Monte Carlo techniques

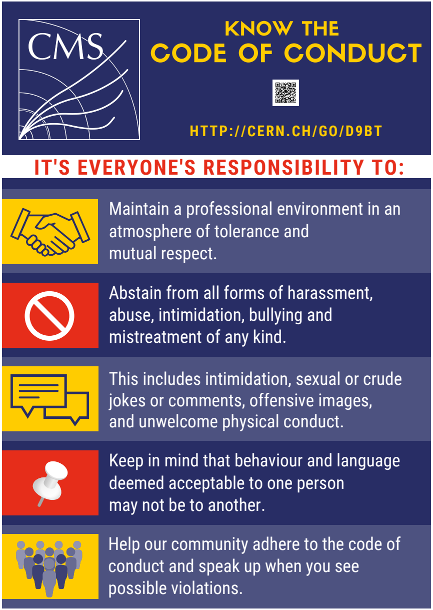
Paris Gianneios[1], Polidamas Georgios Kosmoglou Kioseoglou[2], Mikel Mendizabal Morentin[3]

CMS Physics Object & Data Analysis School
Hamburg, October 2023

[1]Université Libre de Bruxelles (BE), [2]University of Ioannina (GR), [3]DESY (GE)

Outline

Introduction



Introduction

✔ **In this presentation/exercises:**

- Monte Carlo methods.
- Basic principles for Monte Carlo event simulation.

Introduction

✔ In this presentation/exercises:

- Monte Carlo methods.
- Basic principles for Monte Carlo event simulation.

✘ Not in this presentation/exercises:

- How to run a given Monte Carlo event generator.
- Implementation of physics processes in Monte Carlo generators.

Introduction

✔ In this presentation/exercises:

- Monte Carlo methods.
- Basic principles for Monte Carlo event simulation.

✘ Not in this presentation/exercises:

- How to run a given Monte Carlo event generator.
- Implementation of physics processes in Monte Carlo generators.

Main sources for the exercises:

- [A] ▸ Hannes Jung, Terascale Summer School 2020
- [B] ▸ Patrick L.S. Connor, Monte Carlo School 2022

Introduction

✔ In this presentation/exercises:

- Monte Carlo methods.
- Basic principles for Monte Carlo event simulation.

✘ Not in this presentation/exercises:

- How to run a given Monte Carlo event generator.
- Implementation of physics processes in Monte Carlo generators.

Main sources for the exercises:

- [A] ▸ Hannes Jung, Terascale Summer School 2020
- [B] ▸ Patrick L.S. Connor, Monte Carlo School 2022

Further reading:

- [1] *Statistical Methods for Data Analysis in Particle Physics*, by Luca Lista.
- [2] *General-purpose event generators for LHC physics,* A. Buckley et al. ▸ 1101.2599
- [3] Clickable links and references on slides.

Introduction

All the material:

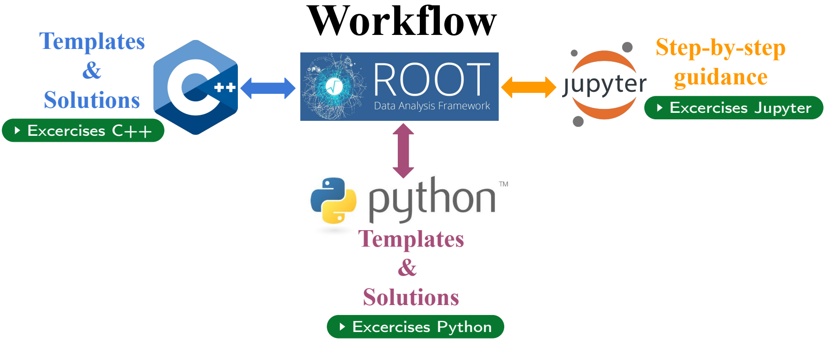
https://gitlab.cern.ch/cms-podas23/topical/mc-techniques

- We will alternate theory and hands-on exercises.
- Interruptions for questions are necessary ☺.
- All the exercises are based on ROOT (guidance for quick setup in the above link).

All the material:

https://gitlab.cern.ch/cms-podas23/topical/mc-techniques

- We will alternate theory and hands-on exercises.
- Interruptions for questions are necessary ☺.
- All the exercises are based on ROOT (guidance for quick setup in the above link).



Random Number Generators

Monte Carlo method

Numerical technique which makes use of sequences of **random numbers** and relies on probability statistics to solve a problem.

Random Number Generators

Monte Carlo method

Numerical technique which makes use of sequences of **random numbers** and relies on probability statistics to solve a problem.

Random Number Generators

1. **True Random Number Generator (TRNG)**: random numbers obtained from unpredictable processes e.g quantum physics, radioactive decay.
2. **Pseudo Random Number Generator (PRNG)**: random numbers generated on a computer according to some algorithm are not really random → *pseudo-random numbers* (deterministic, reproducible).

Random Number Generators

Monte Carlo method

Numerical technique which makes use of sequences of **random numbers** and relies on probability statistics to solve a problem.

Random Number Generators

1. **True Random Number Generator (TRNG)**: random numbers obtained from unpredictable processes e.g quantum physics, radioactive decay.
2. **Pseudo Random Number Generator (PRNG)**: random numbers generated on a computer according to some algorithm are not really random → *pseudo-random numbers* (deterministic, reproducible).

✅ Several **statistical** (serial test, gap test etc.) and **practical** (period, correlation etc.) criteria can be applied to assess the quality of random number generators (to be discussed in Exercise #1). Further reading:

- *The Art of Computer Programming (Vol. 2)*, by Donald E. Knuth.
- *A review of pseudorandom number generators*, F. James 10.1016/0010-4655(90)90032-V

Random Number Generators

PRNGs

- **Uniform**: sequences of uniformly distributed numbers ranging in a specific interval (addressed in Exercise #1).

Random Number Generators

PRNGs

- **Uniform**: sequences of uniformly distributed numbers ranging in a specific interval (addressed in Exercise #1).
- **Non-uniform**: starting from uniformly distributed random numbers and using various algorithms (addressed in Exercise #2).

Random Number Generators

PRNGs

- **Uniform**: sequences of uniformly distributed numbers ranging in a specific interval (addressed in Exercise #1).
- **Non-uniform**: starting from uniformly distributed random numbers and using various algorithms (addressed in Exercise #2).

List of PRNGs : Wikipedia PRNGs

Random Number Generators

PRNGs

- **Uniform**: sequences of uniformly distributed numbers ranging in a specific interval (addressed in Exercise #1).
- **Non-uniform**: starting from uniformly distributed random numbers and using various algorithms (addressed in Exercise #2).

List of PRNGs : Wikipedia PRNGs

Linear Congruential Generator (LCG)

- One of the oldest and best-known PRNGs commonly used in scientific applications for uniform random number generation.

Random Number Generators

PRNGs

- **Uniform**: sequences of uniformly distributed numbers ranging in a specific interval (addressed in Exercise #1).
- **Non-uniform**: starting from uniformly distributed random numbers and using various algorithms (addressed in Exercise #2).

List of PRNGs : ▸ Wikipedia PRNGs

Linear Congruential Generator (LCG)

- One of the oldest and best-known PRNGs commonly used in scientific applications for uniform random number generation.
- Random sequence ranging from 0 to m generated based on the recurrence relation:

$$\mathcal{I}_{i+1} = (\alpha \mathcal{I}_i + c) \bmod m \tag{1}$$

where $\mathcal{I}_0$ is called the *seed* of the sequence, α is called the *multiplier*, c is the *increment*, and m is the modulus.

Random Number Generators

- Implementations for different values of the parameters α, *c* and *m* are widely used in various languages/packages/libraries. See for example: ▸ Wikipedia LCGs

Random Number Generators

- Implementations for different values of the parameters α, c and m are widely used in various languages/packages/libraries. See for example: ▸ Wikipedia LCGs
- ▸ TRandom base class for the random number generation with ROOT.

Random Number Generators

- Implementations for different values of the parameters α, c and m are widely used in various languages/packages/libraries. See for example: ▸ Wikipedia LCGs
- ▸ TRandom base class for the random number generation with ROOT.

Exercise #1 - Linear Congruential Generator (LCG)

- Describe some desirable properties of a PRNG.
- Construct your own LCG based on equation:

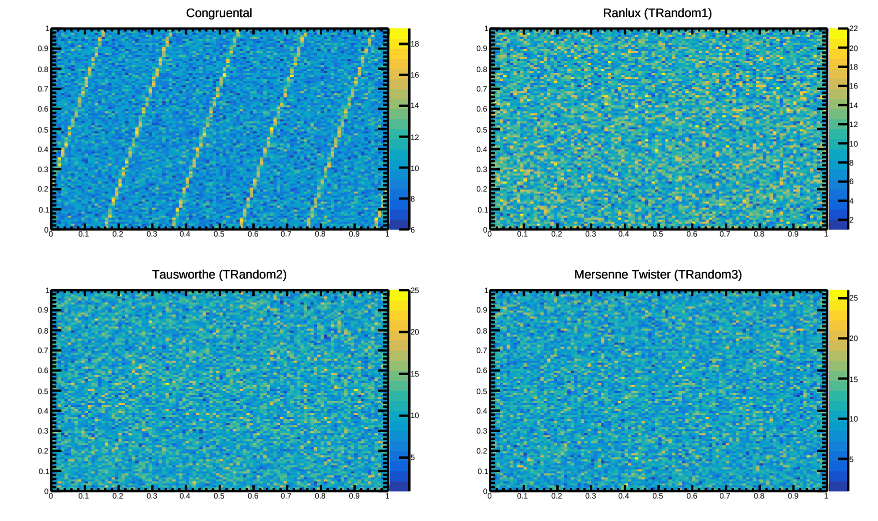
$$\mathcal{I}_{i+1} = (\alpha \mathcal{I}_i + c) \bmod m \tag{2}$$

with $\mathcal{I}_0 = 4711,\ \alpha = 205,\ c = 29573$ and $m = 139968$.

! Generated number should be normalized to m to get numbers from 0 to 1.

- Check the correlations of 2 generated random numbers from your LCG in a 2D histogram.
- Compare the above result with ROOT's built-in algorithms:
  - *RANLUX* implemented in TRandom1.
  - *Tausworthe* implemented in TRandom2.
  - *Mersenne Twister generator* implemented in TRandom3.

Exercise #1: Results and discussion



Random Number Generators

? Question: How to generate a sequence of random numbers distributed according to an arbitrary distribution (non-uniform)?

Random Number Generators

? Question: How to generate a sequence of random numbers distributed according to an arbitrary distribution (non-uniform)?

Approach 1: **Inversion of the cumulative distribution.**

Random Number Generators

? Question: How to generate a sequence of random numbers distributed according to an arbitrary distribution (non-uniform)?

Approach 1: **Inversion of the cumulative distribution.**

Basics: Probability Density Functions

- With x a continuous variable e.g. outcome from an experiment.
- Probability to observe a value of x within the interval $[x, x + dx]$: $f(x)dx$ where $f(x)$ is the **Probability Density Function (PDF)**.
- Normalization condition: $\int_{-\infty}^{+\infty} f(x)dx = 1$.

Random Number Generators

? Question: How to generate a sequence of random numbers distributed according to an arbitrary distribution (non-uniform)?

Approach 1: Inversion of the cumulative distribution.

Basics: Probability Density Functions

- With x a continuous variable e.g. outcome from an experiment.
- Probability to observe a value of x within the interval $[x, x + dx]$: $f(x)dx$ where $f(x)$ is the **Probability Density Function (PDF)**.
- Normalization condition: $\int_{-\infty}^{+\infty} f(x)dx = 1$.

Basics: Cumulative Distribution Functions

- Probability for the random variable to take on a value $\leq x$:

$$F(x) = \int_{-\infty}^{x} f(x')dx' \quad (3)$$

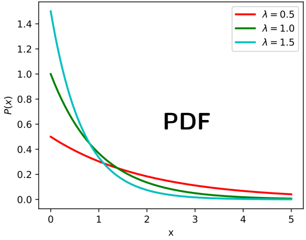
  where $F(x)$ is the **Cumulative Distribution Function (CDF)**.
- For a random number r uniformly distributed between 0 and 1, the transformed variable $x = F^{-1}(r)$ is distributed according to $f(x)$.

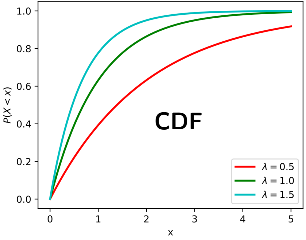
Example: Exp distribution

PDF: $f(x) = \lambda e^{-\lambda x}$ $(x \geq 0)$

CDF: $1 - e^{-\lambda x}$ $(x \geq 0)$

$x = -\frac{1}{\lambda}\log(1 - r)$





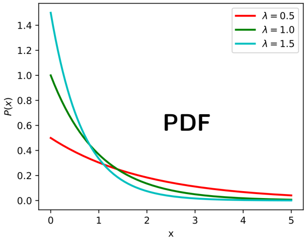
Random Number Generators

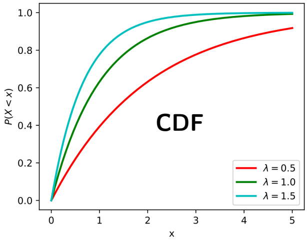
Example: Exp distribution

PDF: $f(x) = \lambda e^{-\lambda x}$ $(x \geq 0)$

CDF: $1 - e^{-\lambda x}$ $(x \geq 0)$

$x = -\frac{1}{\lambda}\log(1 - r)$



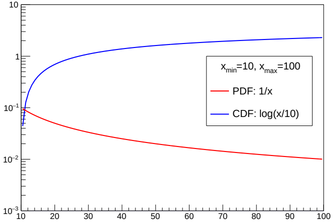


Example: $1/x$ distribution

PDF: $f(x) = 1/x$ (for x in $[x_{min}, x_{max}]$)

CDF: $\frac{1}{\log\frac{x_{max}}{x_{min}}}\log\frac{x}{x_{min}}$

$x = x_{min}\left(\frac{x_{max}}{x_{min}}\right)^r$



Random Number Generators

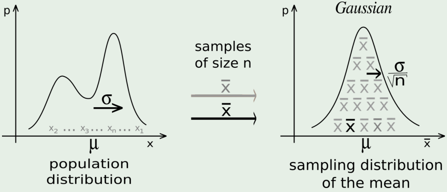
☞ **Approach 2:** **Gaussian generator using the Central Limit Theorem.**

Basics: Statistics definitions

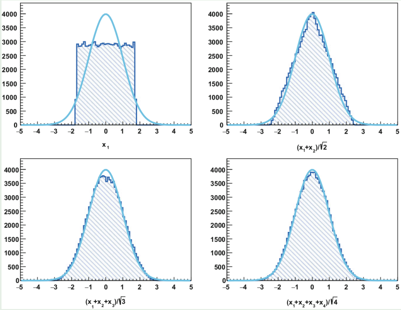
- For a function $f(x)$ whose PDF is $g(x)$:
  - Expectation value (average/population mean):
$$\mathbb{E}[f(x)] = \langle f(x) \rangle = \int f(x) g(x) dx \tag{4}$$
  - Variance:
$$\mathbb{V}[f(x)] = \mathbb{E}\left[(f(x) - \mathbb{E}[f(x)])^2\right] \tag{5}$$
  - Standard deviation:
$$\sigma_{f(x)} = \sqrt{\mathbb{V}[f(x)]} \tag{6}$$
- Examples for $f(x) = x$ and $g(x) =$
  - $1/(b-a)$ (Uniform):
    $\mathbb{E}[x] = (a+b)/2$, $\sigma_x = (b-a)/\sqrt{12}$
  - $\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$ (Gaussian):
    $\mathbb{E}[x] = \mu$, $\sigma_x = \sigma$

Basics: Central Limit Theorem

Suppose we have N independent random variables x_i, each distributed according to a PDF, having means μ_i and variances σ_i^2. The Central Limit Theorem states that in the limit of $N \to \infty$, the sum $\sum_i x_i$ becomes a Gaussian random variable with mean $\sum_i \mu_i$ and variance $\sum_i \sigma_i^2$, regardless of the underlying PDFs.
[*Statistical Data Analysis*, by Glen Cowan]



Taken from ▸ Wikipedia CLT



Taken from [1].

Random Number Generators

Exercise #2 - Gaussian random number generator

- Construct a Gaussian random number generator from a uniform random number generator.

! Hints

- For $R_i \in [0,1]$ following a uniform distribution and $R_n = \sum_i R_i$ (see Slide 12):

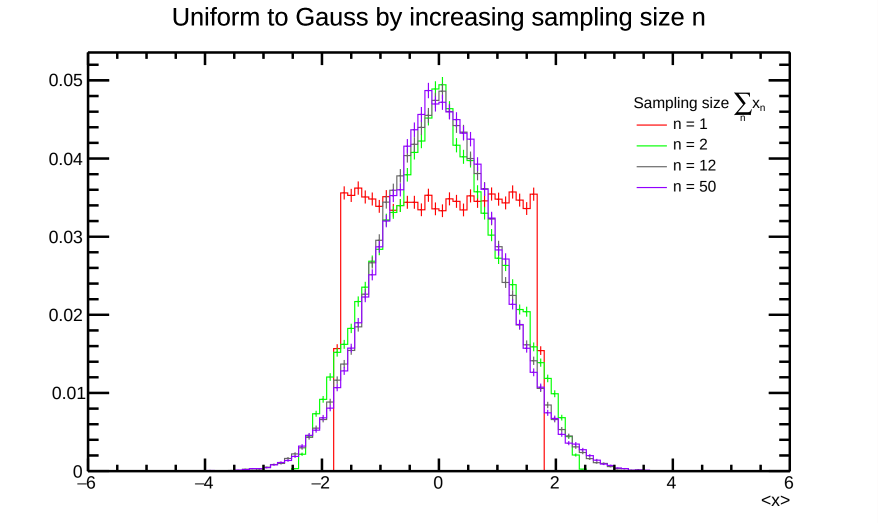
$$\mathbb{E}[R_1] = 1/2,\ \mathbb{V}[R_1] = 1/12$$
$$\mathbb{E}[R_n] = n/2,\ \mathbb{V}[R_n] = n/12$$

- For a Normal Gaussian distribution ($\mu = 0,\ \sigma = 1$) use:

$$\mathcal{N}(0,1) \to \frac{\sum_i x_i - \sum_i \mu_i}{\sqrt{\sum_i \sigma_i^2}} \to \frac{R_n - n/2}{\sqrt{n/12}}$$

- First try with $n = 12$: $\mathcal{N}(0,1) \to R_{12} - 6$
- Then you can try for different values of n. Plot all histograms in the same canvas.

Exercise #2: Results and discussion



Monte Carlo Integration

Why to choose Monte Carlo for integration?

- The Monte Carlo estimation accuracy improves as $1/\sqrt{N}$ (addressed in next slides), **irrespective of the dimension D**.

Method	Error for 1-D	Error for N-D
Trapezoidal	n^{-2}	$n^{-2/D}$
Simpson	n^{-4}	$n^{-4/D}$
Gauss	n^{-2m+1}	$n^{(-2m+1)/D}$
Monte Carlo	$n^{-1/2}$	$n^{-1/2}$

- Cross section predictions for *pp* collisions include the phase space integration:

$$\int d\Phi_n \quad \text{with} \quad d\Phi_n = \prod_{i=1}^{n} \frac{d^3 p_i}{(2\pi)^3 2E_i} \cdot (2\pi)^4 \delta^{(4)}(p_a + p_b - \sum_{i=1}^{n} p_i) \tag{7}$$

  - Integral of dimension: $3n - 4$ (n final-state particles).
  - Three components of momentum per produced particle, minus four constraints of overall energy-momentum conservation.

Further reading in [2].

Monte Carlo Integration

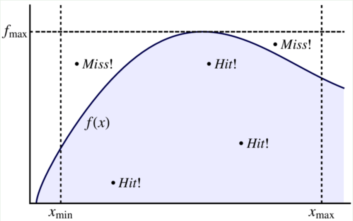
Hit-or-miss Monte Carlo

General method be used for:

- Random number generation according to some PDF.
- Numerical calculation of an integral.

→ **Method:** Calculation of area below $f(x)$

1. Generate random numbers x,y, uniformly distributed within $[x_{min}, x_{max}]$ and $[0, f_{max}]$ respectively.
2. If $y \leq f(x)$ (the point is under the curve) count x as hit $\hat{n}$.
3. Stop after N trials.



Taken from 10.13140/RG.2.2.24616.47367

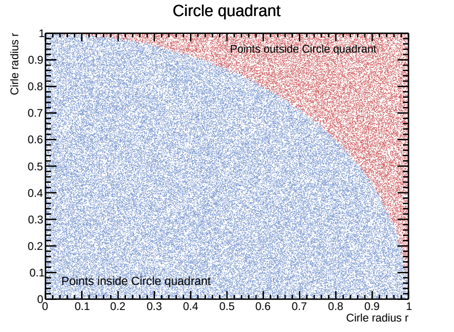
→ The estimation for the area is:

$$I = \int_{x_{min}}^{x_{max}} f(x)dx \approx (x_{max} - x_{min}) \times \frac{\hat{n}}{N} f_{max} \quad (8)$$

Exercise #3 - Approximate the value of π

- Use the Hit-or-miss method (as described above) to estimate the value of π, based on equation:

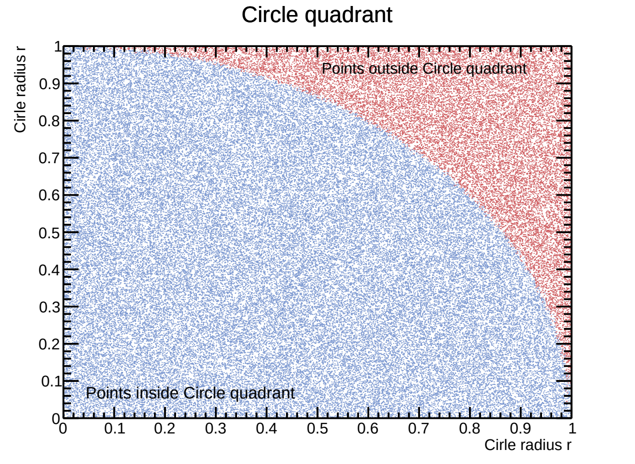
$$I = (x_{max} - x_{min}) \times \frac{\hat{n}}{N} f_{max} \qquad (9)$$



! Hints

- Consider a quadrant with $r = 1$.
- Generate two random numbers.
- You already know the integral value :)

Exercise #3: Results and discussion

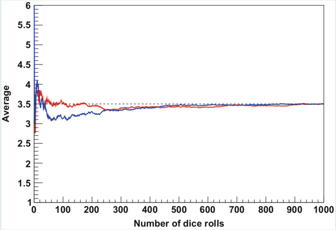


Monte Carlo Integration

The Law of Large Numbers (LLN)

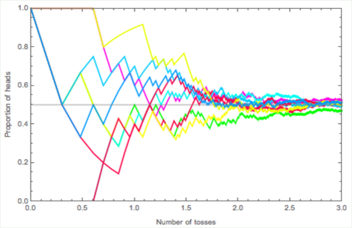
- Suppose we repeat an experiment N times and produce outcomes $x_1, \ldots, x_N$, where $x_1, \ldots, x_N$ are independent random variables with the same underlying distribution (same population mean μ and standard deviation σ).
- The average $\hat{x}_N$ of all results (sample mean) is: $\hat{x}_N = \frac{x_1, \ldots, x_N}{N} = \frac{1}{N}\sum_{i=1}^{N} x_i$.
- **Law of Large Numbers:** As N increases, the sample mean $\hat{x}_N$ converges to the population mean μ (expected value): $N \to \infty \Rightarrow \hat{x}_N \to \mu$.

Example 1. Dice rolls ($\mu = 3.5$) for $N = 1000$ (2 repetitions).



Taken from [1]

Example 2. Coin flip ($\mu = 0.5$) for $N = 1000$ (9 repetitions).



Taken from ▸ Wolfram Demonstrations

Monte Carlo Integration

? Question: The LLN has a broad validity ranges and various applications, but why do we mention it here?

Monte Carlo Integration

? Question: The LLN has a broad validity ranges and various applications, but why do we mention it here?

Reminder (Slide 12): expectation value (population mean):

$$\mathbb{E}[f(x)] = \langle f(x) \rangle = \int f(x) g(x) dx \tag{10}$$

Monte Carlo Integration

? Question: The LLN has a broad validity ranges and various applications, but why do we mention it here?

Reminder (Slide 12): expectation value (population mean):

$$\mathbb{E}[f(x)] = \langle f(x) \rangle = \int f(x) g(x) dx \tag{10}$$

For $g(x)$ the uniform distribution $1/(b-a)$ and based on the LLN, for $N \to \infty$:

$$\frac{1}{N} \sum_{i=1}^{N} f(x_i) \to \mathbb{E}[f(x)] = \frac{1}{b-a} \int_a^b f(x) dx \tag{11}$$

? Question: The LLN has a broad validity ranges and various applications, but why do we mention it here?

Reminder (Slide 12): expectation value (population mean):

$$\mathbb{E}[f(x)] = \langle f(x) \rangle = \int f(x)g(x)dx \tag{10}$$

For $g(x)$ the uniform distribution $1/(b-a)$ and based on the LLN, for $N \to \infty$:

$$\frac{1}{N}\sum_{i=1}^{N} f(x_i) \to \mathbb{E}[f(x)] = \frac{1}{b-a}\int_a^b f(x)dx \tag{11}$$

Therefore, using PRNGs we can estimate the value of an integral $I = \int_a^b f(x)dx$ as:

$$I \approx I_{MC} = (b-a)\frac{1}{N}\sum_{i=1}^{N} f(x_i) \tag{12}$$

? Question: The LLN has a broad validity ranges and various applications, but why do we mention it here?

Reminder (Slide 12): expectation value (population mean):

$$\mathbb{E}[f(x)] = \langle f(x) \rangle = \int f(x)g(x)dx \tag{10}$$

For $g(x)$ the uniform distribution $1/(b-a)$ and based on the LLN, for $N \to \infty$:

$$\frac{1}{N}\sum_{i=1}^{N} f(x_i) \to \mathbb{E}[f(x)] = \frac{1}{b-a}\int_a^b f(x)dx \tag{11}$$

Therefore, using PRNGs we can estimate the value of an integral $I = \int_a^b f(x)dx$ as:

$$I \approx I_{MC} = (b-a)\frac{1}{N}\sum_{i=1}^{N} f(x_i) \tag{12}$$

The error in this estimation depends on N and on the variance of f (proof in [A]):

$$\sigma_{MC}^2 = \mathbb{V}[I_{MC}] = \frac{1}{N}\left(\frac{(b-a)^2}{N}\sum_{i=1}^{N} f_i^2 - I_{MC}^2\right) \tag{13}$$

Monte Carlo Integration

Exercise #4 - MC integration

- Write a program which estimates (with Monte Carlo method) the integral:

$$\int_3^8 \frac{1}{2x+1} dx \tag{14}$$

  and the corresponding error in the estimation.
- Use different values of N and compare with the nominal integral value which is: $I = 0.443652$ (you can calculate it analytically or using the Integral member function of TF1 class in ROOT).

Exercise #4: Results and discussion

- For $N = 1000$: $I_{MC} = 0.445135 \pm \ 0.003673256$
- For $N = 10000$: $I_{MC} = 0.445692 \pm \ 0.001153906$
- For $N = 100000$: $I_{MC} = 0.443575 \pm \ 0.000364692$

Monte Carlo Integration

Importance sampling

? Question: What happens if the integrable function $f(x)$ is very peaked?

Monte Carlo Integration

Importance sampling

? Question: What happens if the integrable function $f(x)$ is very peaked?
$\rightarrow$ **Answer:** The error in the estimation of the integral, using the method described above with uniform random generators, is large. (Why?)

Monte Carlo Integration

Importance sampling

? Question: What happens if the integrable function $f(x)$ is very peaked?
$\rightarrow$ **Answer:** The error in the estimation of the integral, using the method described above with uniform random generators, is large. (Why?)

☞ One approach to improve the accuracy is the **importance sampling**:

- Use an approximate function $g(x)$ such that:

$$I = \int_a^b f(x)dx = \int_a^b \frac{f(x)}{g(x)} g(x)dx = \mathbb{E}\left[\frac{f(x)}{g(x)}\right] \tag{15}$$

  which means that the integral corresponds to the expectation value of f/g, if the values of x are distributed according to the PDF $g(x)$.

Monte Carlo Integration

Importance sampling

? Question: What happens if the integrable function $f(x)$ is very peaked?
$\rightarrow$ **Answer:** The error in the estimation of the integral, using the method described above with uniform random generators, is large. (Why?)

One approach to improve the accuracy is the **importance sampling**:

- Use an approximate function $g(x)$ such that:

$$I = \int_a^b f(x)dx = \int_a^b \frac{f(x)}{g(x)} g(x)dx = \mathbb{E}\left[\frac{f(x)}{g(x)}\right] \quad (15)$$

  which means that the integral corresponds to the expectation value of f/g, if the values of x are distributed according to the PDF $g(x)$.

- I_{MC} and the corresponding error σ_{MC} are now given by:

$$I_{MC} = \frac{1}{N}\sum \frac{f(x_i)}{g(x_i)}, \qquad \sigma^2_{MC} = \frac{1}{N}\left(\frac{1}{N}\sum_{i=1}^{N}\left(\frac{f_i}{g_i}\right)^2 - I^2_{MC}\right) \quad (16)$$

Monte Carlo Integration

Exercise #5 - Importance sampling

Consider the function $f(x) = \frac{(1-x)^5}{x}$.

- Plot it (TF1) to inspect its shape.
- Estimate the integral $\int_{x_{min}}^{1} f(x)dx$ and the error with the same method as in Exercise #4, for $x_{min} = 0.0001$.
- Use importance sampling to improve the result:
  - Approximate $f(x)$ with $g(x) = \frac{1}{x}\frac{1}{log\frac{x_{max}}{x_{min}}}$, where $\frac{1}{log\frac{x_{max}}{x_{min}}}$ is just a normalization factor (you can also plot $g(x)$).
  - The integral estimation is now written as: $I_{MC} = \frac{log\left(\frac{x_{max}}{x_{min}}\right)}{N}\sum\frac{f(x_i)}{\frac{1}{x_i}}$.

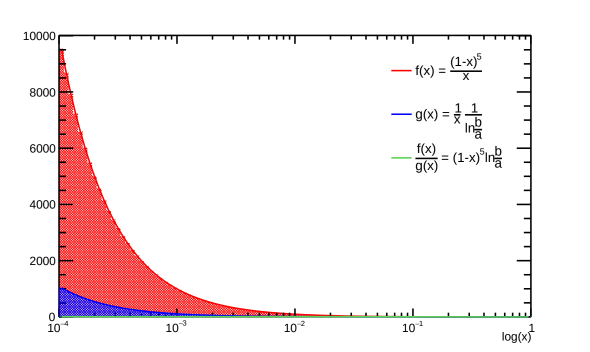
! Hint

The random values x (distributed according to $g(x)$) can be generated from the uniformly distributed numbers r with*: $x = x_{min}\left(\frac{x_{max}}{x_{min}}\right)^r$.

* Calculated with the method described in Slide 9.

Exercise #5: Results and discussion

- $I = 6.927507$ (Analytically or with Integral function).
- $I_{MC} = 6.497323 \pm\ 0.255584339$ (without importance sampling)
- $I_{MC} = 6.935575 \pm\ 0.009913379$ (with importance sampling)



Basics: Parton Distribution Functions (1/2)

Cross section predictions for pp collisions:

$$d\sigma_{(pp\to X)} = \sum_{i,j} \int dx dx' f_{i/p}(x, \mu_f) \cdot f_{j/p}(x', \mu_f) \times \hat{d\sigma}_{(ij\to X)}(x, x', \mu_f, \mu_r, \alpha_s(\mu_r)) \quad (17)$$

where $\hat{d\sigma}_{(ij\to X)}$ is calculated using perturbation theory (includes PS integration: Slide 16) and $f_{i/p}$, $f_{j/p}$ are the **Parton Distribution Functions (PDFs)** which:

- quantify the probability to find a parton i/j with longitudinal momentum fraction x/x' within the proton h at a resolution characterised by the factorisation scale μ_f.
- cannot been calculated perturbatively from first principles.
- can be determined in different processes and at different scales μ_f^2 (universal).

The evolution of PDFs with μ_f^2 can be calculated with a perturbative treatment using the DGLAP evolution equations (Further reading in ▸ The Review of Particle Physics):

$$\mu_f^2 \frac{\partial f_{i/p}(x, \mu_f^2)}{\partial \mu_f^2} = \sum_{j=\{q,\bar{q},g\}} \int_x^1 \frac{dz}{z} \frac{\alpha_s}{2\pi} P_{ij}(z) f_{j/p}(x/z, \mu_f^2) \quad (18)$$

Basics: Parton Distribution Functions (2/2)

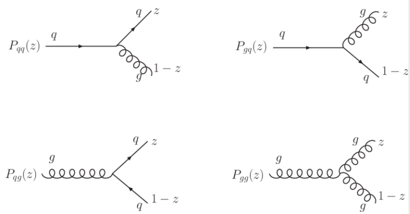
- Once we determine the parton densities at a specific scale, we can predict the parton densities at any scale using the DGLAP equations.
  Online library for PDF evolution: APFEL.
- z parton's momentum ratio before and after splitting.
- P_{ij} are the (regularised) splitting functions (known up to N^{3}LO) that describe the probability of a given parton splitting into two others $i \to jk$ (k is fixed by ij).
- At LO of α_s:

$$P_{qq} = \frac{4}{3}\left(\frac{1+z^2}{1-z}\right) \quad (19)$$

$$P_{gq} = \frac{4}{3}\left(\frac{1+(1-z)^2}{z}\right) \quad (20)$$

$$P_{qg} = \frac{1}{2}\left(z^2+(1-z)^2\right) \quad (21)$$

$$P_{gg} = 6\left(\frac{1-z}{z}+\frac{z}{1-z}+z(1-z)\right) \quad (22)$$



Physics Application

Basics: Solving DGLAP equations (1/2)

- There are several methods to solve integro-differential equations exist either analytically or numerically.
- Today: **Monte Carlo method from iterative procedure**.
- Introduce a new quantity **Sudakov form factor**:

$$\Delta_s(t) = exp\left(-\int_x^{z_{max}} dz \int_{t_0}^{t} \frac{\alpha_s}{2\pi}\frac{dt'}{t'}\hat{P}(z)\right) \tag{23}$$

  where $\hat{P}$ are the unregularised splitting functions.
- Δ_s decribes probability of evolving from scale t_0 to scale t without any splitting: a given particle *does not* to radiate any secondary particle.
- The DGLAP equation (eq. 18) now becomes:

$$t\frac{\partial}{\partial}\frac{f(x,t)}{\Delta_s} = \int_x^1 \frac{dz}{z}\frac{1}{\Delta_s}\frac{\alpha_s}{2\pi}P(z)f(x/z,t) \tag{24}$$

Further reading: *QCD and Collider Physics*, by R.K. Ellis, W.J. Stirling and B.R. Webber.

Basics: Solving DGLAP equations (2/2)

- The complete solution for $f(x,t)$ using an iterative procedure is:

$$\lim_{n\to\infty} \sum_n \frac{1}{n!} \log^n \left(\frac{t}{t_0}\right) \left(\int \frac{dz}{z} \hat{P}(z)\right)^n \otimes \Delta_s(t) f(x/z, t_0) \tag{25}$$

- Starting function (paths without splittings between scales t_0 and t):

$$f_0(x,t) = f(x,t_0)\Delta_s(t) \tag{26}$$

- One iteration (paths with one splitting between scales t_0 and t):

$$f_1(x,t) = f(x,t_0)\Delta_s(t) + \frac{\alpha_s}{2\pi} \int_{t_0}^{t} \frac{dt'}{t'} \frac{\Delta(t)}{\Delta(t')} \int_x^1 \frac{dz}{z} \Delta_s(t') \hat{P}(z) f(x/z, t_0) \tag{27}$$

Further details in [A].

Physics Application

Exercise #6 - Sudakov form factor

Using Monte Carlo integration method as in Exercise #4, calculate and plot the Sudakov form factor as a function of the scale t:

$$\log\Delta_s = -\int_{t_1}^{t_2} \frac{dt}{t} \int_{z_{min}}^{z_{max}} dz \frac{\alpha_s}{2\pi} P(z) \tag{28}$$

using:

- Different values for the scale: $t \in [1, 500]$ GeV^2.
- Limits for the integral: $z_{min} = 0.01$ and $z_{max} = 0.99$.
- $P_{gg} = 6\left(\frac{1-z}{z} + \frac{z}{1-z} + z(1-z)\right)$ and $P_{qq} = \frac{4}{3}\left(\frac{1+z^2}{1-z}\right)$
- The one-loop level solution for the α_s running: $\alpha_s(Q) = \frac{1}{b_0 \cdot \ln\left(Q^2/\Lambda^2_{QCD}\right)}$, with $b_0 = \frac{33-2n_f}{12\pi}$ $(n_f = 3)$ and $\Lambda^2_{QCD} = 0.2$ GeV^2.

Exercise #6: Results and discussion