

First- and second-order error estimates in Monte Carlo integration

R. Bakx¹, R.H.P. Kleiss², F. Versteegen³

Radboud University Nijmegen,
Institute for Mathematics, Astrophysics and Particle Physics,
Heyendaalseweg 135, NL-6525 AJ Nijmegen, The Netherlands.

Abstract

In Monte Carlo integration an accurate and reliable determination of the numerical integration error is essential. We point out the need for an independent estimate of the error on this error, for which we present an unbiased estimator. In contrast to the usual (first-order) error estimator, this second-order estimator can be shown to be not necessarily positive in an actual Monte Carlo computation. We propose an alternative and indicate how this can be computed in linear time without risk of large rounding errors. In addition, we comment on the relatively very slow convergence of the second-order error estimate.

1 Monte Carlo integration and its errors

It does not need to be stressed that in numerical integration, including Monte Carlo (MC) integration [1], a determination or estimate of the integration error made is essential. The Central Limit Theorem (CLT) practically ensures that if the number N of MC points is sufficiently large the numerical value of the MC integral - itself a stochastic variable - will have a Gaussian distribution around the true integral value, with a standard deviation that

¹renz.bakx@student.ru.nl

²R.Kleiss@science.ru.nl

³f.versteegen@student.ru.nl

can itself also be estimated: this is the *first-order error*. The results of MC integrations are therefore usually reported as

$$\text{”result”} \pm \text{”error”}$$

with the understanding that the ”error” value quoted is the Gaussian’s standard deviation. In this way one can, for instance, assign confidence levels when comparing the integration result with a measurement. However, since the Gaussian distribution is quite steep, a modest change in the value of the error can change the confidence levels considerably. It is therefore preferable to also have a *second-order error* that estimates how well the first-order error was computed. The better way to report the result of a MC integration is then

$$\text{”result”} \pm \left(\text{”first-order error”} \pm \text{”second-order error”} \right) .$$

A first attempt to implement such a method was presented in [2]. However, in that paper no explicit form of the second-order error estimator was presented, nor were its numerical stability properties and its convergence behaviour discussed: also it was (wrongly) stated that the second-order error was the square root of the estimator, while it ought to be the fourth root. The present paper addresses and corrects these issues. In what follows we shall arrive at an estimator for the second-order error that, like the first-order one, can be evaluated in linear time *i.e.* at essentially no extra CPU cost. We shall also discuss several of its numerical aspects, and suggest an improvement.

2 Error estimators

We will start by defining some mathematical tools. We consider an integral over an integration region Γ of an integrand $f(x)$, with $x \in \Gamma$. We have at our disposal a set of MC integration points x_j , $j = 1, 2, \dots, N$, assumed to be iid (Independent, Identically Distributed) with a probability distribution $P(x)$ in Γ . We define

$$J_p = \int_{\Gamma} dx P(x) w(x)^p \quad , \quad w(x) = \frac{f(x)}{P(x)} \quad , \quad (1)$$

so that $J_1 = \int dx f(x)$, the integral we want to compute. The numbers $w_j \equiv w(x_j)$ are called the *weights* of the points. We see that J_p is nothing but the expectation value of $w(x)^p$:

$$\langle w^p \rangle = J_p . \quad (2)$$

Furthermore, we define the following multiple sums:

$$S_{p_1, p_2, \dots, p_k} = \sum_{j_1, 2, \dots, k=1}^N w_{j_1}^{p_1} w_{j_2}^{p_2} \dots w_{j_k}^{p_k} \quad (3)$$

with the condition that the indices $j_{1,2,\dots,k}$ are all *different*. As an example, the sum $S_{1,1}$ does not contain N^2 but $N^{\underline{2}} = N^2 - N$ terms. The *falling powers* are defined by

$$N^{\underline{p}} = N!/(N-p)! = N(N-1)(N-2)\dots(N-p+1) . \quad (4)$$

The simple sums S_p can be evaluated in linear time (that is, using N additions), but a multiple sum S_{p_1, \dots, p_k} needs time of the order N^k . In calculating estimators we therefore want to use only simple sums. On the other hand, only the multiple sums have a simple expectation value:

$$\langle S_{p_1, p_2, \dots, p_k} \rangle = N^{\underline{k}} J_{p_1} J_{p_2} \dots J_{p_k} . \quad (5)$$

We can relate simple and multiple sums to one another by the following obvious rule:

$$\begin{aligned} S_{p_1, p_2, \dots, p_k} S_q &= S_{p_1+q, p_2, \dots, p_k} + S_{p_1, p_2+q, \dots, p_k} + \dots + S_{p_1, p_2, \dots, p_k+q} \\ &\quad + S_{p_1, p_2, \dots, p_k, q} . \end{aligned} \quad (6)$$

We are now ready to construct the various estimators, starting with the well-known MC formulæ for clarity. For the integral we have

$$E_1 = \frac{1}{N} S_1 , \quad (7)$$

since $\langle E_1 \rangle = J_1$; moreover we see that this estimator is unbiased. For the variance of E_1 we have

$$\begin{aligned} \langle E_1^2 \rangle - \langle E_1 \rangle^2 &= \frac{1}{N^2} \langle S_2 + S_{1,1} \rangle - J_1^2 \\ &= \frac{1}{N} (J_2 - J_1^2) = \frac{1}{N^2} \langle S_2 \rangle - \frac{1}{N^{\underline{2}} N} \langle S_{1,1} \rangle \end{aligned} \quad (8)$$

so that the appropriate estimator is

$$E_2 = \frac{S_2}{N^2} - \frac{S_{1,1}}{N^2 N} = \frac{1}{N^2 N} \Sigma_2 \quad , \quad \Sigma_2 = N S_2 - S_1^2 \quad . \quad (9)$$

The latter form is more suited to computation since it can be evaluated in linear time. From Eq.(8) we see that the first-order error, defined as $E_2^{1/2}$ decreases as $N^{-1/2}$, as is of course very well known. Moreover, the expected error is defined for all functions that are quadratically integrable, as is equally well known.

The *second-order error* should have as *its* expectation value the variance of E_2 , which by the same methods as above can be shown to be

$$\begin{aligned} \langle E_2^2 \rangle - \langle E_2 \rangle^2 &= \frac{1}{N^3} \left(J_4 - 4J_3 J_1 + 3J_2^2 - 4(J_2 - J_1^2)^2 \right) \\ &\quad + \frac{2}{N^2 N^2} (J_2 - J_1^2)^2 \quad . \end{aligned} \quad (10)$$

We see that the second-order error, defined as $E_4^{1/4}$ decreases, for large N , as $N^{-3/4}$. Moreover we see that the second-order error is only meaningful for integrands that are at least *quartically* integrable. The appropriate unbiased estimator with the correct expectation value is

$$\begin{aligned} E_4 &= \frac{1}{N^4 N^3} (N^2 \Sigma_4 - 4 \Sigma_2^2) + \frac{2}{N^4 N^2 N^2} \Sigma_2^2 \quad , \\ \Sigma_4 &= N S_4 - 4 S_3 S_1 + 3 S_2^2 \quad . \end{aligned} \quad (11)$$

An important observation here concerns the asymptotic behaviour of the relative errors. Whereas the relative first-order error, *i.e.* the ratio $E_2^{1/2}/E_1$, goes as $N^{-1/2}$ according to the ‘standard’ behaviour in MC, the relative second-order error $E_4^{1/4}/E_2^{1/2}$ only decreases as fast as $N^{-1/4}$. It will therefore take much longer for the error to be well-determined than for the integral itself⁴.

A final point is in order. By the CLT we know that the distribution of E_1 in an ensemble of MC computations is normally distributed, which tells us the *meaning* of E_2 , as discussed above. Since E_2 is not computed as a simple average, *its* distribution is not governed by the *same* CLT. Nevertheless,

⁴Note that the relative errors as defined here are the *dimensionless* ratios, the only meaningful measures of performance of the computation.

as is shown in the Appendix a good case can be made for it being also approximately normally distributed, so that the relation between E_4 and the confidence levels of E_2 can be treated in the usual manner. Below, we shall illustrate this with several examples.

3 Positivity and numerical stability

In principle, equations (7), (9) and (11) are what is necessary to obtain the integral and its first- and second-order errors. However, a number of considerations must modify this picture. In the first place, the issue of positivity. Writing $w(x) = J_1 + u(x)$ so that $\int dx P(x) u(x) = 0$, we have

$$\begin{aligned} J_2 - J_1^2 &= \int dx P(x) u(x)^2 \quad , \\ J_4 - 4J_3J_1 + 3J_2^2 &= \int dx P(x) u(x)^4 + 3 \left(\int dx P(x) u(x)^2 \right)^2 \quad , \\ J_4 - 4J_3J_1 + 3J_2^2 - 4 \left(J_2 - J_1^2 \right)^2 &= \\ &= \frac{1}{2} \int dx dy P(x) P(y) (u(x)^2 - u(y)^2)^2 \quad , \end{aligned} \tag{12}$$

so that the *expectation values* of $E_{2,4}$ are positive, as they should. In addition, since with the notation $W_j = E_1 + u_j$ the Σ_2 can be written as

$$\Sigma_2 = \frac{1}{2} \sum_{j,k} (u_j - u_k)^2 \quad , \tag{13}$$

also E_2 itself is strictly nonnegative in any actual MC calculation. For E_4 this does not hold, however. A counterexample can be constructed as follows. Let us assume that the MC weights w_j take on only the values 0 and 1, and that $E_1 = Nb$, $b \in [0, 1]$. We then have

$$\Sigma_2 = \Sigma_4 = N^2 a \quad , \quad a = b - b^2 \in [0, 1/4] \quad . \tag{14}$$

The value of E_4 now comes out as

$$E_4 = \frac{1}{N^4} \left(\frac{N^2}{N} a - \frac{4N^3 - 6N^2}{N^2} a^2 \right) \quad , \tag{15}$$

which is actually *negative* for

$$a > \frac{(N-1)^2}{N(4N-6)} = \frac{1}{4} - \frac{N-2}{2N(4N-6)} . \quad (16)$$

Although by small margin (surprisingly, in this counterexample, for $b \approx 1/2$), the positivity of E_4 cannot be guaranteed, so that $E_4^{1/4}$ may be undefined. As an improvement on this situation we propose to abandon the estimator E_4 in favour of

$$\hat{E}_4 = \frac{1}{N^4 N^3} (N^2 \Sigma_4 - 4 \Sigma_2^2) . \quad (17)$$

This estimator has a slight (order $1/N$) bias, which ought to be acceptable since we are dealing with only the second-order error here; its advantage is that, since

$$N^2 \Sigma_4 - 4 \Sigma_2^2 = \frac{N^2}{2} \sum_{j,k} (u_j^2 - u_k^2)^2 , \quad (18)$$

it always evaluates to a nonnegative number.

The second issue is that of numerical stability. It is well known that already the evaluation of Σ_2 involves large cancellations which may destroy the numerical stability of the calculation and can actually lead to negative values for E_2 : this is the reason why the straightforward computation of E_2 usually cannot be reliably performed with single-precision arithmetic⁵. This problem has been widely discussed, for instance in [3, 4]. The situation of E_4 , which involves even larger cancellations, is certainly worse. To tackle these problems, we adopt the CGV algorithm first described in [4]. The strategy of this algorithm can best be summarized as follows. In the first place, one concentrates on objects that are supposed to go to a finite asymptotic value. E_1 is such an object, but $\Sigma_{2,4}$ are not. In the second place, the algorithm focuses on the update of these numbers as N is increased by 1. So let us define

$$\begin{aligned} M(N) &= S_1(N)/N , \\ P(N) &= S_2(N)/N - S_1(N)^2/N^2 , \\ Q(N) &= S_3(N)/N - 3S_2(N)S_1(N)/N^2 + 2S_1(N)^3/N^3 , \\ R(N) &= S_4(N)/N - 4S_3(N)S_1(N)/N^2 + 3S_2(N)^2/N^2 - 4P(N)^2 . \end{aligned} \quad (19)$$

⁵As anyone who has ever taught courses on Monte Carlo integration can testify.

Here we have explicitly indicated the N dependence of the running sums $S_{1,2,3,4}$. We also define

$$m = M(N-1) \quad , \quad p = P(N-1) \quad , \quad q = Q(N-1) \quad , \quad u = w_N - m \quad . \quad (20)$$

The authors of [4] have already established the update rules

$$\begin{aligned} M(N) &= m + \frac{1}{N}u \quad , \\ P(N) &= \frac{N-1}{N} \left(p + \frac{1}{N}u^2 \right) \quad . \end{aligned} \quad (21)$$

We see that in particular the computation of $P(N)$ is free of large cancellations. Some algebra leads us to supplement these update rules by

$$\begin{aligned} Q(N) &= \frac{N-1}{N} \left(q + \frac{N-2}{N^2}u^3 - \frac{3p}{N}u \right) \quad , \\ R(N) &= \frac{N-1}{N} \left(R(N-1) + \frac{1}{N} \left(p - \frac{N-2}{N}u^2 \right)^2 - 4 \left(\frac{q}{N}u - \frac{p}{N^2}u^2 \right) \right) \quad . \end{aligned} \quad (22)$$

Using these results, for any given N we then have

$$E_2 = \frac{N}{N^2}P(N) \quad , \quad \hat{E}_4 = \frac{N}{N^4}R(N) \quad . \quad (23)$$

4 A case study

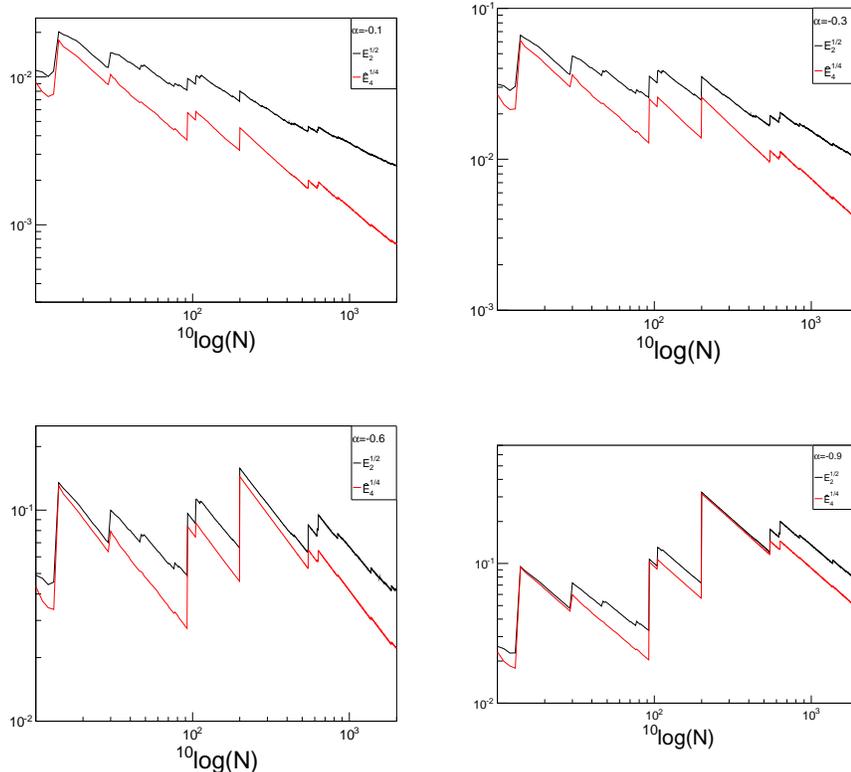
To illustrate all the above, we can perform a simple but enlightening study. Let us consider the following class of integrands:

$$f_\alpha(x) = (1 + \alpha)x^\alpha \quad , \quad x \in (0, 1] \quad , \quad -1 < \alpha \leq 0 \quad , \quad (24)$$

which we shall integrate by employing N pseudorandom numbers, iid uniformly in $(0, 1]$. These functions are all integrable (with $J_1 = 1$), but divergent as $x \rightarrow 0$. For $\alpha \leq -0.5$ they are not quadratically integrable, and for $\alpha \leq -0.25$ they are quadratically integrable but not quartically integrable. Consequently, for $\alpha \leq -0.25$ the expectation value $\langle \hat{E}_4 \rangle$ is not defined, and for $\alpha \leq -0.5$ not even $\langle E_2 \rangle$ is defined. Nevertheless, in any actual MC

calculation of this integral, $\Sigma_{2,4}$ and E_2, \hat{E}_4 will have definite, well-defined numerical values. So how, then, are these to be interpreted?

Below, we give the results for $E_2^{1/2}$ and $\hat{E}_4^{1/4}$ in a MC run where $N \leq 10^4$, monitoring their behaviour while N increases. This we do for values of α running from -0.1 down to -0.9 .



The upper line is the evolution of $E_2^{1/2}$, and the lower line displays $\hat{E}_4^{1/4}$. For the smoothest case, $\alpha = -0.1$, the N^{-1} behaviour for E_2 and the N^{-3} behaviour for \hat{E}_4 are evident⁶, marred by smallish jumps whenever an x value close to the singularity at $x = 0$ is encountered. As α decreases to -0.3 quartic integrability is lost, which can be seen from the fact that the jumps in \hat{E}_4 are now much larger while those in E_2 remain modest. Note that in all cases exactly the same set of pseudorandom numbers was used. Therefore in the various plots the jumps are in the same place, they simply become larger and larger. For $\alpha = -0.6$ where the integrand is also no

⁶Note that in the plots the values given are those of $E_2^{1/2}$ and $\hat{E}_4^{1/4}$.

longer quadratically integrable even the N^{-1} behaviour of E_2 becomes quite distorted by the growing jumps. Finally, at $\alpha = -0.9$ where the function itself is barely integrable, the jumps have become so large that the short-term N^{-1} and N^{-3} behaviour inbetween the jumps can no longer ensure this behaviour over longer N ranges. It is this kind of behaviour — short-range smooth decrease interspersed with (for increasing singularity of the integrand) increasingly large local jumps — that ruins the usefulness of \hat{E}_4 , then E_2 , and, for non-integrable functions, finally even E_1 .

From this exercise we conclude that it should always be a good idea, in any MC calculation, to monitor the behaviour of E_2 and \hat{E}_4 as N increases; and that this may tell us whether the second-order error, or indeed even the first-order error itself, can be assigned any useful meaning. It should be pointed out that, in our case study, the jumps in \hat{E}_4 are typically larger than those in E_2 and that \hat{E}_4 is therefore a more sensitive probe of possible convergence problems; and, independently of that, an estimate of how accurately the integration error itself is estimated is in our opinion *always* advisable.

Conclusions

We have argued that the current practice of MC integration, resulting in a report on the integral estimate and its error estimate, should always be accompanied by a second-order error estimate, if only to validate the assignment of confidence levels to the result (which can be, for instance, crucial in comparing the results of different MC calculations, which is good and common practice). We have presented the relevant estimators. A closer look at E_4 shows potential positivity problems and we have emended this by defining an improved estimator \hat{E}_4 . We also point out that, on the one hand, the convergence of the second-order error, $\hat{E}_4^{1/4}/E_2^{1/2} \sim N^{-1/4}$, rather than the ‘well-known’ $E_2^{1/2}/E_1 \sim N^{-1/2}$ convergence of the error itself, and that on the other hand E_2 satisfies its own version of the central-limit theorem. In addition, we have extended the methods of the Chan-Golub-Leveque algorithm [4] to allow for a numerically stable computation of not only E_2 but \hat{E}_4 as well.

Appendix

In this Appendix we will argue that the values of Σ_2 obey their own version of CLT. This is not automatically obvious, since we can write

$$\Sigma_2 = N \sum_{j=1}^N (w_j - M(N))^2 \quad (25)$$

and therefore the summed quantities are not independent of one another. Let us therefore consider a number of MC weights w_j , $j = 1, 2, \dots, N$, that are identically distributed with probability density $P(w)$ but under the constraint that

$$\sum_{j=1}^N w_j = 0 \quad . \quad (26)$$

We define

$$X = \frac{1}{N} \sum_{j=1}^N w_j^2 \quad , \quad (27)$$

and estimate the distribution of X for large N as follows. The moment-generating function of X reads

$$\begin{aligned} \langle e^{izX} \rangle &\propto \int du \, dw_1 \cdots dw_N \, P(w_1) \cdots P(w_N) \exp \left(iu \sum w_j + i \frac{z}{N} \sum w_j^2 \right) \\ &= \int du \left\{ \int dw \, P(w) \exp \left(iuw + i \frac{z}{N} w^2 \right) \right\}^N \quad , \end{aligned} \quad (28)$$

where the integrals run from $-\infty$ to $+\infty$. Introducing

$$\Phi_k(u) = \int dw \, P(w) \, e^{iuw} \, w^k \quad (29)$$

we can estimate

$$\begin{aligned} &\left\{ \int dw \, P(w) \exp \left(iuw + i \frac{z}{N} w^2 \right) \right\}^N \\ &= \exp \left(N \log \left(\Phi_0(u) + i \frac{z}{N} \Phi_2(u) - \frac{z^2}{2N^2} \Phi_4(u) + \mathcal{O} \left(\frac{1}{N^3} \right) \right) \right) \\ &\approx \Phi_0(u)^N \exp \left(iz\lambda(u) - \frac{z^2}{2N} \tau(u) \right) \quad , \end{aligned} \quad (30)$$

$$\lambda(u) = \Phi_2(u)/\Phi_0(u) \quad , \quad \tau(u) = \Phi_4(u)/\Phi_0(u) \quad . \quad (31)$$

Now, since $\Phi_0(0) = 1$ is the absolute maximum of $\Phi_0(u)$, and

$$\Phi_0(u) = 1 + iu \langle w \rangle - \frac{u^2}{2} \langle w^2 \rangle + \mathcal{O}(u^3) \quad , \quad (32)$$

we can estimate

$$|\Phi_0(u)|^2 = 1 - u^2 \sigma^2 + \mathcal{O}(u^4) \quad , \quad \sigma^2 = \langle w^2 \rangle - \langle w \rangle^2 \quad , \quad (33)$$

so that we may approximate

$$|\Phi_0(u)|^N \approx \exp\left(-\frac{u^2 N}{2} \sigma^2\right) \quad (34)$$

and the u integral is dominated by the values of u around zero; consequently,

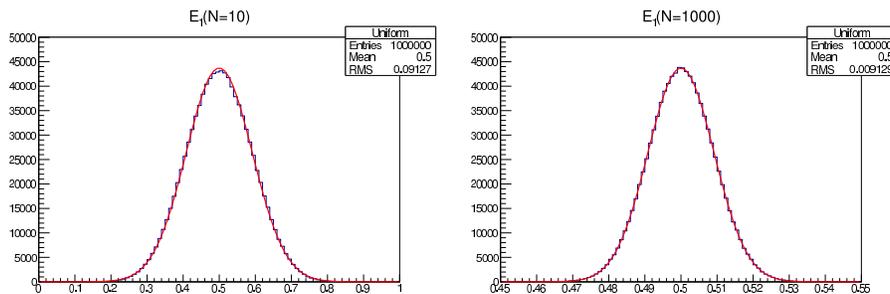
$$\langle e^{izX} \rangle \approx \exp\left(iz\lambda(0) - \frac{z^2}{2N}\tau(0)\right) \quad (35)$$

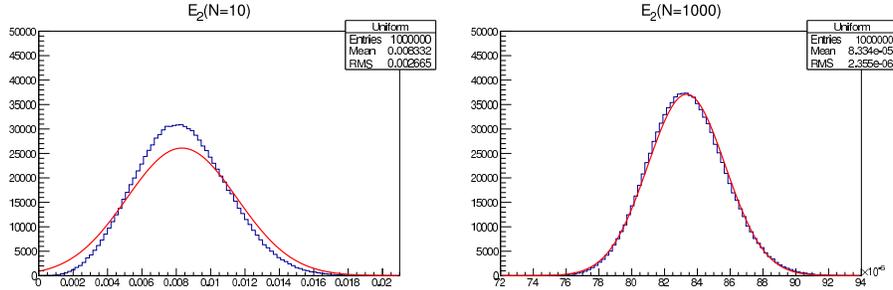
and the probability density for X to take on the value x is

$$\begin{aligned} \Pr(X = x) &\propto \exp\left(-\frac{N}{2\tau(0)}(x - \lambda(0))^2\right) \quad , \\ \lambda(0) &= \langle w^2 \rangle \quad , \quad \tau(0) = \langle w^4 \rangle - \langle w^2 \rangle^2 \quad . \end{aligned} \quad (36)$$

We see that in this sense a CLT holds for the distribution of $\sum w_j^2$.

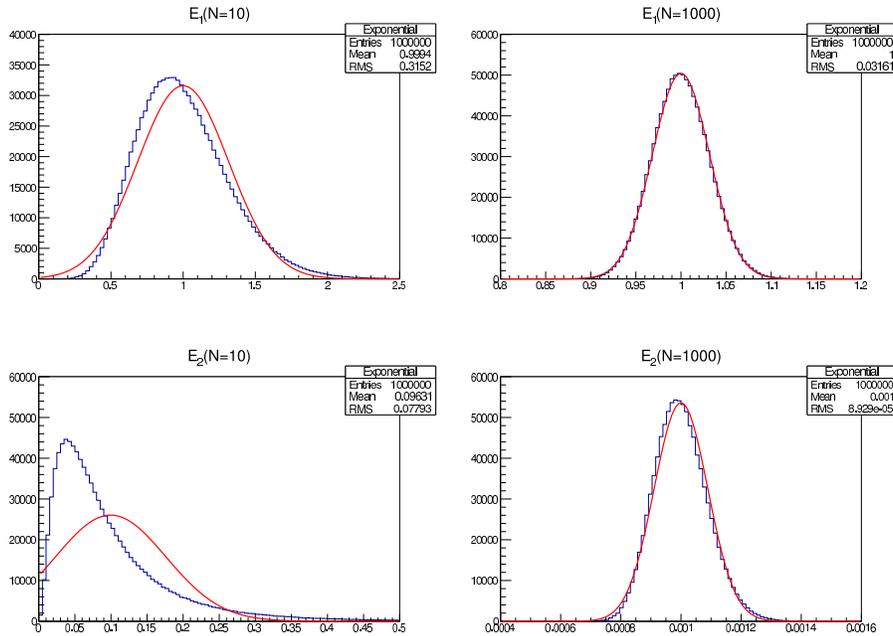
As an illustration, we generate a large number (10^6) of samples of N (pseudo-)random numbers uniformly in the interval $[0, 1]$, and compute for these $E_{1,2,4}$. Below, we give the actual distribution of the E_1 values together with the CLT Gaussian approximation with a width given by $E_2^{1/2}$. Similarly, we also give the actual distribution of the E_2 values with their CLT Gaussian approximation with width $\hat{E}_4^{1/2}$. We do this both for $N = 10$ and for $N = 1000$.





Unsurprisingly, for $N = 1000$ the CLT approximation is excellent, but for $N = 10$ it is evident that the approximation is much worse for E_2 than for E_1 .

We repeat the same exercise for numbers that are exponentially distributed, that is, with probability density $P(x) = \exp(-x)$, $x \in [0, \infty)$.

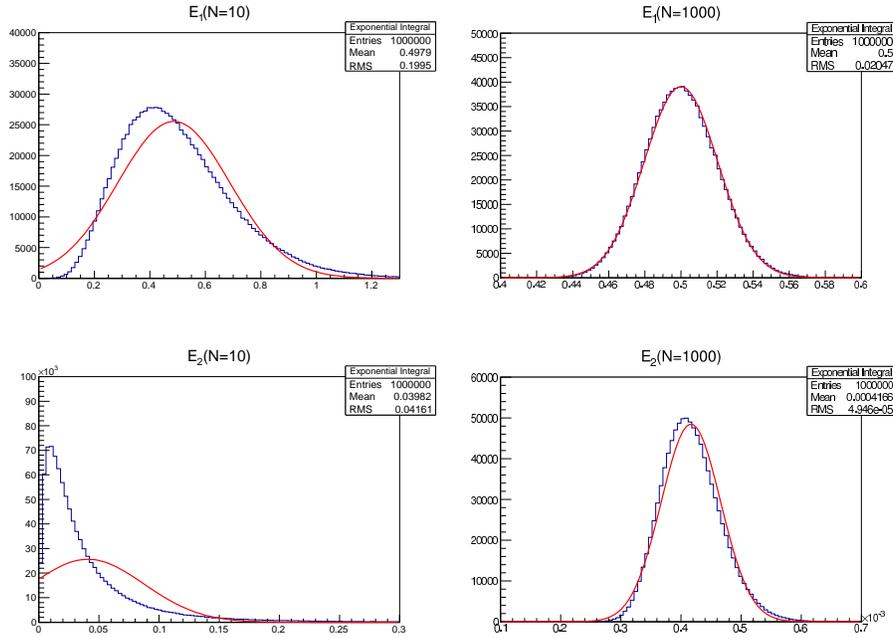


Because of the long high- x tail of this $P(x)$, the CLT approximation is appreciably worse for $N = 10$ although still very good for $N = 1000$.

Finally, we consider numbers distributed according to the exponential integral [5]:

$$P(x) = E_1(x) \equiv \int_x^\infty dt \frac{e^{-t}}{t}, \quad x \in (0, \infty), \quad (37)$$

which looks like e^{-x} for large x , and like $-\log(x)$ for x close to zero. Such a distribution, with both many low- x values and a high- x tail, is typical for how weights arising from MC event generators in particle physics are distributed.



The CLT approximation is, unsurprisingly, very poor for $N = 10$. However, for large N values it is still seen to be quite good, where we must recall that $N = 1000$ is actually quite a small number for any serious calculation.

References

- [1] N. Metropolis, S. Ulam, *The Monte Carlo Method*, J. Am. Stat. Ass. 44 (1949) 335.
- [2] F.A.Berends, R.Pittau and R.Kleiss, *Excalibur: A Monte Carlo program to evaluate all four fermion processes at LEP-200 and beyond*, Comput. Phys. Commun. **85** (1995) 437.
- [3] F. James, *Monte Carlo Theory and Practice*, Rept.Prog.Phys. **43** (1980) 1145.

- [4] T.F. Chan, G.H. Golub, R.J. Leveque, *Algorithms for Computing the Sample Variance: Analysis and Recommendations*, The American Statistician 37 (1983) 242.
- [5] M. Abramowitz, and I.A. Stegun (eds), *Handbook of Mathematical Functions*, Dover, 1965.