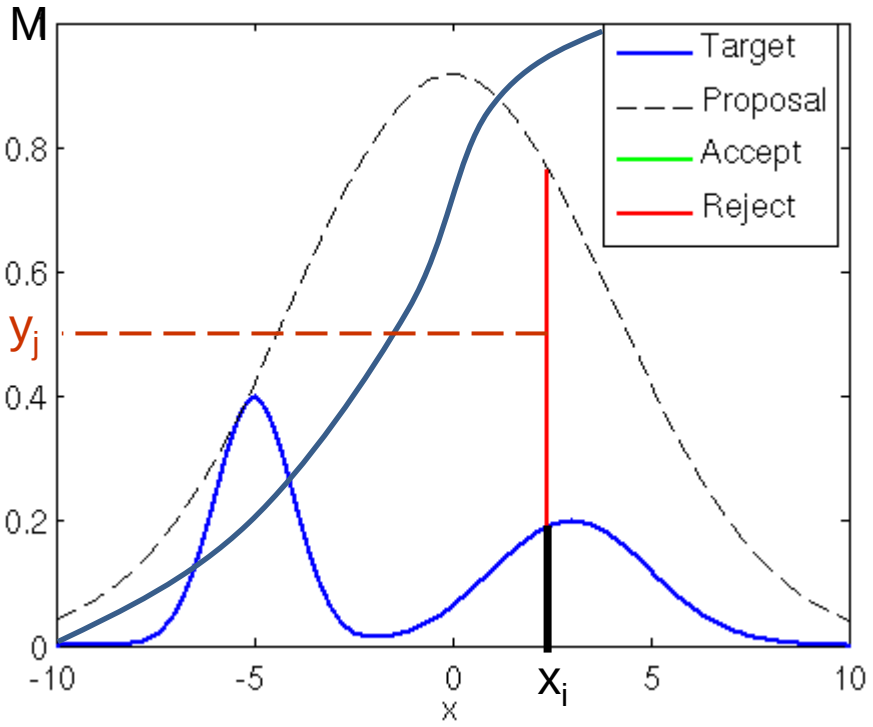


- The derived x_i is distributed according to $Mg(x)$
- As $Mg(x)$ envelopes f(x), x_i might be distributed according to f(x) or might not
- From all the x_i we have to get rid of the ones that are not distributed according to f(x)
- The ratio of x_i that "belong" to f(x) is $\frac{f(x_i)}{Mg(x_i)}$

- In order to decide whether we keep or not x_i according to the above ratio we draw another uniform random number y_j U[0,1]

Monte Carlo
Acceptance-Rejection method

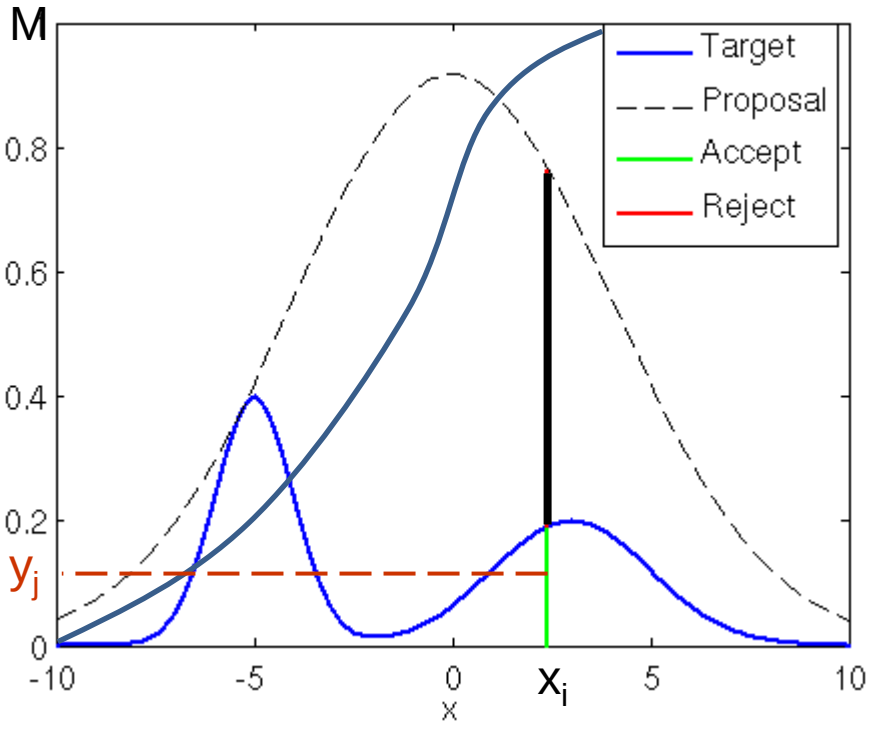
- Assume that we want to sample x from the pdf f(x), but F(x) quite complicated to cope with



- If $y_j > \frac{f(x_i)}{Mg(x_i)}$ then reject this x_i

Monte Carlo
Acceptance-Rejection method

- Assume that we want to sample x from the pdf f(x), but F(x) quite complicated to cope with



- If $y_j > \frac{f(x_i)}{Mg(x_i)}$ then reject this x_i
- If $y_j \leq \frac{f(x_i)}{Mg(x_i)}$ then keep it and

consider it as a random number distributed according to f(x)

Monte Carlo Simulation

- As mentioned in the last lecture, the Monte Carlo method was not originally a simulation method, but a technique to solve multidimensional integro-differential equations
- The basis of the technique is to build a stochastic process (use of random numbers) in order to solve this equation (we have seen how to solve integrals)
- It was soon realized that when this stochastic technique was applied to an equation describing a physical/mathematical stochastic process, then the technique could be identified with the process itself
- This is then called an analog Monte Carlo and is the basic notion of an MC simulation
- Every "step" in the technique/method corresponds to an identical step of the simulated process

- In the case of a fully analog MC simulation, the natural laws (or the description we have of them) are preserved

- We are "imitating" what could happen to the particle, we try to reproduce an exact and realistic particle history

- There is also the non-analog MC simulation. We need it in order to reduce computational time, acquire statistically more significant results for more interesting cases/events

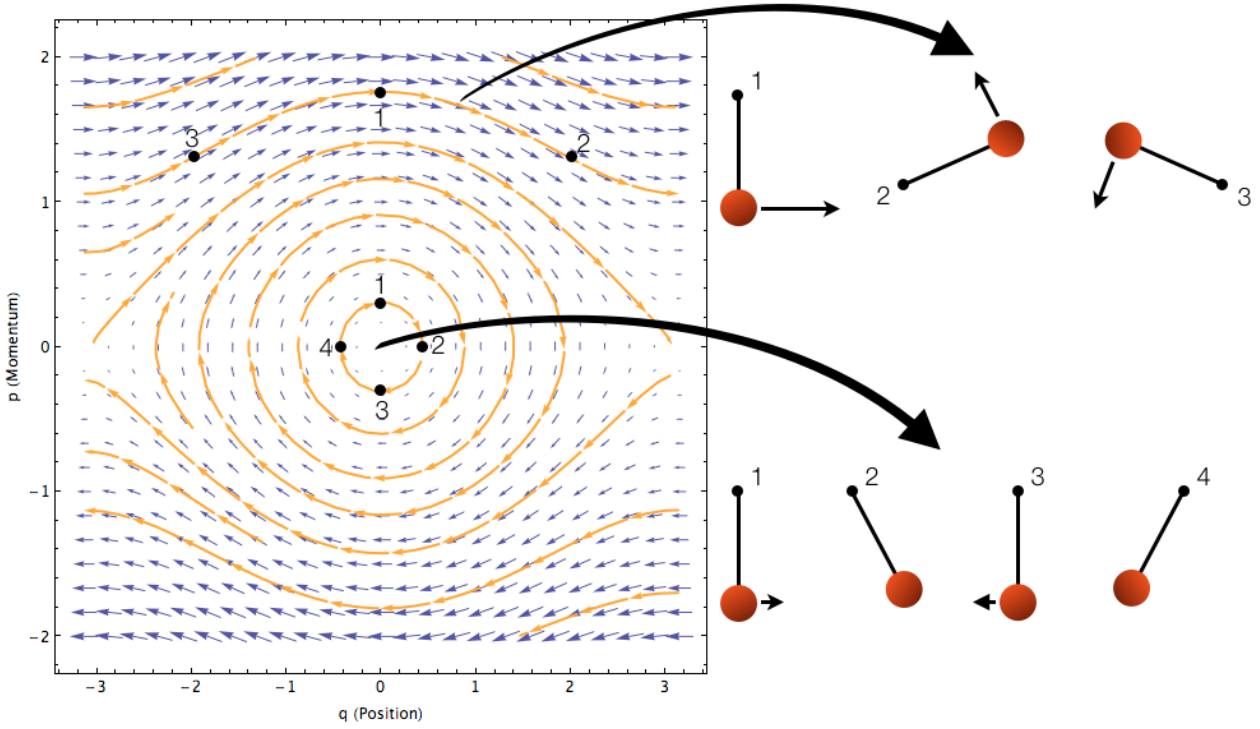
- In the case of non-analog or biased MC simulations, we "enhance" some probable outcomes, while we "suppress" some others, with the usage of biasing techniques:

  - Absorption suppression
  - Particle splitting
  - Russian roulette
  - Range rejection
  - Interaction forcing
  - ….

- Main components of an MC simulation:
  - Have a set of pdf's that describe our physical/mathematical system
  - A good uniform random number generator
  - One or more sampling rules (we described and worked with some simple ones during the last weeks). In this way we go from uniform random sampling to random "steps" that exactly reproduce the pdf of our system (ie particle interactions)
  - Scoring of the results
  - Error estimation. Variance determination as a function of at least the number of trials

Monte Carlo Simulation
Phase Space

- Phase Space is a concept of classical Statistical Mechanics
- Each dimension of the Phase Space represents a degree of freedom of the studied system
- For example, in the case of a simple pendulum, the momentum and the position can be the dimensions of a two-dimensional Phase Space



- Phase Space is a concept of classical Statistical Mechanics
- Each dimension of the Phase Space represents a degree of freedom of the studied system
- For the description of a particle:
  - 3 dimensions correspond to the position in real space: x, y, z
  - 3 dimensions correspond to the momentum: p_x, p_y, p_z
  - More dimensions can be the mass, particle type, spin, …
- Each particle is represented by a point in the Phase Space
- Time can be also considered as an extra dimension. Alternatively, it can be treated as an independent variable. In the latter case, the Phase Space will evolve in time and the variation of a point along a multi-dimensional trajectory will represent the "history" of a particle.

Monte Carlo Simulation
Boltzmann equation

- The Boltzmann equation is a conservation equation of particle "flow" in the Phase Space
- It involves the integration of particle interaction probabilities over position **r** (x, y, z), time t and over all directions **Ω** (Ω_1, Ω_2, Ω_3) => multidimensional integration

$$\frac{1}{v}\frac{d}{dt}\Psi(\boldsymbol{r},\boldsymbol{\Omega},E,t)+\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\Psi(\boldsymbol{r},\boldsymbol{\Omega},E,t)-\Sigma_t\Psi(\boldsymbol{r},\boldsymbol{\Omega},E,t)+S=\iint\Sigma(\boldsymbol{\Omega}'\rightarrow\boldsymbol{\Omega},E'\rightarrow E)\Psi(\boldsymbol{r},\boldsymbol{\Omega},E,t)dE'\,d\boldsymbol{\Omega}'$$

time dependence | translation | absorption source | interaction - scattering

- $\Phi(\mathbf{r}, \boldsymbol{\Omega}, E, t)$: particle fluence
- $\Psi(\mathbf{r}, \boldsymbol{\Omega}, E, t)$: derivative of fluence with respect to the phase space coordinates
- E : particle energy
- $\boldsymbol{\Omega}$: initial particle propagation direction (flight path)
- $\boldsymbol{\Omega}'$: particle propagation direction after interaction
- Σ_t : total cross section per unit path length
- $\Sigma(\boldsymbol{\Omega}, E)$: differential cross section per unit path length

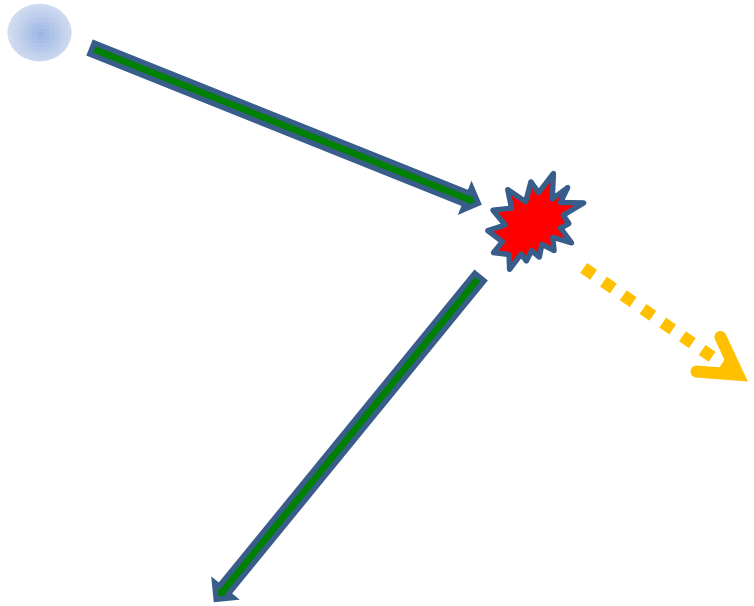
- All particle transport calculations are attempts to solve the Boltzmann equation
- The latter is a balance equation in Phase Space: at any phase space point, the change of the particle flux in an infinitesimal Phase Space volume is equal to:

Sum of all "production terms"
minus
Sum of all "destruction terms"

- Production: Sources, Translational motion in to the volume element, Scattering in, Particle production, Decay in to the volume element
- Destruction: Absorption, Translational motion out of the volume element, Scattering out, Decay outwards the volume element

- A schematic example of applying Monte Carlo to particle transport and interactions:

1) Each particle is created with some properties (E, position, direction etc), either predefined or selected from distributions using random numbers



2) The particle is transported along a step. The length of the step is decided by random selection (total cross section)

3) The type of interaction and the change of particle status is determined by random selection (partial cross sections, interaction kinematics, …)

4) Secondaries are stored for further simulation and a new particle step is again defined using random selection as in 1)

- Most of the interactions are described in a discrete way. The particle traverses a medium and according to the properties of the region and the particle an interaction takes place

- These kind of interactions are:

  - Atomic interactions:
    - Photons: Compton, photoelectric, pair production, …
    - Charged particles: bremsstrahlung, δ ray emission, …

  - Nuclear interactions:
    - Nuclear elastic scattering
    - Nuclear inelastic reactions
    - Absorption

  - Decays

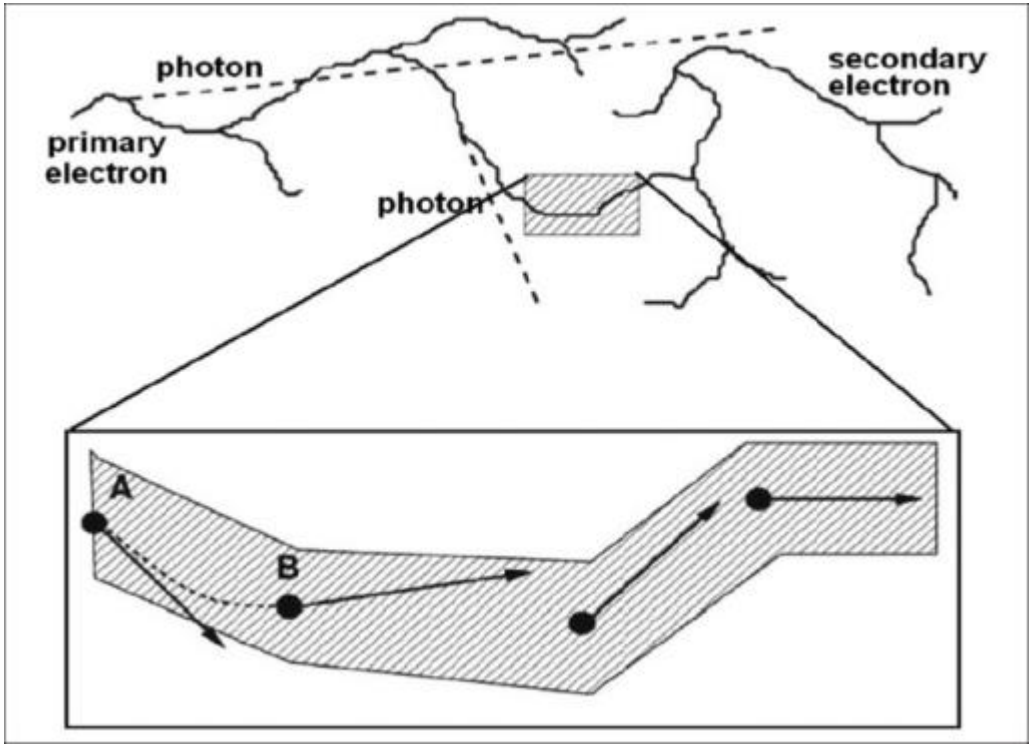
  - Geometrical “events”: boundary crossing, escape

Monte Carlo Simulation
Continuous processes

- Nevertheless, especially for charged particles, there can be a very high number of interactions along a single step. For example, charged particle energy loss and scattering is the result of thousands discrete collisions (very CPU time consuming)

- A way around this problem is to simulate many discrete interactions as a "continuous" process all along the step. For example, all the collisions with the atomic electrons are "condensed" into a single energy loss dE/dx along the step and all the changes in directions to an overall scattering angle

- This is called the condensed-history technique



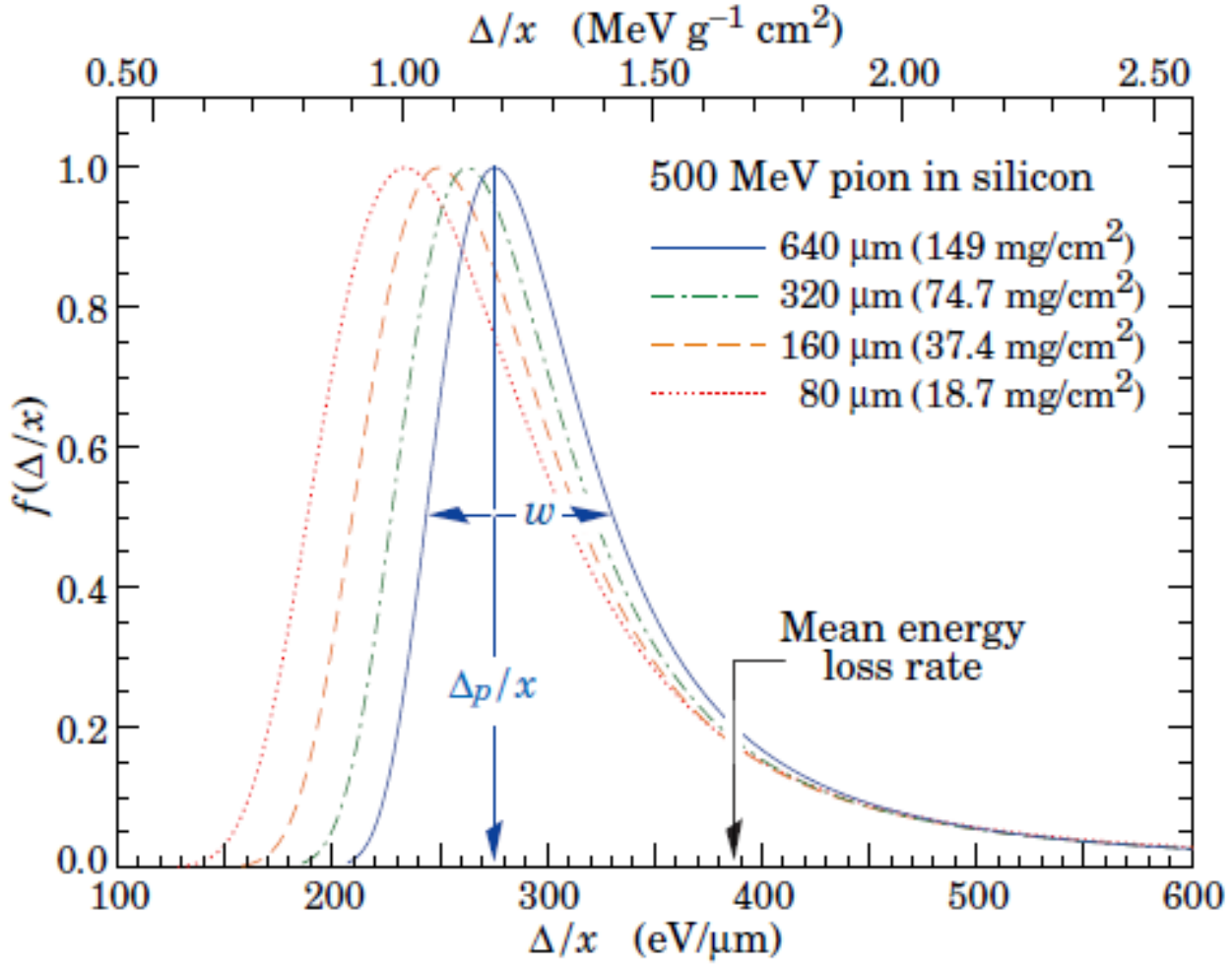
- Two common implementation examples of condensed history are:

- Ionization energy losses:
  - An energy threshold for δ electrons is applied. When the energy loss to be simulated is higher than this threshold, the δ electrons are explicitly simulated (discrete)

  - When the energy loss is lower than the threshold, no δ electron is simulated and the energy is distributed continuously along the step.

  - In order to simulate energy loss straggling in the case of continuous simulation (remember first lecture - Landau distribution of energy loss), fluctuations are sampled from the appropriate distribution

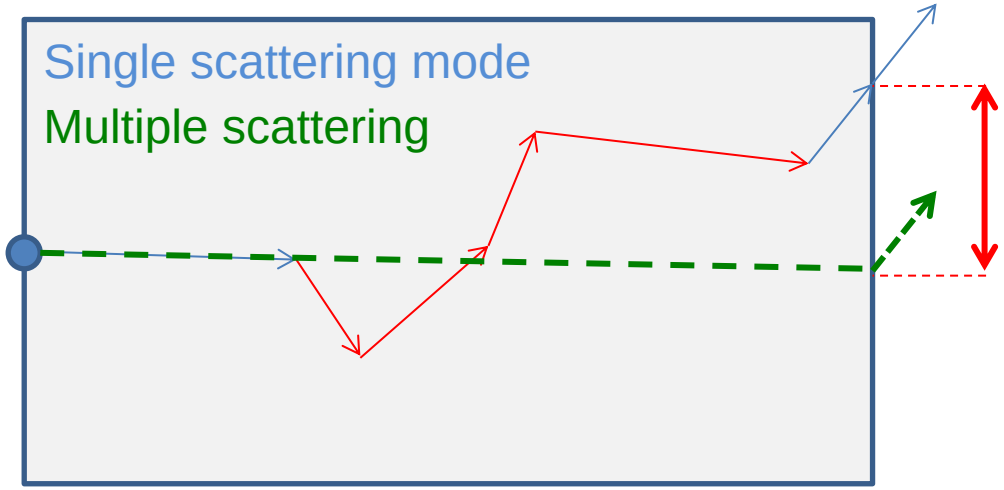
- Energy loss straggling:



- Two common implementation examples of condensed history are:

- Multiple Coulomb scattering:
  - Alternatively to single scattering mode, a total deflection angle, sampled from a theoretical distribution (see first lecture - Gaussian for small angle deflections), is applied to each particle step

  - Corrections applied to the path length and lateral displacement



Monte Carlo Simulation
Thresholds and Cut-off's

- As mentioned in the last few slides, when implementing modeling of a physical process, we might need to introduce thresholds and cut-off's
- They are needed in order to limit CPU time and also because of the limited validity of the physical models
- Two generic types of thresholds are the transport and production threshold:
  - Transport thresholds define what is the minimum energy, down to which we keep tracking a particle
  - They also depend on the scale of the geometry and the desired accuracy of our simulation, and of course on the particle type
  - Productions thresholds define what is the minimum energy, above which an explicit production/simulation of a particle happens
- Additional cut-off's can be used when we reach the limit of validity of a model and we either have to switch to another model or terminate particle tracking

Monte Carlo Simulation
Variance and CPU time reduction

- The figure of merit for the efficiency of a MC estimation ran in a CPU is: $\sigma^2 \cdot t$

- Where σ is the standard deviation of our estimation and t is the CPU time needed
- In order to increase the efficiency we use techniques that either aim in reducing the variance or the CPU time (or combinations of both)
- Often reducing σ increases t and vice versa
- A careful selection is needed otherwise we produce unreliable results or explode CPU consumption
- The methods that reduce variance are called variance reduction techniques

Monte Carlo Simulation
Basic rules of biasing

- The basic ideas of biasing can be explained from the Boltzmann equation

$$\frac{1}{v}\frac{d}{dt}\Psi(\boldsymbol{r},\boldsymbol{\Omega},E,t)+\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\Psi(\boldsymbol{r},\boldsymbol{\Omega},E,t)-\Sigma_t\Psi(\boldsymbol{r},\boldsymbol{\Omega},E,t)+S=\iint\Sigma(\boldsymbol{\Omega}'\rightarrow\boldsymbol{\Omega},E'\rightarrow E)\Psi(\boldsymbol{r},\boldsymbol{\Omega},E,t)dE'\,d\boldsymbol{\Omega}'$$

- which has two ingredients: particle fluence derivative Ψ and various operators acting on Ψ

- Both ingredients can be biased, leading to the assignment of a statistical weight to each particle

- Actual Ψ can be replaced by a fictitious Ψ'. The weight w must be such that Ψ=w Ψ' For example, each particle can be duplicated by assigning a statistical weight of 0.5 to each new particle

- Operators are based on probability distribution functions (pdf)
  - Total cross section Σ is probability of interaction per unit distance
  - Differential cross section Σ(Ω,E) is probability distribution in energy and angle
  - Sources are space-energy-angle pdf's
  - …

Monte Carlo Simulation
Basic rules of biasing

- The basic ideas of biasing can be explained from the Boltzmann equation

$$\frac{1}{v}\frac{d}{dt}\Psi(\boldsymbol{r},\boldsymbol{\Omega},E,t)+\boldsymbol{\Omega}\cdot\boldsymbol{\nabla}\Psi(\boldsymbol{r},\boldsymbol{\Omega},E,t)-\Sigma_t\Psi(\boldsymbol{r},\boldsymbol{\Omega},E,t)+S=\iint\Sigma(\boldsymbol{\Omega}'\rightarrow\boldsymbol{\Omega},E'\rightarrow E)\Psi(\boldsymbol{r},\boldsymbol{\Omega},E,t)dE'\,d\boldsymbol{\Omega}'$$

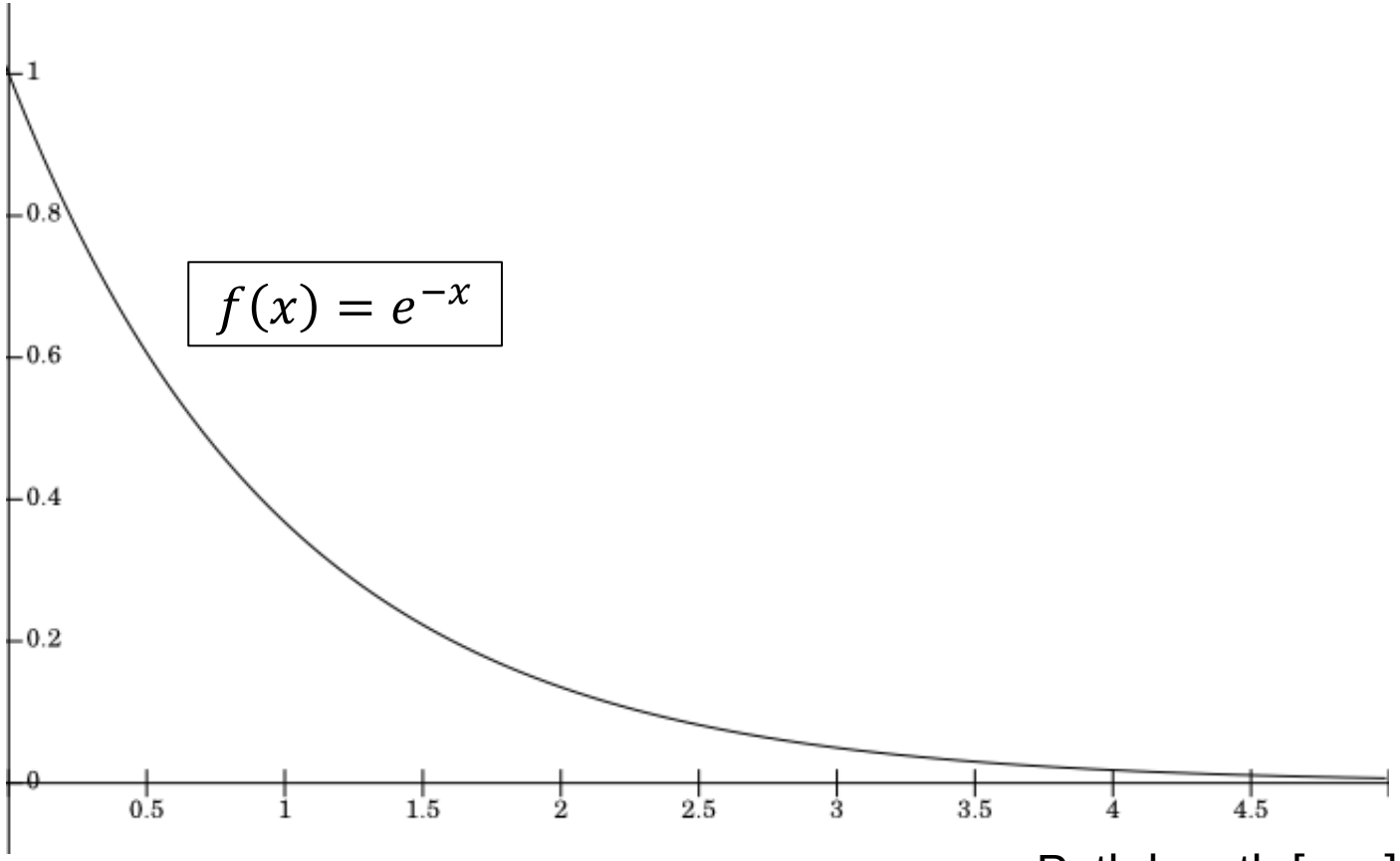
- which has two ingredients: particle fluence derivative Ψ and various operators acting on Ψ
- Both ingredients can be biased, leading to the assignment of a statistical weight to each particle
- Actual Ψ can be replaced by a fictitious Ψ'. The weight w must be such that Ψ=w Ψ' For example, each particle can be duplicated by assigning a statistical weight of 0.5 to each new particle
- Operators are based on probability distribution functions (pdf)
- The pdf of an operator can be changed as P=w P'. For example for each photon, duplicate the cross section for photoelectric effect and assign a statistical weight of 0.5 to each new photoelectron

Monte Carlo Simulation
Programming Exercise

- Programming Exercise 4:
- Let's build our own simple (simplistic) MC simulation of particle transport
- Modelling of exact physical processes can be cumbersome. So assume simulation of a fictional particle (you name it)
- Write a small MC simulation program (I would again recommend using one of the compiled rather than the interpreted languages) that does the following:
  - "Builds" a small, two dimensional homogeneous world with size 105x100 mm from x=-5 to x=100 and y=-50 to y=50
  - Creates a source of fictional particles at (x,y)(0,0) that are emitted towards the X axis
  - Assume that this type of particle has only two types of interaction
    - Absorption
    - Elastic scattering

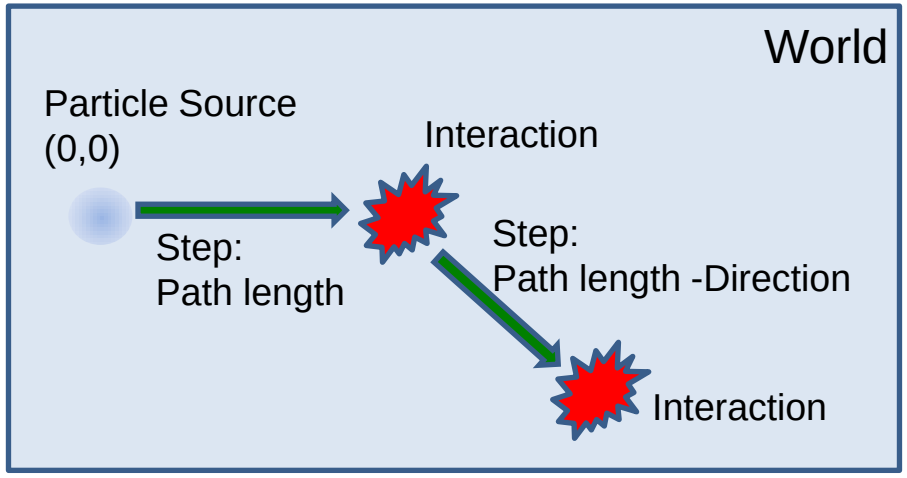
- Programming Exercise 4:

- Distribution of particle path lengths



Monte Carlo Simulation Programming Exercise

- Programming Exercise 4:



1) The particle source for you is simply the starting position in space of your simulation

2) Then you will have to transport your particle to the next interaction point:
   - "Theory" shows that the particle path length is a stochastic process of exponential form $f(x) = e^{-x}$, x (0,∞)

3) You have then to decide what process will take place (remember you have two
   - Possibilities – Absorption or Elastic scattering). Again our "theory" tells us that there is an 80% probability for Elastic scattering and 20% for Absorption

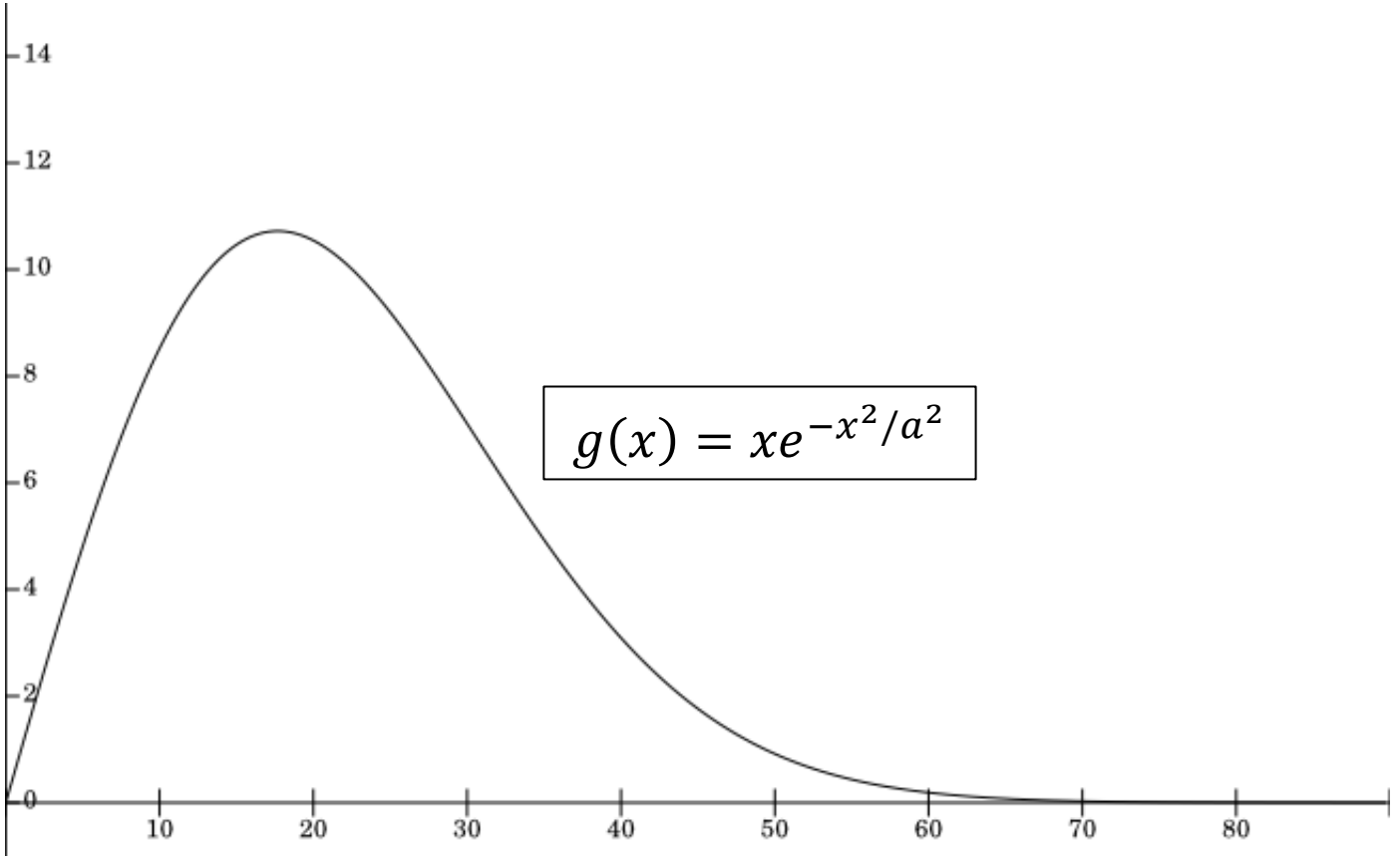
   - If the particle is absorbed, its "history" ends there and you proceed to the next particle from the source position and with direction X (step 1)

4) If the particle is scattered, you will have to decide which direction:
   - Past scattering "experiments" with this particle show that the distribution of angles can be fitted with: $g(x) = xe^{-x^2/a^2}$, x [0,180°] and α=25°
   - Resample next interaction point (step 2 and continue)

Monte Carlo Simulation
Programming Exercise

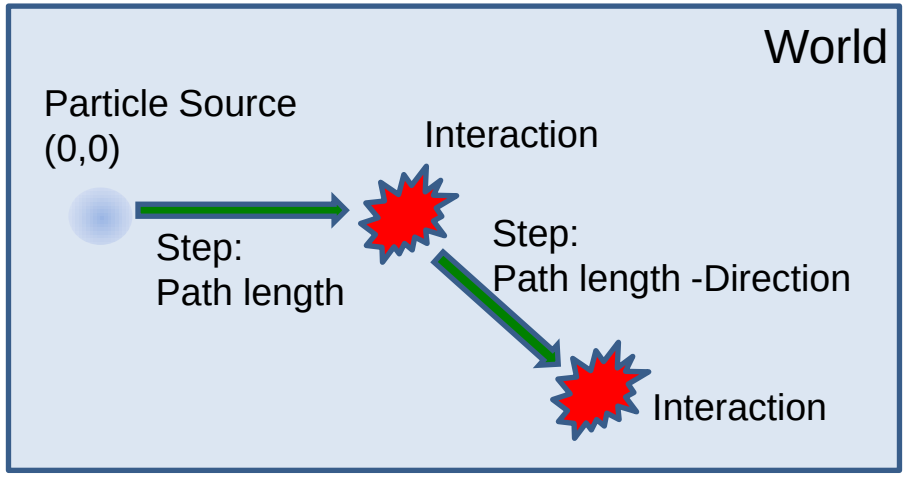
- Programming Exercise 4:

- Distribution of particle scattering angles



Monte Carlo Simulation Programming Exercise

- Programming Exercise 4:



1) The particle source for you is simply the starting position in space of your simulation

2) Then you will have to transport your particle to the next interaction point:
   - "Theory" shows that the particle path length is a stochastic process of exponential form $f(x) = e^{-x}$, x (0,∞)

3) You have then to decide what process will take place (remember you have two
   - Possibilities – Absorption or Elastic scattering). Again our "theory" tells us that there is an 80% probability for Elastic scattering and 20% for Absorption

   - If the particle is absorbed, it's "history" ends there and you proceed to the next particle from the source position and with direction X (step 1)

4) If the particle is scattered, you will have to decide which direction:
   - Past scattering "experiments" with this particle show that the distribution of angles can be fitted with: $g(x) = xe^{-x^2/a^2}$, x [0,180°] and α=25°
   - Resample next interaction point (step 2 and continue)

- Programming Exercise 4:

- Some technical hints:

1) Always check if the pdf's are normalized before sampling

2) The programming exercise is designed so that it can be easily solved using Inverse Transformation

3) At some point during the normalization of the scattering angle pdf g(x), you will find $e^{-180^2/625}$. Consider that equal to 0

4) Finally, the sampling of the angles will give you two possible solutions (square root). Use again random numbers to decide which one you will use

5) The world will be simply used in order to stop the simulation/tracking of a particle when it exits the world area

- Programming Exercise 4:
- Results:
  - Angular distribution of scattered particles
  - Path length distribution of particles
  - Ratio of absorption– scattering interactions
  - Do the above for a few different number of particles N
  - Plot a 2D tracking for a couple of particles
  - Assuming that you put detectors at x=1mm and x=5 mm,
    score number of particles that cross them and the position distribution
    of particles hitting the detector planes

- Important dates:
  - Deadline for handing in the programming exercises: 17th of January
  - Exam: 7th of February