Statistical Errors and Monte Carlo

Bernhard Krauss
(*Universität Erlangen-Nürnberg*)

13. 4. 2005

1 Introduction

Monte Carlo studies are a very important tool in high energy physics, as they allow to simulate the outcome of a complicated experiment simply from knowing the original cross-section and the detector behaviour. General information about applied statistics and Monte Carlos studies can e.g. be obtained from references [1, 2].

There is a rather strong difference between two different applications of Monte Carlo techniques:

1. **Integration** of a complex differential probability distribution function over phase space.
2. **Simulation** of a real experiment.

As has been pointed out in reference [2] the two subjects usually need different methods although the theoretical apparatus appears to be the same at first sight. In principle both issues are important in high energy physics.

However the difference is rather clear-cut: For an integration we usually require a correctly normalised MC, which tells us the effective integrated cross-section after detector smearing and acceptance. As smearing and acceptance are usually entangled, it is important that in principle they can be understood as one abstract mathematical operation/mapping:

$$\sigma_{MC} = \int\int dx\, dy \int\int dx'\, dy' S(x, y, x', y') \frac{d\sigma}{dx'\, dy'}, \tag{1}$$

where the function S is the detector response function. It expresses the differential probability that a particle with original kinematics (x', y') is detected with new kinematics (x, y). It is clear that all applied analysis cuts can be included in S. x and y are just some arbitrary variables and in principle the required number of dimensions has to be included.

It has to be noted that the detector response function is only the mean value of the detector response and the width of random fluctuations is not explicitly considered. In fact, S is difficult to obtain from first principles and is effectively defined from the infinite statistics Monte Carlo result. It is also important that S is independent of the applied Monte Carlo Method and the original differential cross-section. I.e. if a different process with a different cross-section but the same final state is considered, the same response function S will be obtained.

The usual philosophy for integration tasks is to generate as much Monte Carlo statistics as possible. The error of such integrations can be difficult to estimate and has nothing to do with real events.

This is instead the task of a simulation. In this case it is important that the final statistical fluctuations of the results correspond exactly to the ones of a real experiment. The required MC method may be slow and/or inefficient for an integration task, but this does not matter. Such a simulation will also represent the expected experimental event rates in the limit of very low statistics.

In the following sections three different Monte Carlo approaches and their suitability for individual applications will be discussed:

1. weighted events
2. rejection method (hit-or-miss-method)
3. inverse transformation method

All of these methods are also discussed in [2], but some additional comments have to be made.

2 Weighted events

The method of weighted events is often adopted for the HERMES Monte Carlo. This method is ONLY useful for an integration, but not for a simulation (at least not without considerable modifications!). The method works in the following way:

The events are generated as uniformly distributed events in a multi-dimensional phase-space. To each event a weight w_i is assigned, which corresponds to the differential cross-section (times phase-space divided by generated number of events). As a result, the cross-section after detector effects and analysis cuts is obtained from

$$\sigma = \sum_i w_i. \tag{2}$$

This relation is exact. One may wonder, what the weighting is doing in combination with the detector simulation. The reasoning in this context is that it does not matter for which reason and how often a particle track is produced. The probability that some peculiar interaction in the detector takes place is always the same for the same track kinematics. This is the reason why the detector response function S is independent of the generating process. It is clear that in the limit of infinite statistics a weighted Monte Carlo will converge to the correct result.

Problems with the weighting method only occur in special cases: One of them is the calculation of asymmetry moments using the fit method, but only if two other conditions are fulfilled:

- the fitted function does not very well describe the asymmetry in the limit of infinite event number,
- the statistical error bars of data and MC in the fitted bins are very different.

The first condition is necessary, since the different statistical errors (second condition) have no impact as long as the fitted function is equal to the true asymmetry; the statistical errors of the asymmetry coefficients would be correct for the real data as well as for the Monte Carlo.

Let us consider a very simple example: The true asymmetry has one constant value c_1 for the first 3 bins in some variable and a different value c_2 for the other three bins. A constant function of value c is fitted. If the first three bins have a much lower statistical error than the second three bins, the extracted value of c will be close to c_1 and the estimated error will be wrong. If the second three bins have a much lower error than the first three bins, c_2 will be obtained and again the error will be wrong. In fact, not only the estimated error is wrong, but also χ^2 does not indicate a problem with the fit.

Fortunately, the two conditions mentioned above do not affect gmc_dvcs. First of all the fitted function does quite well describe the true asymmetry and in addition, the statistical errors bars in all ϕ-bins are typically of similar size (as for the data).

Another important comment is that reweighting of the final events (e.g. by some ϕ-dependent asymmetry) may be problematic if the original event weight has been used in the generation process to determine a fractional contamination of some other process (like in gmc_disng). Obviously this contamination is only correct if the original event weight is not reweighted in the end.

For other generators (like gmc_dvcs) no such problems are encountered, since the event weight is computed at the beginning and never looked at again until the analyser uses it.

Weighted Monte Carlos have very nice features: E.g. an unweighted MC may generate many events in a region that is outside the detector acceptance. However it may be very difficult to state exactly, where the acceptance (including smearing) stops. On the other hand a weighted MC does not care about this problem: Events are generated in the interesting region and if some events have a high weight, but never enter into the acceptance, one does not care. Weighted MCs can also be used to calculate theoretical asymmetries in kinematical regions, where the cross-section is low.

The main problem with weighted Monte Carlos is their statistical behaviour. They never behave like real data and it is problematic to estimate the convergence and the error of the result. Usually some definitions are made: The error on σ is calculated as

$$d\sigma = \sqrt{\sum_i w_i^2}, \tag{3}$$

which already assumes, that the events are distributed according to a Poisson distribution. The equation can be understood in the following way: Let's assume that in 50% of the accepted events the event weight is w_1 and in the other 50% of the cases the event weight is w_2. The error is then simply

$$\begin{aligned} d\sigma &= \sqrt{(d\sum_i^{N_1} w_1)^2 + (d\sum_j^{N_2} w_2)^2} \\ &= \sqrt{(w_1 dN_1)^2 + (w_2 dN_2)^2} \\ &= \sqrt{(w_1\sqrt{N_1})^2 + (w_1\sqrt{N_2})^2} \end{aligned}$$

$$= \sqrt{N_1 w_1^2 + N_2 w_2^2}, \quad (4)$$

where N_1 and N_2 are the corresponding event numbers to the two different weights and are equal in this simple case. It is obvious that in the limit of many different weights equation 3 is recovered.

Now the following problem is encountered: The error defined in this way is the best possible estimate of the true error, but not necessarily correct. There are two situations in which the error estimate will be wrong:

Let's assume that a flat cross-section in one variable (e.g. ϕ) is integrated with the weighting method. If N events are generated, the calculated cross-section is $\sigma = (N/N) \times 2\pi dN/d\phi$, which is certainly correct and has no error at all (as the results is always the same). The error estimate according to equation 3 results in $\sigma = \sqrt{(N/N^2) \times (2\pi dN/d\phi)^2}$ which is $\sigma/\sqrt{N}$. Obviously the estimated error never reflects the true statistical fluctuations if several Monte Carlo samples are compared.

The problem in this case is due to the fact that the original event distribution of generated events was not Poissonian. In fact the number of events was exactly known to be N and thus equation 3 is simply wrong. One could make the distribution Poissonian by accepting e.g. only a fraction p of the events and rescaling the event weight by $1/p$. This method may not be very efficient as p must really be small. The difference between the Poissonian $dN = \sqrt{N}$ and the error $dN = \sqrt{N} \times (1-p)$ of the binomial distribution still gives a difference of 5%, if $p = 0.1$.

It has to be noted that weighted MCs usually achieve a Poissonian distribution in the following way: If the interval in ϕ is split into two bins, p in each bin becomes 0.5 and thus the cross-section in each bin has an error which is only overestimated by a factor of $1/\sqrt{2}$ (however, the errors in the two bins are of course completely correlated). The more crucial effect in practical applications is that only a fraction of the generated events passes all analysis cuts. Thus the remaining event distribution becomes effectively Poissonian.

There is also a second situation in which the error estimate will be completely wrong, namely in the limit of low statistics. If one tries to use weighted Monte Carlos for detector simulations, one may be tempted e.g. to generate the same number of events as experimentally observed. Looking at equation 3 this is certainly not correct because the statistical error may be much worse than for the real data. A second slightly better way would be to generate the same number of equivalent events defined as

$$N_{eff} = \frac{\sigma^2}{d\sigma^2}. \tag{5}$$

It is obvious that for a constant event weight this is simply the number of generated events. If one event weight is much bigger than all others, $N_{eff} = 1$ will be found, which indicates, that the result is essentially unknown.

If one now generates the same number of effective events as experimentally observed events, the result may be ok for very flat cross-sections, but in general it will still be wrong. This has the following reason:

Let's assume that the cross-section is flat in ϕ, but huge (like a factor 100 bigger) in 1% of the ϕ-range. If one generates 5 effective events, they will usually correspond to 5 generated events and the estimated error will usually be $d\sigma = \sigma/\sqrt{5}$. Only in very rare cases, namely with a probability of 5% at least one in 5 generated events will hit the high cross-section region. If one of 5 generated events hits this region the effective number of events will be 1.08, and the problem becomes obvious, but this almost never happens. In histograms this problem can usually be identified by spikes and/or fluctuations that are outside the estimated errors. But there is no guarantee to find it.

On the other hand it is possible to improve the convergence of a weighted Monte Carlo by suitable event generation. The basic idea is that the assigned event weight should be as constant as possible. In the example above one would try to generate a lot more events in the high cross-section region, which can be achieved by choosing a different variable (and not ϕ) for the uniform event generation process.

As a conclusion the weighted Monte Carlo should always be used to perform cross-section calculations in the limit of infinite statistics. In contrary to experimentally observed events the production of more Monte Carlo statistics may actually increase the estimated error and not decrease it. If a sufficient number of events is discarded, the statistics will be Poissonian and all observed errors will be uncorrelated and correctly estimated.

If a weighted Monte Carlo has to be used for a simulation (because nothing else is available), one should take the cross-section σ (from a huge statistics MC) and calculate the expected number of events in each bin. Then e.g. the CERNLIB-Routine RNPSSN can be used to generate the observed event number according to a Poissonian distribution.

3 Rejection Method

The rejection method is a very simple but time-consuming method that can be used for integration as well as for simulation. This means that the obtained events usually behave like true events and the statistical error will always decrease with increasing statistics.

The method generates e.g. some event in ϕ from a uniform distribution. Then the ratio of the differential cross-section and the maximum differential cross-section is calculate. Another random number between 0 and 1 is generated and if this number is smaller than the ratio, the event is accepted and stored.

In this case it may be difficult to set a reasonable maximum weight and thus the method is not very fast. In addition it is almost impossible to calculate asymmetries for low cross-section regions (for such an integration the weighted MC will be much faster).

It is clear that the statistical error of N reconstructed events is simply $\sqrt{N}$ but also there one has to be careful: In regions where the differential cross-section is close to its maximum value the statistical error is again a binomial and not a Poissonian one. This means that in contrast to the real experiment the real error is reduced by a factor of $\sqrt{(1-p)}$ such that the real error is only 30% of the estimated error if the average cross-section is 0.9 times the maximum cross-section and all simulated events are considered in one bin.

Especially for applications like the BH-cross-section the event rejection method is prohibitively slow.

In order to perform a cross-section integration with this method, the maximum differential cross-section is multiplied by phase-space and then by the ratio of accepted events to generated events.

4 Inverse Transformation Method

The inverse transform method is by definition the most intelligent way to do a simulation. However it only works for simple cross-sections or with binned step-like cross-sections. The details are again given in reference [2].

The idea is the following: It may be possible to find a coordinate space in which a differential cross-section is completely flat. If one generates uniform distributions in such a coordinate space, the obtained statistics will automatically reflect the correct cross-section. For simple curves like a Gaussian or a Lorentzian this procedure can be solved analytically by inverting the integrated, normalised distribution.

For more complicated cases one can also use histograms as a starting point. Multi-dimensional problems can be solved by integration of the cross-section over each variable (one after the other) and subsequent selection of each variable starting at the top level. This is e.g. done in the unpolarised radiative correction code in gmc_disng.

As can be seen this method will get very complicated in higher dimensions and it has also some systematic problems due to the required numerical integration before the event is generated.

Again the statistical error on N events is usually $\sqrt{N}$ but this estimate will be wrong if the whole range in ϕ is taken and no events are rejected by cuts or acceptance. Again there is no error in the number of events and thus the error gets overestimated. Note that the cross-section integration with the inverse transformation method works in a similar way as with the weighting method. The integrated cross-section for the process is known from the numerical integration. Thus the cross-section per bin i is simply the total cross-section rescaled by N_i/N.

5 Conclusion

It is always necessary to keep advantages and disadvantages of each Monte Carlo method in mind. Extreme care is needed if weighted event Monte Carlos are operated with limited statistics. The result as well as the estimated error can be completely wrong. Under special conditions also the extraction of asymmetry moments can be dangerous.

On the other hand weighted event Monte Carlos have the advantage that nothing has to be known about the integrated cross-section and the acceptance function. The process is relatively fast and asymmetries can be extracted even if the cross-section is low.

The rejection method is a very slow but sometimes useful method to generate complicated cross-sections without too many mathematical tricks. It can be used for event simulations in the same way as the inverse transformation method. This last method can also be very slow and the numerical accuracy of the cross-section integration is an important point. However, it is often the preferred method for Monte Carlo simulations.

Due to the advantages/disadvantages of the different methods, the rejection method has now been implemented as an additional option in gmc_dvcs. The efficiency of this method has been optimised and is now suitable for simulations.

References

[1] L. Lyons, ”Statistics for nuclear and particle physicists”, Cambridge University Press, Cambridge, 1986.

[2] W.J. Metzger, ”Statistical Methods in Data Analysis”, Experimental High Energy Physics Group, Katholieke Universiteit Nijmegen, 2002.

[3] HERMES, ”Monte Carlo documentation”, /hermes/r21/gmc/help.