MONTE CARLO METHODS

FOR HIGH ENERGY PHYSICS

by S. Jadach

Institute of Nucl. Physics, Kraków, Poland

Lecture 1

History, basic concepts, random numbers

Slides, program sources: http://home.cern.ch/jadach

Numerical examples exploit ROOT package: http://root.cern.ch/

Outline:

1. Role of MC in H.E. physics
2. Probability distribution, events, integrals
3. Examples of n-dim. distributions
4. Use of Dirac δ-function
5. MC simulation versus MC integration
6. Simple example of MC simulation
7. Simple example of MC integration
8. MC error and convergence
9. What is histogram? Errors in histogram
10. Central limit theorem, introduction
11. Central limit theorem, proof and illustrations
12. Random number generators

Golden references:

- Buffon, Kelvin and many other precursors, but...
- Modern Monte Carlo is a child of Los Alamos atomic bomb project (neutron transport), with many contributors (Fermi, Feynman, Ulam, Von Neuman...)
- N. Metropolis and S. Ulam, “The Monte Carlo method”, J. Amer. Stat. Assoc (1949) 44, **247**, 337-341.
- MC in Particle physics: I Kopylov, JETP **35** (1958) 1426, with acknowledgements to Podgoretski and Danysz.
- G.R. Lynch, FAKE, UCRL-10335 (1962).
- F. James, FOWL, CERLNLIB W-505, 1966-70.
- C.J. Everet and E.D. Cashwell, “A Monte Carlo Sampler”, Los Alamos internal report, LA-5081-MS, October 1972.
- G. Peter Lepage, “A new Algorithm for adaptive Multidimensional Integration”, Journ. of Comp. Phys. **27** (1978) 192.

The role of MC in H.E. physics

It has increases significantly because:

- HE experimental detectors see final state events in increasingly fine detail,
- Precise theoretical prediction has to be provided for more-than-two particle final states.

The significant shift of the MC role was between LEP1 and LEP2 with the advent MC program for high precision $< 0.5\%$ SM prediction for $e^- e^+ \rightarrow WW \rightarrow 4f$ process.

Even data fitting at LEP2 (mass of W) is done with help of M.C. programs!

Eerly precursors were MC programs for precise $< 0.1\%$ low angle Bhabha (luminosity) at LEP1.

MC programs will be standard indespensable tool for precision theory predictions in future LHC and LC experiments.

Another important role of MC programs was, and will be, to help taking out of experimental data all effects due to detector imperfections. Here, less precision but complete phase space coverage is important.

Notation, Definitions, Terminology

Probability distribution Physicist definition: Real positive integrable function $p(x) = p(x_1, x_2, ..., x_n)$ defined within domain $\Omega \in \{R^n\}$ of n-dim. real space.

$$\int_\Omega \prod_{i=1}^n dx_i \; p(x) = \int_\Omega p(x) dV = 1, \quad \rho(x) \geq 0$$

Event: Single point $x = (x_1, x_2, ..., x_n)$ in the domain Ω.

Physicist Extensions: Probability $p(x)$ is not necessarily continuous, may contain step function: $\theta(z) = 1$ if $z > 0$ else $\theta(z) = 0$.

Dirac $\delta(z)$ function is admitted as a component of $p(x)$!!!

$$\delta(z) \equiv \lim_{\epsilon \to 0} \frac{1}{\epsilon\sqrt{2\pi}} \exp\left(-\frac{z^2}{2\sigma^2}\right), \quad \int \delta(z) f(z) dz = f(0), \quad \int \delta(z) f(z) dz = f(0).$$

Density distribution and its Integral: Integrable function $\rho(x)$ defined as $p(x)$, but not necessarily positive and usually with "unknown" normalization,

$$R = \int_\Omega \rho(x) dV \neq 1$$ (ρ may contain $\delta(z)$ and $\theta(z)$).

Examples of n-Dim distributions

Uniform Distributions

Riemann hyper-cube $p(x) = \rho(x) = \prod_{i=1}^{n} \theta(x_i)\theta(1 - x_i)$

n-Dim Simplex $p(x) = n!\theta(x_1) \; \prod_{i=2}^{n} \theta(x_i - x_{i-1})\; \theta(1 - x_n)$

n-Dim Sphere $\rho(x) = \delta\big(1 - \sqrt{\sum_{i=1}^{n} x_i^2}\;\big), p(x) = \frac{1}{S_n}\rho(x), S_n = \frac{2\,\pi^{n/2}}{\Gamma(n/2)}$

Non-Uniform Distributions:

n-Dim Gaussian $p(x) = \frac{1}{(\sigma\sqrt{2\pi})^n} \exp\big(-\frac{1}{2\sigma^2}(\sum_{i=1}^{n} x_i^2)\big)$

n-Particle Lorentz-invariant phase space (Lips)

$$\rho(p_1, ..., p_n) = \delta^{(4)}(P - \sum_{i=1}^{n} p_i) \; \prod_{i=1}^{n} \delta(p_i^2 - m_i^2)\theta(p_i^0)$$

Note 1: $V_n = \int \rho(p_1, ..., p_n) \prod_{i=1}^{n} d^4 p_i,$ analytically is known only for $n = 2, 3$.

Note 2: The above “Fermi Lips” can be also regarded as an uniform distribution (see sphere).

Examples of distributions

Discrete Distributions

Having admitted δ-functions all discrete distribution can be regarded as particular case of singular distrinution in R^n. Poisson distribution can be written as: $p(x) = e^{\lambda} \sum_{n=0}^{\infty} \frac{\lambda^n}{n!} \delta(x-n)$. Keep it in mind, but in practice using integers to construct space of events is usually more convenient (think about permutations).

Dimension of space as another variable

Probability distribution can be spanned over subspaces of different dimensions. In this case dimension n enters every event record $x = \{n; x_1, x_2, ..., x_n\}$.

Example: $p(n, \vec{x}) = e^{-\lambda} \theta(x_1) \prod_{i=2}^{n} \theta(x_i - x_{i-1})\, \theta(\lambda - x_n), \quad \lambda > 0.$

Normalization: $\sum_{n=0}^{\infty} \int d^n x\; p(n, \vec{x}) \equiv 1.$

The above is infi nite-dimensional probability distribution – quite common in practice.

Small exercise on the use of δ-function

We shall calculate area of the surface of the n-dim sphere S_n:

$$1 = \frac{1}{(\sqrt{2\pi})^n} \int d^n x\; e^{-\frac{1}{2}(\sum_{i=1}^n x_i^2)}$$
$$= \frac{1}{(\sqrt{2\pi})^n} \int dr \int d^n x\; e^{-\frac{1}{2}(\sum_{i=1}^n x_i^2)} \delta(r - \sqrt{\textstyle\sum_{i=1}^n x_i^2})$$

Change variables $x^i = r y^i$ and use $\delta(af(z)) = (1/a)\delta(f(z))$:

$$1 = \frac{1}{(\sqrt{2\pi})^n} \int dr\; r^n \frac{1}{r}\; e^{-\frac{r^2}{2}} \int d^n y\; \delta(1 - \sqrt{\textstyle\sum_{i=1}^n y_i^2})$$
$$= S_n\; (2\pi)^{-n/2} \int dr\; r^{n-1}\; e^{-\frac{r^2}{2}}$$
$$= S_n\; (2\pi)^{-n/2} 2^{n/2-1} \int du\; u^{n/2-1}\; e^{-u}$$
$$= S_n\; 2^{-1} \pi^{-n/2}\; \Gamma(n/2).$$

Hence $\boxed{S_n = \frac{2\;\pi^{n/2}}{\Gamma(n/2)}}$.

Note: In the above calculation we have projected n-dim. Gaussian dist. on $(n-1)$-dim. sphere radially, something which is also often done in the MC generation.

MC Simulation versus MC Integration

Monte Carlo simulation

Large number (list) of the events $x = (x_1, x_2, ..., x_n)$ is "fabricated" randomly, independently, exactly according to predefined probability distribution $p(x)$. These events are stored or used as an input for further work.

For example to calculate distribution of an "observable" $G(x_1, x_2, ..., x_n)$: $\frac{dp}{dg} = \int d^n x\, \delta\big(g - G(x_1, x_2, ..., x_n)\big)$ with help of a "histogram".

Monte Carlo integration

Here that main aim is to calculate single number, an integral $R = \int_\Omega \rho(x) d^n x$. It is done using events $x_I, I = 1, 2, ...N$ generated according to $p(x)$.

$R = \lim_{N\to\infty} \frac{1}{N} \sum_{I=1}^{N} \frac{\rho(x_I)}{p(x_I)}$. Ratio $w(x) = \frac{\rho(x)}{p(x)}$ is called weight of event x.

For finite N integral estimate R has statistical error $\delta R = \frac{\sigma}{\sqrt{N}} = \frac{\sqrt{\langle\ (w - \langle w\rangle)^2\ \rangle}}{\sqrt{N}}$ according to central limit theorem, see next slides.

MC Simulation versus MC Integration

- MC Simulation is much more difficult than MC Integration.
- Computer realization of MC Simulation usually contains MC Integration.
- MC Integration usually provides "variable-weight" events, while MC Simulation is synonymous with "weight-one" events.
- Efficiency of MC Integration is measured by the ratio $\sigma/\langle w \rangle$. The smaller variance σ the better.
- Variance-reduction or importance-sampling in MC Integration is an effort to get $\rho(x) \longrightarrow const \times p(x)$, i.e., $p(x)$ should have similar peaks and singularities as $\rho(x)$.
- Weight $w = \rho/p$ should never include δ-functions.
- We exclude from consideration the use of correlated events.

Simple example of MC Simulation

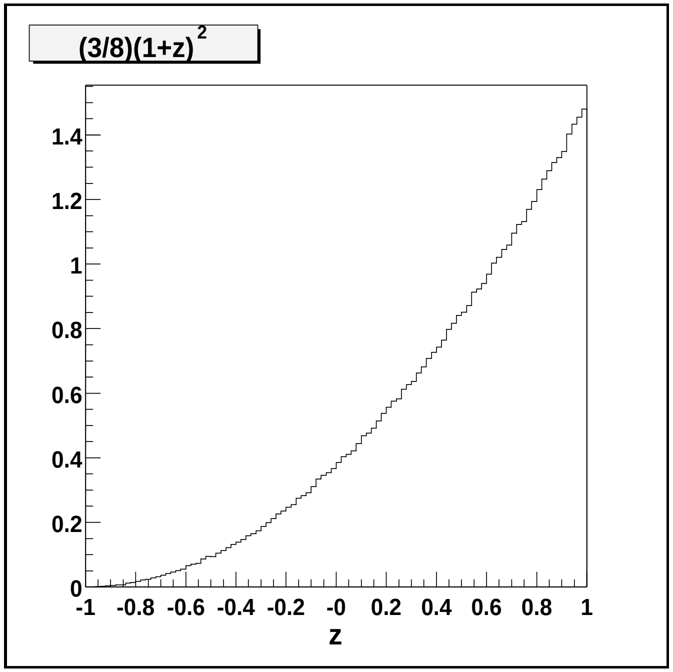
$$p(z) = \frac{3}{8}(1+z)^2\theta(1-z)\theta(1+z)$$

Define cumulative function $r(z) = \int_{-1}^{z} p(x)dx$ and use it as a new integration variable:

$$\int p(z)dz = \int_0^1 dr\ \frac{dx}{dr}p(x(r)) = \int_0^1 dr\ \frac{1}{\frac{dr}{dx}}p(x(u)) = \int_0^1 dr\ 1.$$

Variable $r \in [0,1]$ has uniform distribution, readily available from any r.n. gen.

For our $p(z)$ we have $r(z) = \frac{1}{8}(1+z)^3$ and $z = -1 + (8r)^{1/3}$.




```
Double_t  RndCthe(TRandom *RNgen ){
// p(z)=(3/8)(1+z)^2 distribution
  Double_t r = RNgen->Rndm(0);
  Double_t z = -1.0+pow( 8.0*r, 1.0/3.0 );
  return z;
}
```

Simple example of MC integration

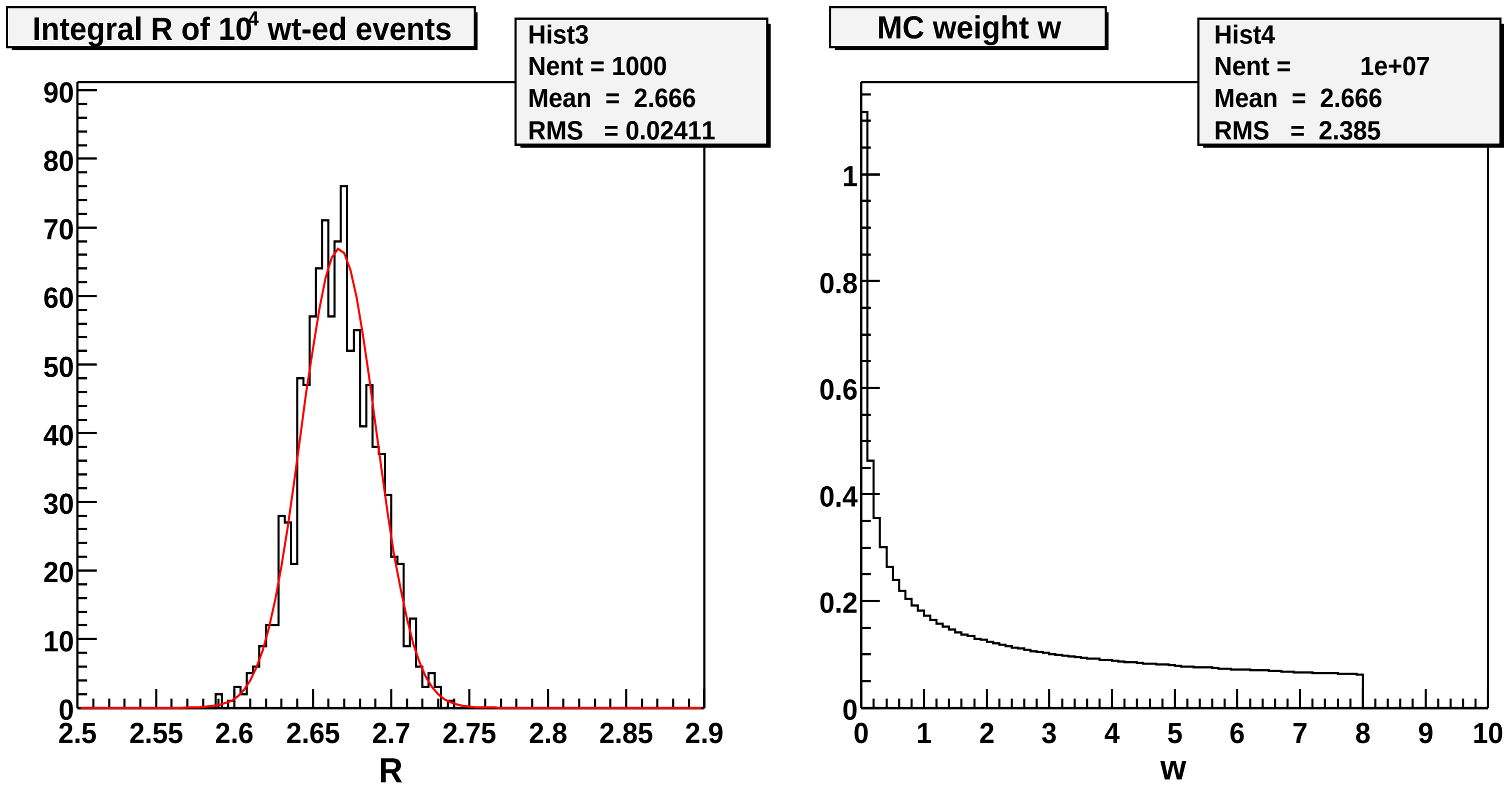
$$\rho(z) = (1+z)^2\theta(1-z)\theta(1+z)$$

We know $R = \int \rho(z)dz = 8/3 = 2.6666\ldots$. Nevertheless, we calculate it with help of the MC method.

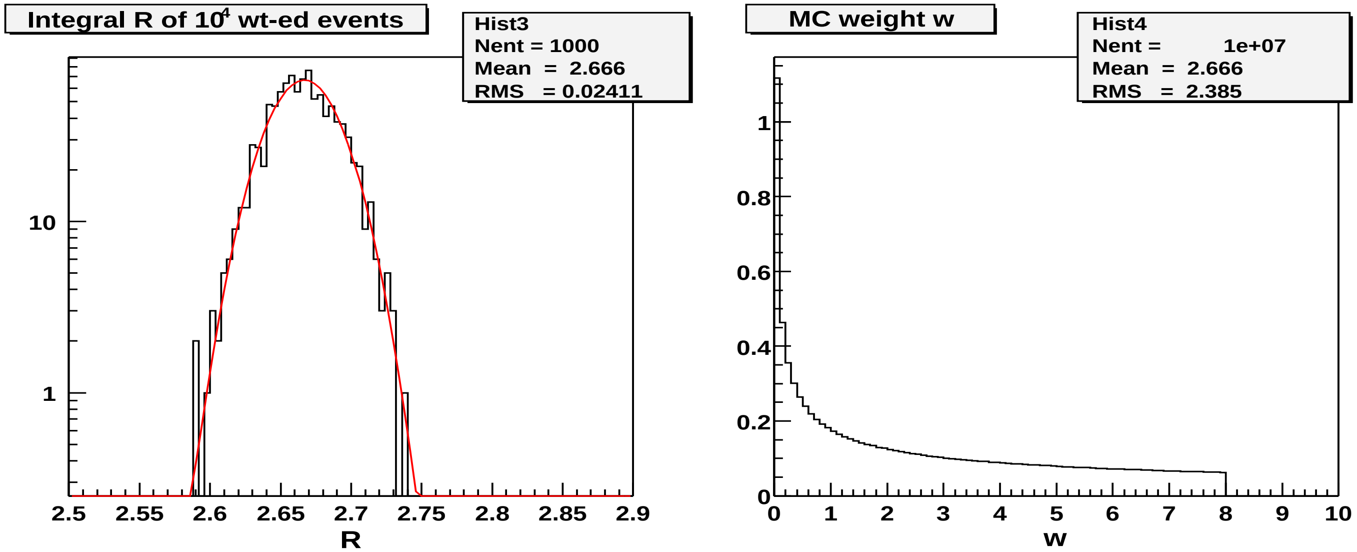
Generate z uniformly within $(-1,+1)$. Define MC weight $w = \rho(z)$.

$R \simeq \langle w \rangle = \frac{1}{N}\sum_{I=1}^{N} w(z_I)$.

We generate $N = 10^4$ MC events. Repeat the calculation 1000 times!



Simple example of MC integration, Cont.



- Variance of the distribution of the MC weight w is $\sigma = 2.835$
- Result of 10^3 independent integrations $R = \langle w \rangle$, each for for $N = 10^4$ events, is spread randomly according to normal (Gaussian) distribution.
- Distribution of R is exactly as expected form central limit theorem (see next slides), with the variance $\sigma_R = 2.835/\sqrt{10^4} = 0.024$.
- Standard deviation $\sigma_R = \sigma/N^{1/2}$ is therefore used as an error estimate.

Simple example of MC integration, Cont.

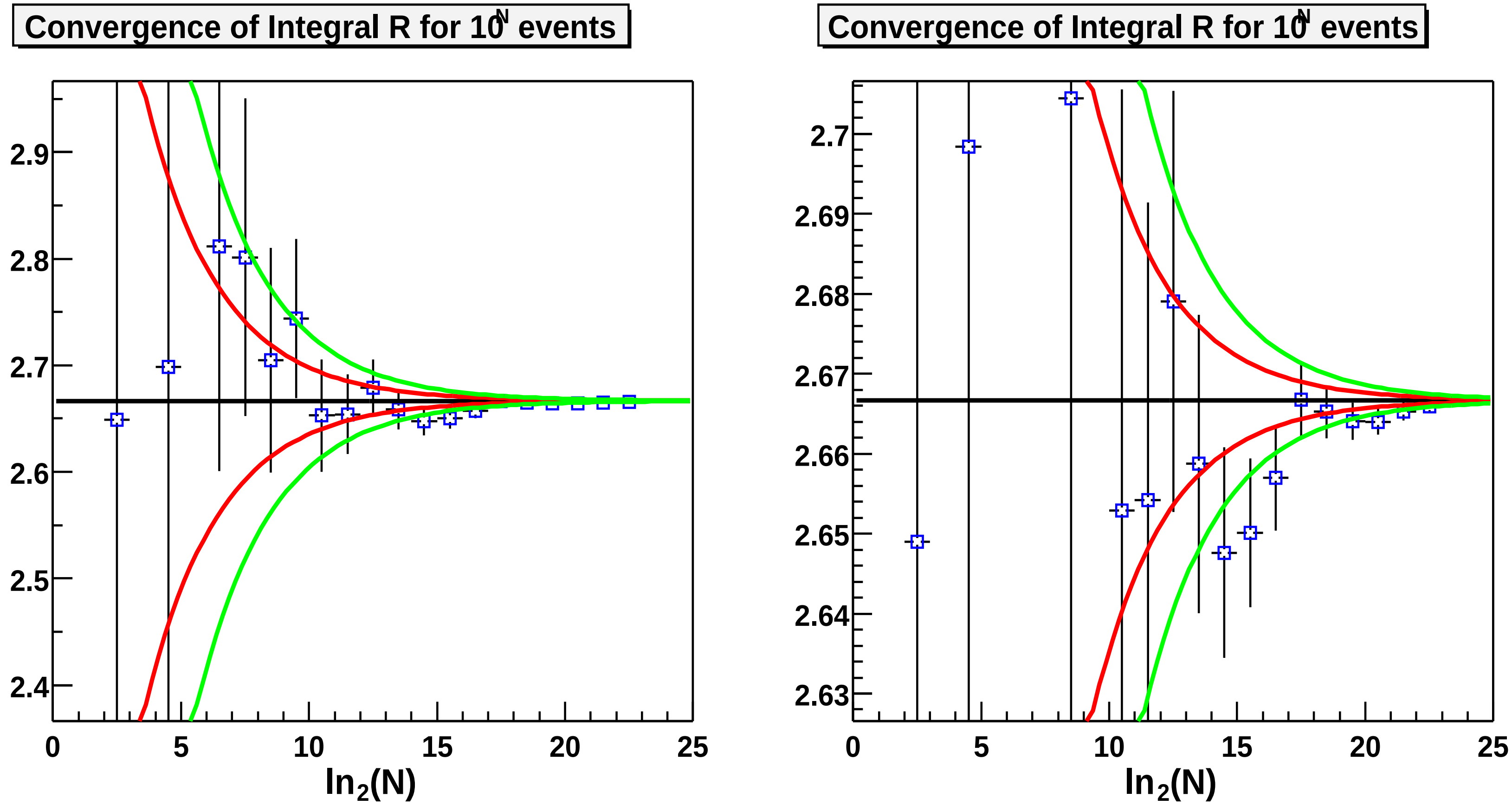
Simple code providing weighted event (z, w):

```
void GenCthe(TRandom *RNgen, Double_t &z, Double_t &w){
// p(z)=(3/8)(1+z)^2 distribution
  Double_t r = RNgen->Rndm(0);
  z = -1.0+2.0*r;
  Double_t p = 0.5;
  Double_t rho = (1+z)*(1+z);
  w = rho/p;
}
```

Simple example of MC integration, Convergence

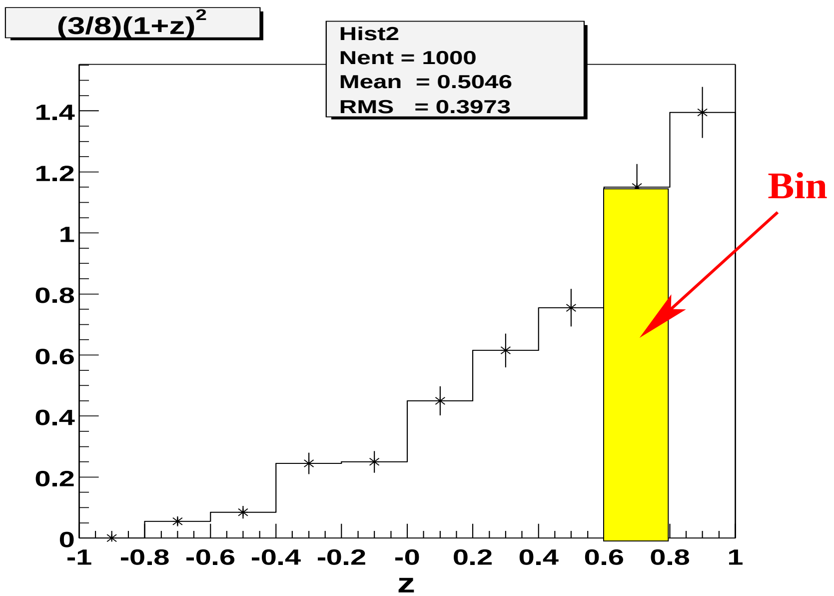
MC error $\delta R_N = \sigma/\sqrt{N}$ of integral estimator falls like $\sim 1/\sqrt{N}$ only!!!

Probability $|R - R_N| < 2\delta R_N$ is 96%.



Probab. of $|R - R_N| > \delta R_N$ is 32%. Not so rare. Red band: $\pm\delta R_N$. Green: $\pm 2\delta R_N$.

What is histogram?



Histogram provides estimate for 1-dim. distribution of the observable $z(x) \in [z_1, z_2]$,

$$\frac{dp}{dz}(z) = \int_\Omega d^n x\; p(x)\; \delta(x - Z(x_1, x_2, ..., x_n)\theta(z + z_1)\theta(z_2 - z)),$$

$$\frac{dR}{dz}(z) = \int_\Omega d^n x\; \rho(x)\; \delta(z - Z(x_1, x_2, ..., x_n)\theta(z + z_1)\theta(z_2 - z)),$$

Defi ne n_b equal bins, **Bin$_i$=** $[z_1 + (i-1)\Delta z,\; z_1 + i\Delta z]$, $i = 1, 2, ...n_b$, $\Delta z = \frac{z_2 - z_1}{n_b}$.

Step-line: $$\frac{\Delta R}{\Delta z}(z \in \mathrm{Bin}_i) = \lim_{N\to\infty} \frac{1}{N\Delta z} \sum_{Z(x_I)\in \mathrm{Bin}_i} w(x_I) \equiv \frac{1}{\Delta z} \int_{z\in \mathrm{Bin}_i} \frac{d\rho}{dz}(z) dz$$

Continuous limit: $$\frac{dR}{dz}(z) = \lim_{n_b\to\infty} \frac{\Delta R}{\Delta z}(z)$$, for error estimate, see next slide...

What is histogram? Cont.

Error estimate of the bin content:

$$\frac{\Delta R}{\Delta z} \simeq \frac{1}{\Delta z}\frac{1}{N}\sum_{\rm Bin} w = \frac{1}{\Delta z}\langle w_{z\in {\rm Bin}}\rangle$$

N is total no. of **all** MC events in **all** bins and beyond!

$$\delta\frac{\Delta R}{\Delta z} \simeq \frac{1}{\Delta z}\frac{1}{\sqrt{N}}\sqrt{\frac{1}{N}\sum_{\rm Bin} w^2 - \frac{1}{N^2}\Big(\sum_{\rm Bin} w\Big)^2} \underset{n_b>>1}{\simeq} \frac{1}{\Delta z}\frac{1}{N}\sqrt{\sum_{\rm Bin} w^2}$$

Relative error: $$\boxed{\frac{\delta\Delta R}{\Delta R} \simeq \frac{\left\{\sum_{\rm Bin} w^2\right\}^{1/2}}{\sum_{\rm Bin} w}}$$ for large bin number n_b.

All histograming packages use $n_b >> 1$ approximation. In practice one has to remember about it for $n = 2$ only.

Note: In case of MC simulation ($w = 1$ events) we get "Poissonian error":

$$\boxed{\frac{\delta\Delta R}{\Delta R} \simeq \frac{1}{\sqrt{N_i}}}$$, where $N_i =$ no. of events registered in i-th bin, for large n_b.

Central Limit Theorem (Law of big numbers)

Central Limit Theorem is “central” for the MC integration and simulation.

In the following we provide “sketchy proof” illustrated with numeric results.

Central Limit Theorem assumes “statically independent events”.

This assumption is not strictly true for events based on “pseudorandom” numbers.

In practice no problem for good quality “pseudorandom” number generators.

1D case: Average, Variance and Moments

Consider $p(x)$, normalized $\int_{-\infty}^{\infty} p(x)dx = 1$, probability distribution

Moments $\quad \mu_n = \int_{-\infty}^{\infty} p(x)x^n dx, \quad \mu_0 \equiv 1$

Average $\quad \langle x \rangle = \langle x \rangle_p = \mu_1$

Variance $\quad \sigma^2 = \mu_2 - \mu_1^2 = \langle\, (x - \langle x \rangle)^2 \,\rangle$

Generating function, 1D case

Definition $g(x) = \int_{-\infty}^{\infty} e^{tx} p(x) dx = \sum_{n=0}^{\infty} \frac{t^n}{n!} \mu_n; \quad \mu_n = g^{(n)}(0)$

Properties For convolution $p(x) = (p_1 \otimes p_2 ... \otimes p_n)(x) =$
$\int dx_1 p_1(x_1) \int dx_2 p_2(x_2) ... \int dx_n p_n(x_n) \delta(x - x_1 - x_2 ... - x_n)$
generating function simply multiply $g(t) = g_1(t) g_2(t) ... g_n(t)$

Example 1 Uniform: $p(x) = 1$ if $x \in [0, 1]; \quad g(t) = \frac{1}{t}(e^t - 1)$,
$\mu_n = \frac{1}{n+1}, \quad \mu_1 = \frac{1}{2}, \quad \mu_2 = \frac{1}{3}, \quad \sigma^2 = \mu_2 - \mu_1^2 = \frac{1}{12}$

Example 2 Normal: $p(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}; \quad g(t) = e^{t^2/2};$
$\mu_{2n} = \frac{(2n)!}{2^n} n!$, $\mu_{2n+1} = 0$

Example 3 Gaussian: $p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu_1)^2/(2\sigma^2)};$
$g(t) = e^{t^2\sigma^2/2 + t\mu_1};$

Application Convolution of two Gaussian, $p(x) = p_1 \otimes p_2(x)$:
$g(t) = g_1(t) g_2(t) = e^{t^2(\sigma_1^2 + \sigma_2^2)/2 + t(\mu_1 + \mu_2)}$,
hence $\sigma^2 = \sigma_1^2 + \sigma_2^2$ and $\mu = \mu_1 + \mu_2$

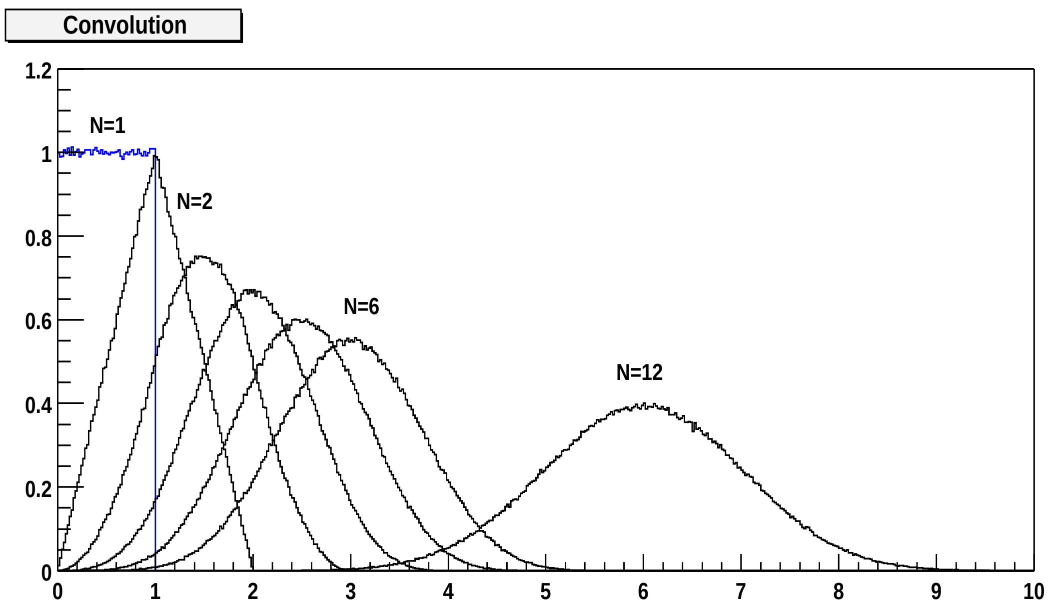
Central Limit Theorem (Law of big numbers)

Define $p_N(x) = \int \prod_1^N dx_i p(x_i) \delta(x - \sum_1^N x_i)$ (sum of independent trials)

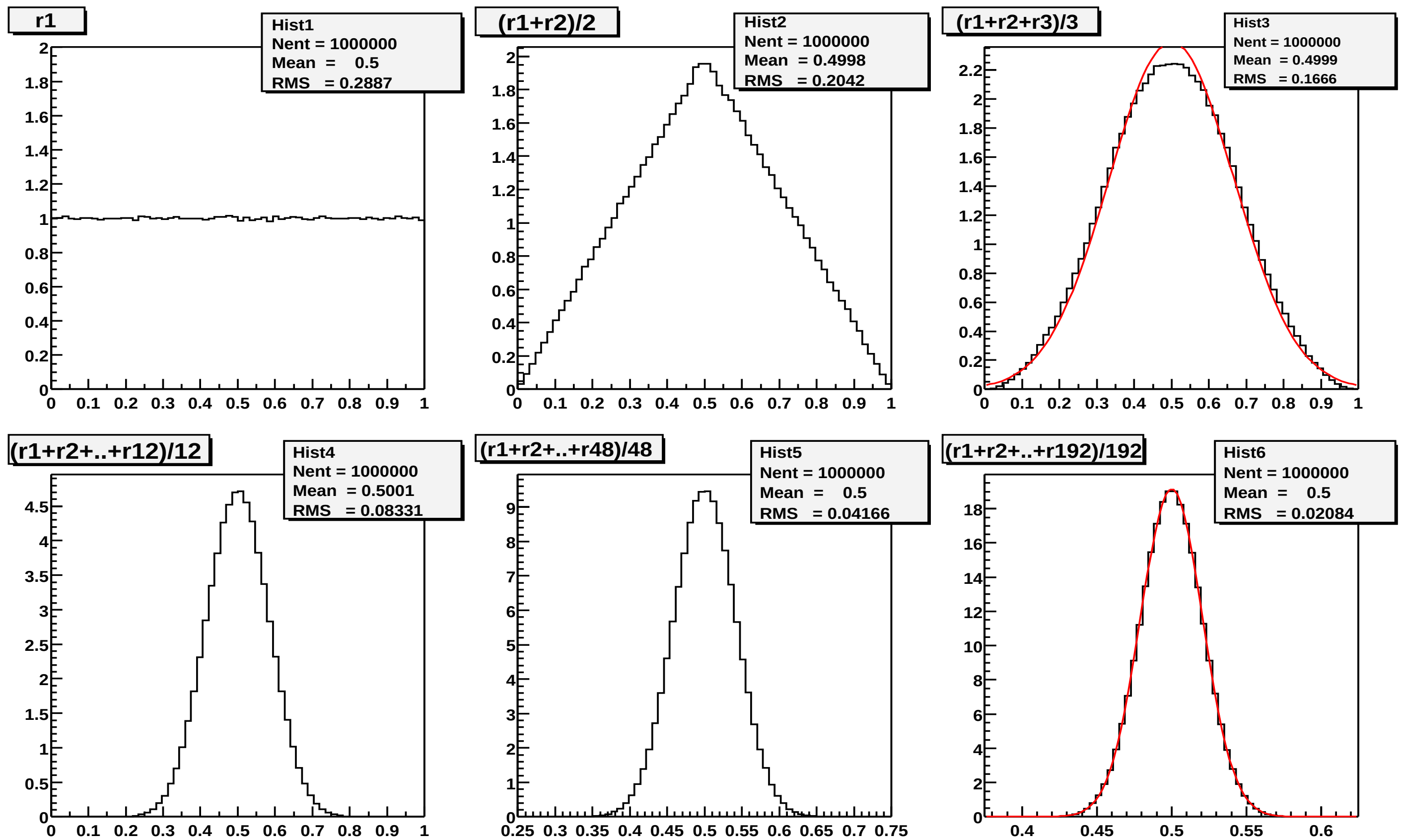
where elementary $p(x)$ is characterized by the mean μ_1 and variance σ.

Theorem $p_N(x)$, for $N \to \infty$, for ANY starting $p(x_i)$, will always converge to a normal (gaussian) distribution with an average $N\mu_1$ and variance $\sigma\sqrt{N}$

Note The $N\mu_1$ and $\sigma\sqrt{N}$ we get immediately from generating funct. $g_N(t)$.



Central Limit Theorem – Illustration 1

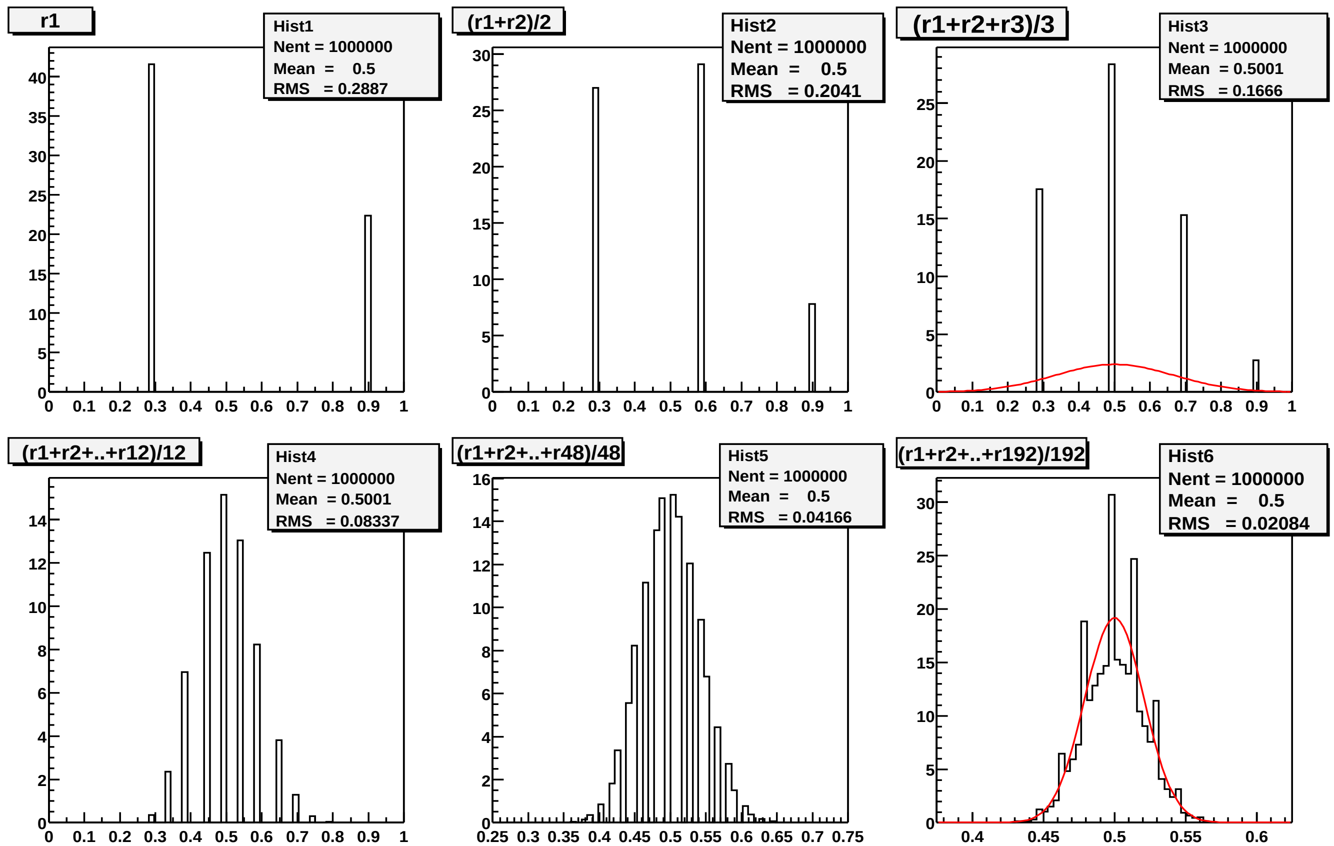


Examine $p_N(r) = \int \prod_1^N dx_i p(r_i)\; \delta(r - \frac{1}{N}(\sum_1^N r_i))$, $N = 1, 2, 3, 12, 48, 192$,

for continuous uniform distribution $p(r) = 1$, $r \in [0, 1]$,

$\mu_1 = \langle r \rangle = \frac{1}{12}$, $\sigma^2 = \frac{1}{2} = 0.83333$, $\sigma = 0.288675$

Central Limit Theorem – Illustration 2



Examine $p_N(r) = \int \prod_1^N dx_i p(r_i)\; \delta(r - \frac{1}{N}(\sum_1^N r_i))$, $N = 1, 2, 3, 12, 48, 192$,

for discrete non-uniform distribution $p(r) = q\delta(r-a) + (1-q)\delta(r-b)$,

with the same $\mu_1 = \langle r \rangle = \frac{1}{12}, \quad \sigma^2 = \frac{1}{2} = 0.83333, \quad \sigma = 0.288675$

Central Limit Theorem – Sketchy Proof

Normalized variable y $p_N(y) = \int \prod_1^N dx_i p(x_i)\delta(y - \frac{1}{\sigma\sqrt{N}}(\sum_1^N x_i - N\mu_1))$

Theorem We shall prove that $\lim_{N\to\infty} g_N(t) = \exp(y^2/2)$, that is the generating function of $p_N(y)$ is the same as that of the normal distribution.

"Proof" Expand $g_N(t) = e^{-t\frac{\mu_1\sqrt{N}}{\sigma}} \left[g\left(\frac{t}{\sigma\sqrt{N}}\right)\right]^N$ in small parameter $\frac{1}{\sqrt{N}}$:

$$g_N(t) \simeq \exp\left\{ t\frac{\mu_1\sqrt{N}}{\sigma} + N\left[g'(0)\frac{t}{\sigma\sqrt{N}} + \frac{1}{2}g''(0)\frac{t^2}{\sigma^2 N} - \frac{1}{2}(g'(0))^2\frac{t^2}{\sigma^2 N}\right]\right\},$$

remembering that $g'(0) = \mu_1$ and $g''(0) = \sigma^2 + \mu_1^2$ we get:

$g_N(t) = \exp\left\{ t^2/2 + \mathcal{O}\left(\frac{1}{\sqrt{N}}\right)\right\}$ we therefore conclude:

$$\lim_{N\to\infty} p_N(y) = \frac{1}{\sqrt{2\pi}} \exp(-y^2/2)$$

(Generating function $g(t)$ uniquely defi nes $p(x)$, because they are related by the Fourier transform which can be unambiguously inverted.)

Random numbers and r.n. generators

- Types of random numbers and their generators
- What is “Good RNGen”?
- Examples of pseudorandom RNGen’s
- Practical hints

Types of random numbers and their generators

- True random numbers. Generated from physical process. Until recently very slow. Nowadays, PC card with CDC looking into fluorescent lamp is fast source of them. 50 years ago, long tables of RN's on paper (thick book). Probably still available somewhere on WWW.
- Pseudorandom numbers. This is what we all use. Simple and fast computer program generates them. They imitate very efficiently true random numbers. (More later).
- Quasirandom numbers. Special random-like numbers used for MC integration which provide faster than $1/\sqrt{N}$ convergence (for instance $\sim 1/N$). Not covered here, our main topic is MC simulation, not integration.

What is "good random number generator"?

As random as possible? Yes and no!

- Ideally uniform (flat distribution).
- Long series (32bit integer arithmetics provides can do 10^9 only).
- Repeatability. No problem for pseudorandom numbers.
- Availability of long non-overlapping sub-series (calculations of PC farms)
- Portability. Should give the same series with different compilers, processors.
- Rather fast in terms of CPU time

Never trust, never use RNGens provided by compilers, operating systems etc.!!!

Luckily there is nowadays a plethora of good quality RNGens.

Believe it or not, RNGens are based on exact mathematical theories!

Typical examples of RNGens

- Multiplicative: $s_{i+1} = (as_i + c) \bmod m$, $r_i = s_i/m$. (Lehmer 1948).
- Lagged Fibonnaci: $s_i = (s_{i-p} \odot s_{i-q}) \bmod m$, where $\odot = +, -, \times$ or logical XOR.
- Combination of two or more simpler RN generators.

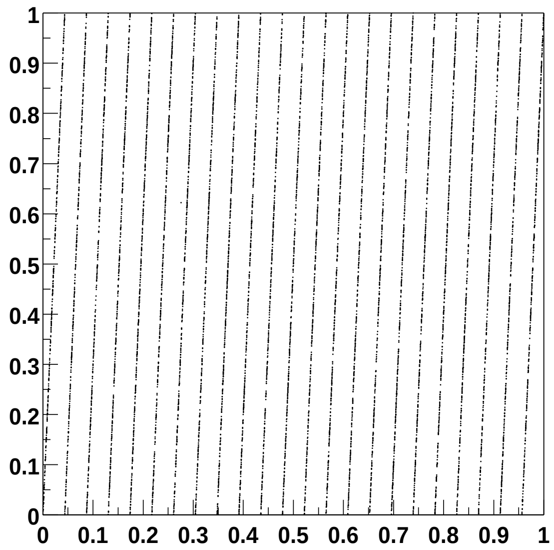
Multiplicative:

Famous coeffi cients $(c = 0)$:

1. $a = 23$, $m = 10^8 + 1$, Lehmer 1948,
2. $a = 65539$, $m = 2^{29}$, RANDU of IBM 370, the worst pitfall in history of RNGens,
3. $a = 69069$, $m = 2^{32}$, RN32 of CDC and VAX, bad for $d > 6$,
4. $a = 1664525$, $m = 2^{32}$, Knuth,
5. $a = 742938285$, $= 2^{31} - 1$ and $a = 40014$, $m = 2147483563$ L'Ecuyer.

Multiplicative RNgens, cont.

Simple Multiplicative RNgens feature famous Marsaglia planes. d-dimensional vectors $x = (s_i, s_{i+1}, ...s_{i+d-1})$ do not fill space completely. Instead they aggregate within certain hyperplanes. Potentially dangerous empty space left in between. Watch out in case of strong rejection!



Muliplicative of Lehmer (1948).

This is probably true for any RNGen, however, instead of hyperplanes vectors are placed within certain complicated "fractal structures", so it does not harm.

Lagged Fibonnaci

They are usually better, in particular provide very long series $(2^p - 1)(2^m - 1)$

Combined random number generators

Modern random number generators usually mix in a clever way two (or more) RNGens. For instance two multiplicative (RANECU od L'ecuyer, 1988) or two lagged Fibonacci (RANMAR of Marsaglia and Zaman, 1987).

Other

Every 5 years or so there is another interesting new random number generator on the market.

One may recommend RANLUX of 90's, which features switchable "luxury level" for trading CPU speed for quality.

In the present exercises I used "Mersenne Twistor" of M. Matsumoto and T. Nishimura (1998)

Practical hints:

- Never trust, never use RNGens provided by compilers, operating systems etc.
- Always use two RN generators and for final calculation switch among them.
- Leave construction of RNgens to professionals.

Conclusions

- **MC has long history and bright future.**
- **Central limit theorem is what we basically need, it provides us solid control of statistical errors.**
- **Pseudo-random number generators are all what we need and there is plenty of them.**