Lecture 10. Errors in Monte Carlo Simulations.

One of the main differences between rigorous and slipshod mathematical proofs is that in the former one always bounds the importance of neglected terms. Of course, we assume that the logic of the proof is correct in both cases. Similiarly, good numerical simulations are distinguished from bad ones not just by the correctness of the algorithm and program, but also by the accuracy of error estimates. The main ingredient is common sense. Some of the remaining technicalities are dealt with in this lecture.

The usual problem of estimation starts from a set of random variates $\{x_i\}$ and some function over them, yielding the data set $\{f_i\}$. We are usually required to estimate the mean, $\overline{f}$, and its error, $E(\overline{f})$. Usually one assumes that the central limit theorem and applies the formulæ

$$\overline{f} \;=\; \frac{1}{N}\sum_{i=1}^{N} f_i, \qquad E(\overline{f}) \;=\; \sqrt{\frac{1}{N-1}\left(\frac{1}{N}\sum_{i=1}^{N} f_i^2 - \overline{f}^2\right)}. \tag{1}$$

The interpretation used is that there is a 68% probability that the mean lies in the range $\overline{f} \pm E(\overline{f})$.

The most efficient way of programming Eq. (1) is to use only one loop for the summations in both the formulæ. This is demonstrated in the FORTRAN program fragment below.

```
      av=0
      er=0
      do i=1,ndata                                % There are ndata pieces of
         av=av+x(i)                               % data stored in the array
         er=er+x(i)*x(i)                          % x. Note the single loop.
      end do
      av=av/ndata
      er=sqrt((er/ndata-av*av)/(ndata-1))
```

The danger in using Eq. (1) blindly is that one neglects the cases where the central limit theorem breaks down. This may happen, for example, if the measurements f_i are not independent, but actually have correlations. The solution to this problem are dealt with in the next section. Even if there are no correlations, N may not be large enough for the distribution of $\overline{f}$ to be Gaussian. One solution to this is also given.

We shall use the symbol $\langle f\rangle$ to denote ensemble averages of a quantity f, and $\overline{f}$ for estimates of $\langle f\rangle$ using some finite data set of length N.

1. Autocorrelations and Errors.

Consider the mean-squared deviation in the average of the measurements. This is given by the formula ($\Delta_i = f_i - \overline{f}$)

$$\begin{aligned} &\left\langle \frac{N}{N-1}\left(\frac{1}{N}\sum_{i=1}^{N}\Delta_i\right)^2\right\rangle = \\ &\qquad \frac{1}{N(N-1)}\left[\sum_i \langle\Delta_i^2\rangle + 2\sum_i \langle\Delta_i\Delta_{i+1}\rangle + \cdots + 2\sum_i \langle\Delta_i\Delta_{i+n}\rangle + \cdots\right]. \end{aligned} \tag{2}$$

Now define the **autocorrelation function** of the measurements f_i by the formula

$$C(t) = \frac{\langle\Delta_i\Delta_{i+t}\rangle}{\langle\Delta_i^2\rangle}. \tag{3}$$

Clearly, the mean square deviation is given by

$$\left\langle \frac{N}{N-1}\left(\frac{1}{N}\sum_{i=1}^{N}\Delta_i\right)^2\right\rangle = \frac{\langle\Delta_i^2\rangle}{N(N-1)}\left[1 + 2\sum_{t=1}^{\infty} C(t)\right]. \tag{2}$$

The term within the square brackets is called the **integrated autocorrelation time**, τ_{int}. In the presense of autocorrelations, an estimate of the error in $\langle f\rangle$ is therefore given by $\sqrt{\tau_{int}\mathrm{Var}(\langle f\rangle)}$. Since $\tau_{int} > 1$, error estimates by Eq. (1) always lead to underestimates of the actual error.

To obtain an accurate error estimate, one must first estimate the autocorrelation time. If this is known, then a simple procedure would be to divide the data set into some number, B, of blocks, such that each block is much greater than τ_{int}. Then estimate the mean and error by applying Eq. (1) to the averages of f over each block.

$C(t)$ may be estimated from the data using the formula

$$C(t) = \frac{\frac{1}{N-t}\sum_{i=1}^{N-t}(f_i - \overline{f})(f_{i+t} - \overline{f})}{\frac{1}{N}\sum_{i=1}^{N}(f_i - \overline{f})^2}. \tag{4}$$

The integrated autocorrelation time can be computed directly from this. Usually, when $C(t)$ is small, this is a noisy estimator, and hence the sum over all t required to estimate τ_{int} is also noisy. It is better, then to truncate the sum at some T, and look for a plateau in the estimate of τ_{int} as T varies.

Here is a rough and ready FORTRAN code for the analysis of correlations and errors in a set of data. Note that a computation of the mean and naive variance takes effort of order N. Computing the correlation function takes effort of order N^2. As a result, it is best to compute only as much of the correlation function as necessary.

The loop over t in the computation of $C(t)$ (starting on line 27) therefore has an escape activated when Var(x)$C(t)$ becomes less than some preset ϵ.

```
 1        subroutine quick(ndata)
 2  ********************************************************************
 3  *                                                                  *
 4  *     A rough estimate of autocorrelations in data.                *
 5  *        ndata   number of data points (must be less than nmax)    *
 6  *        x       array containing the data set (in common block)   *
 7  *                                                                  *
 8  ********************************************************************
 9        parameter (nmax=5000, epsilon=1.0E-3)
10        common /data/ x(nmax)
11  ********************************************************************
12  *     Compute the average and variance of the data set.            *
13  ********************************************************************
14        av=0
15        er=0
16        do i=1,ndata
17           av=av+x(i)
18           er=er+x(i)*x(i)
19        end do
20        av=av/ndata
21        er=er/ndata-av*av
22        write(6,6001) av, er
23  ********************************************************************
24  *     Compute the correction due to autocorrelations.              *
25  ********************************************************************
26        tau=er                                   % initialisation
27        do it=1,ndata                            % loop over t
28           add=0
29           do i=i,ndata-it                       % loop over the data set
30              add=add+x(i)*x(i+it)
31           end do
32           add=2*(add/(ndata-it)-av*av)          % addition to tau(int)
33           write(6,6002) it, add/er              % the correlation function
34           if (abs(add).le.epsilon) go to 10     % escape condition
35           tau=tau+add                           % add next term to tau
36        end do
37  10    write(6,6003) tau/er, sqrt(tau/(ndata-1))
38  *
39        return
40   6001 format(5x,'Error Analysis with ',i5,' data points'/
41       1 5x,'Average='e12.4,5x,'Naive error=',e12.4/
42       2 /5x,'Correlation function')
43   6002 format(5x,2e14.4)
44   6003 format(/5x,'Integrated autocorrelation time=',e12.4/
45       1 5x,'Corrected error=',e12.4)
46        end
```

In the case of interest, the measurements are generated by a Monte Carlo algorithm which proceeds through a Markov chain. Hence,

$$C(t) \;=\; \sum_n C_n \exp(-t/\tau_n), \qquad (\tau_n > 0). \tag{5}$$

We will use the convention that $\tau_1 \geq \tau_2 \geq \cdots$. Then, for $t >> \tau_1 - \tau_2$, only the leading exponential behaviour remains, and one can estimate τ_1 by the method of **local masses**

$$m(t) \;=\; \log\left(\frac{C(t)}{C(t+1)}\right) \;\to\; \frac{1}{\tau_1}. \tag{6}$$

Given $C(t)$, this variable can be constructed for all t in order to check whether $m(t)$ goes to a constant, and then extract τ_i from this constant.

Assuming the form in Eq. (5), we see that

$$\tau_{int} \; = \; 1 + 2 \sum_n C_n \tau_n. \tag{7}$$

Hence, the estimator for $E(\overline{f})$ in Eq. (1) can be correct only if all τ_i or C_i are zero, *i.e.*, there are no autocorrelations. As τ_1 increases, so does τ_{int} (unless $C_1 = 0$). In a Metropolis simulation of a system at a critical point, τ_i becomes infinite, thus leading to the phenomenon of **critical slowing down**. This means that accurate measurements of critical properties of models through Metropolis simulations require infinite time. Special algorithms have to be designed to minimise the effect of this phenomenon.

2. Modern Statistical Methods.

The central limit theorem, which is the basis of Eq. (1) may break down when N is small or moderately large. In this case, Eq. (1) may give grossly inaccurate results. For such values of N one needs estimators for statistical properties which are independent of models for the distribution function of the random variates.

A second problem that often arises in the estimate of various quantities is that of bias. An example is the computation of $\langle \exp x \rangle$. The obvious estimator consistently overestimates the expectation value by a number that goes to zero as $1/N$ as $N \to \infty$. The estimator we seek should also incorporate a correction for this bias.

We introduce **Jack-knife estimators** for a function F of the data set $\{x_i\}$ by the following algorithm.

1) Construct the quantities

$$F_0 = F\big(\{x_j\}\big), \qquad F_i = F\big(\{x_j | j \neq i\}\big). \tag{8}$$

2) Construct the jack-knife variables $J_i = N F_0 - (N-1) F_i$.
3) Construct the estimator $\overline{F}$ and the error $E(F)$

$$\overline{F} \; = \; \frac{1}{N} \sum_{i=1}^{N} J_i, \qquad E(F) \; = \; \sqrt{\frac{1}{N-1} \left(\frac{1}{N} \sum_{i=1}^{N} J_i^2 - \overline{F}^2 \right)}. \tag{9}$$

These two estimators are claimed to be model-independent and free of $1/N$ biases.

It is easy to check that the Jack-knife estimator of a simple average of the data set is the same as the usual estimator given in Eq. (1). To see this, take

$$F_0 \; = \; \frac{1}{N} \sum_j x_j, \qquad F_i \; = \; \frac{1}{N-1} \sum_{j \neq i} x_j.$$

Then J_i turns out to be exactly equal to x_i. When F is a non-linear function of the data-set, the Jack-knife estimators are different from the normal estimators.

The most important use of the Jack-knife method is for the estimates of local masses. A FORTRAN program segment for jack-knife analysis of the quantity

$$f \;=\; \log\left(\frac{\langle x\rangle}{\langle y\rangle}\right),$$

is given below.

```
      subroutine ratio(ndata)
************************************************************************
*                                                                      *
*     The logarithm ratio of the averages of two sets of data.         *
*        ndata   number of data elements                               *
*        x       data for the numerator                                *
*        y       data for the denominator                              *
*                                                                      *
************************************************************************
      xnum=0
      xden=0
      do i=1,ndata
         xnum=xnum+x(i)
         xden=xden+y(i)
      end do
      f0=log(xnum/xden)                         % the variable f_0
*
      mdata=ndata-1
      do i=1,ndata
         xn=xnum-x(i)                           % numerator without x(i)
         xd=xden-y(i)                           % denominator without y(i)
         fi=log(xn/xd)                          % the variable f_i
         yi=ndata*x0-mdata*xi                   % the jack-knife variable
         avj=avj+yi
         erj=erj+yi*yi
      end do
      avj=avj/ndata                             % the jack-knife average
      erj=sqrt((erj/ndata-avj*avj)/mdata)       % the jack-knife error
      write(6,6001) avj, erj
*
      return
 6001 format(5x,'Jack-knife Analysis of ratio: ',2e12.4)
      end
```

When the number of data points, N, is very large, the numerical effort in the jack-knife may be very steep. In this case it may be better to partition the data into $\mathcal{N} << N$ blocks are perform the jack-knife by leaving out one block at a time. If the data contain autocorrelations, then blocking is a must. The size of each block should be significantly larger than τ_{int}.

3. The Ferrenberg-Swendson Method.

For a phase space Γ with Hamiltonian $H(\mathbf{x})$ ($\mathbf{x} \in \Gamma$), the canonical weight has the special form $W(\mathbf{x}) = \exp(-\beta H)$. Thus, a simulation at one β can be used to obtain information on a nearby $\beta' = \beta + \Delta\beta$, since

$$Z(\beta') \;=\; Z(\beta)\langle \exp(-\Delta\beta H)\rangle .$$

This forms the basis of the **reweighting** or the Ferrenberg-Swendson method.

The expectation value of some phase space function $A(\mathbf{x})$ can also be obtained by reweighting, provided the values of the Hamiltonian, H_i, are stored along with A_i. The reweighting formula is

$$\langle A\rangle_{\beta'} \;=\; \frac{\frac{1}{N}\sum_{i=1}^{N} A_i \exp(-\Delta\beta H_i)}{\frac{1}{N}\sum_{i=1}^{N} \exp(-\Delta\beta H_i)}. \tag{10}$$

Since this is a strongly non-linear expectation value, the error estimate should be obtained by jack-knife. The bias-correction in the jack-knife estimator is also useful here.

Extrapolations of this kind may be strongly affected by autocorrelations in the original data. In order to prevent this, one should use well seperated measurements of A_i and H_i. A good rule of the thumb is that no two measurements used in the analysis should be seperated by less than $2\tau_{int}$ updates.

Problems

10.1) Reanalyse the data (for the magnetisation, m, the susceptibility, χ, and the specific heat c_V) generated in the Metropolis simulations of Langevin paramagnetism. While writing the programs for the analyses described below, keep in mind that you will have to use them for other problems later.

(a) By summing the correlation functions, find τ_{int} and its error. By putting different upper limits, T, on the sum over t defining τ_{int}, check whether there is a plateau in τ_{int} plotted as a function of the upper limit. Before doing a sophisticated analysis of the autocorrelations, it may be a good idea to perform a rough and ready computation of τ_{int}. This can save a lot of time, specially if it turns out that τ_{int} is much less than the number of data points you have stored.

(b) Measure the autocorrelation functions for m and χ. Use a jack-knife procedure for finding the errors in $C(t)$. Estimate the local masses by jack-knife and find τ_1. Give an accurate estimate of the error.

(c) Are τ_1 and τ_{int} equal in the measurements of m and χ? Are these two measures of the autocorrelation time dependent on β and B? Illustrate by plotting the data.

(d) Using the information on τ_{int} correct the previous estimates of errors. Find also the error estimates through the variance of block averages where the sizes of the blocks are $5\tau_{int}$. Do these estimates agree? Give reasons.

(e) Perform reweightings based at different β and draw curves for m, χ and c_V as functions of β, drawing only one measurement out of every $N\tau_1$ (choose $N = 2, 4, 6$). Estimate errors on these extrapolations by jack-knife.

References

1) *Computers and the Theory of Statistics: Thinking the Unthinkable*, by B. Efron, *SIAM Revirew*, 21 (1979) 460.
2) *The Jackknife– a Review*, by R. G. Miller, *Biometrika*, 61 (1974) 1.
3) *New Monte Carlo Technique for Studying Phase Transitions*, by A. M. Ferrenberg and R. H. Swendsen, *Phys. Rev. Lett.* 61 (1988) 2635.