20.3.1 *Block averages*

We consider a time series of $\mathcal{T}$ measures $\{A_t\}_{t=0,\ldots,\mathcal{T}-1}$ (note we use the indices as in a C array, starting from 0). From this, we want to estimate the true average of the process that generated the data. It is well-known that the best estimator of the true average is the data's average value

$$\langle\langle A\rangle\rangle \equiv \frac{1}{\mathcal{T}}\sum_{t=0}^{\mathcal{T}-1} A_t \ , \quad \text{with error} \quad \sigma_{\langle\langle A\rangle\rangle} = \sqrt{\frac{\mathrm{var}(A_t)}{\mathcal{T}-1}} \ . \tag{20.11}$$

The expression for the uncertainty of the average's estimate $\sigma_{\langle\langle A\rangle\rangle}$ is correct only if the $\mathcal{T}$ measures are fully uncorrelated from each other. With the symbol $\mathrm{var}(A_t)$ we indicate the variance of the data series defined by

$$\mathrm{var}(A_t) \equiv \frac{1}{\mathcal{T}}\sum_{t=0}^{\mathcal{T}-1}\big(A_t - \langle\langle A\rangle\rangle\big)^2 = \frac{1}{\mathcal{T}}\sum_{t=0}^{\mathcal{T}-1} A_t^2 - \langle\langle A\rangle\rangle^2 = \langle\langle A^2\rangle\rangle - \langle\langle A\rangle\rangle^2 \ .$$

Instead, in case the measures are correlated the expression (20.11) underestimates the statistical uncertainty when determining the average, and we need a more refined analysis.

Suppose the measures are correlated with a correlation time τ_A. We now show how to proceed in order to get a correct estimate of the error on the average. We group the measures in blocks containing each b consecutive measures and we call $A_t^{(b)}$ the average of the measurements in the tth block,

$$A_t^{(b)} = \frac{1}{b}\sum_{t'=bt}^{b(t+1)-1} A_{t'} \ .$$

The number of blocks, and therefore, the length of the new data sequence $\{A_t^{(b)}\}$ is $\mathcal{T}/b$.[3] In particular, note how the $A_t^{(1)}$ are the original data and that the average value of the data $\langle\langle A\rangle\rangle$, does not change when these are combined in blocks of equal size (the average value remains the best estimate of the true average).

For $b < \tau_A$, the data of the sequence $\{A_t^{(b)}\}$ are still correlated (with a correlation time of about τ_A/b), while, for $b > \tau_A$, the data become uncorrelated, because the majority of temporal correlations ended up inside the blocks. Let us take advantage of this property in order to correctly estimate the statistical error, $\sigma_{\langle\langle A\rangle\rangle}$, of the average's estimate. We define

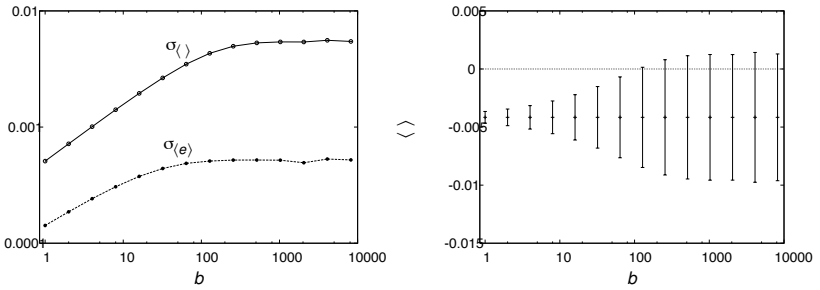
$$\sigma^{(b)}_{\langle\langle A\rangle\rangle} \equiv \sqrt{\frac{\mathrm{var}(A_t^{(b)})}{(\mathcal{T}/b)-1}} \ ,$$

[3] In order to simplify the formulas, we are assuming $\mathcal{T}$ to be multiple of b; otherwise we could have to treat differently the last block, if smaller.

where the variance of the sequence $\{A_t^{(b)}\}$ is defined as before

$$\mathrm{var}(A_t^{(b)}) \equiv \frac{b}{\mathcal{T}} \sum_{t=0}^{\mathcal{T}/b-1} \left(A_t^{(b)}\right)^2 - \langle\langle A \rangle\rangle^2 \, .$$

The $\sigma^{(b)}_{\langle\langle A \rangle\rangle}$ corresponds to the average's uncertainty. It is obtained by assuming the data grouped in blocks of size b are uncorrelated. Because of what we just discussed, we expect $\sigma^{(b)}_{\langle\langle A \rangle\rangle}$ to underestimate the error of the average as long as $b < \tau_A$ and, instead, to provide the correct uncertainty for $b > \tau_A$.



(a) (b)

Fig. 20.5 Uncertainty of the estimate of $\langle e \rangle_T$ and $\langle m \rangle_T$ (a) and the average magnetization with its estimated error (b) as a function of the number b of measures per block.

In Figure 20.5(a) we show the uncertainties on the estimates of the averages $\langle e \rangle_T$ (below) and $\langle m \rangle_T$ (on top) for the two-dimensional Ising model of size $L = 20$ at temperature $T = 2.4$. the data were obtained from a sequence of 10^6 measurements at equilibrium. Note that the data follow the theoretically predicted behavior well. They grow up to a certain value of b, corresponding to the correlation time, after which they stabilize. The value of the *plateau* is the correct error on the average. Also note that the data given in Figure 20.5 fluctuate, given that the variances $\sigma^{(b)}$ are random variables as well.

Having 10^6 available data, we could have combined them in groups using larger values of b than those given in Figure 20.5. This might have obscured our analysis though. Indeed, when the number of blocks becomes too small the value of $\sigma^{(b)}_{\langle\langle A \rangle\rangle}$ possibly fluctuates a lot. Therefore, we advise to use at least about a hundred blocks.

The data presented in Figure 20.5(b) should convince us how important a good estimate of the errors is. The figure contains, for various values of

the block size b, the average value of the magnetization with its error (the model is still the two-dimensional Ising model with $L = 20$ and $T = 2.4$). The important thing to note is that, if we did not correctly estimate the statistical error, i.e., if we had used the average corresponding to $b = 1$, we would have concluded the average value of the magnetization was not compatible with zero (which is a wrong result, causing us to rethink the entire numerical simulation!). Instead, once we estimated the correct error, the average magnetization is compatible with zero, as it should be.

The data given in Figure 20.5 are obtained with the function of Listing 20.1, in which the average and the variance of the data sequence is computed to obtain $\sigma^{(b)}_{\langle\langle A\rangle\rangle}$ (lines 6-8). At the same time, the next sequence of data is generated by grouping the current ones two by two (line 9). The function `binning` receives in input the pointer `data` to the beginning of the array containing the `numMeas` data of which we want to compute the block average. Since the function writes the block averages in the same array (more precisely, in its first half), it is important we pass a pointer to a *copy* of the data, if we do not want to loose the original ones.

```
1 double binning(double *data, int numMeas) {
2   int i, tmp = numMeas / 2;
3   double mean = 0.0, variance = 0.0;
4
5   for (i = 0; i < tmp; i++) {
6     mean += data[2 * i] + data[2 * i + 1];
7     variance += data[2 * i] * data[2 * i] +
8                 data[2 * i + 1] * data[2 * i + 1];
9     data[i] = 0.5 * (data[2 * i] + data[2 * i + 1]);
10  }
11  if (2 * i < numMeas) {
12    mean += data[2 * i];
13    variance += data[2 * i] * data[2 * i];
14  }
15  mean /= numMeas; variance /= numMeas;
16  return sqrt((variance - mean * mean) / (numMeas - 1));
17 }
```

Listing 20.1 The function combining the data in groups to calculate the correct error of the average.

The statements on lines 11-14 are executed only if `numMeas` is odd, as in this case the cycle on lines 5-10 runs along all data except for the last one. In this case, we loose a data item in the new sequence. This is no problem whatsoever, though, as we usually start from very long data sequences.

The following lines of code show a possible way to use the function

`binning`.

```
binSize = 1;
while (numMeas >= 100) {
  printf("%i %lg %lg %lg %lg %i\n",
         binSize, aveM, binning(m, numMeas),
         aveE, binning(e, numMeas), numMeas);
  binSize *= 2;
  numMeas /= 2;
}
```

Before executing these lines of code, the arrays `m` and `e` have been filled with `numMeas` values of the magnetization and the energy. The variables `aveM` and `aveE` contain the averages of these two data sets. The variable `binSize` is the size of the block and `numMeas` represents the length of the current sequence. After each function call to `binning` these variables are updated, respectively, by multiplying and dividing them by 2. The cycle in these lines of code is executed as long as the number of data in the sequence is larger than or equal to 100, such that the variance is a stable measure. If we try to decrease this limit, it is easy to convice ourselves we do not gain any extra information because the measurements of $\sigma^{(b)}_{\langle\langle A\rangle\rangle}$ start to fluctuate a lot.

20.3.2 *Jackknife*

The method described in Section 20.3.1 always allows us to produce data sequences which are practically uncorrelated, given a series of equilibrium measurements. For example, grouping with $b = 10^3$ (this is the value of b at which the plateau starts in Figure 20.5) the 10^6 magnetization measurements, we can obtain 10^3 uncorrelated data.

However, we still need to solve a very common problem in data analysis. Namely, we still need to find a correct estimate of the error for the average of complicated function of the measurements. A concrete example, related to what we saw in Section 20.2.3, is the estimate of the uncertainty when computing the Binder parameter defined in equation (20.10). We can easily evaluate the error on the estimate of $\langle m^2\rangle_T$ and $\langle m^4\rangle_T$ with the method given in Section 20.3.1, using, respectively, $A_t = m_t^2$ and $A_t = m_t^4$ as data sequences. Once the uncertainties $\sigma_{\langle m^2\rangle}$ and $\sigma_{\langle m^4\rangle}$ are known, we could find the one on the estimate of $\langle B(T,L)\rangle_T$ by propagating the errors. This result largely overestimates the true error though. The reason is easy to understand: the second and the fourth moments of the magnetization enter in $B(T,L)$ only through their ratio. So, if the estimate of the numerator $\langle\langle m^4\rangle\rangle$

fluctuates above the true average, also the denominator's estimate $\langle\langle m^2 \rangle\rangle^2$ has a positive fluctuation and the ratio does not change much. Therefore, the fluctuations of the estimate of the ratio are very small compared to those expected from a simple propagation of the errors, which assumes the fluctuations of the numerator and the denominator to be uncorrelated. We show a method allowing to compute the correct uncertainty also if strong correlations such as these are present.

This method, called *jackknife*, is extremely useful and should always be kept close at hand when analyzing data (just like a jackknife, when going camping for example). The jackknife method allows to compute the correct uncertainty on the estimate of the average value of any function of the data. For example, given a series of $\mathcal{T}$ uncorrelated measures of the magnetization, the estimate of the average value of the Binder parameter, defined in (20.10), is given by the function

$$B\big(\{m_t\}\big) = \frac{1}{2}\left(3 - \frac{\mathcal{T}\sum_t m_t^4}{\left(\sum_t m_t^2\right)^2}\right) .$$

It is not clear though, which error we should associate to this estimate.

The jackknife method is based on the following idea. Suppose all measurements are available but the ith one. With these $\mathcal{T}-1$ measures we can still estimate the average value of the function. Typically, this estimate is slightly different from the one obtained with all $\mathcal{T}$ measures and the difference between these two estimates precisely give us the information needed to compute the uncertainty of the average's estimate.

In slightly more formal terms, suppose the best estimate, μ, of a generic quantity is given by the following function of $\mathcal{T}$ *uncorrelated* measures [4],

$$\mu = f\big(\{x_t\}_{t=1,\ldots,\mathcal{T}}\big) .$$

By defining the averages without measure j as

$$\mu_j = f\big(\{x_t\}_{t\neq j}\big) ,$$

we can determine the uncertainty of μ as

$$\sigma_\mu = \sqrt{\frac{\mathcal{T}-1}{\mathcal{T}}\sum_{j=1}^{\mathcal{T}}(\mu_j - \mu)^2} , \tag{20.12}$$

that is the spread of the averages μ_j around μ.

[4] In case the measures are correlated, we first need to group them in blocks of appropriae size.

Exercise 1 - Computation of the average with the jackknife method

In case we want to estimate the sample mean, the best estimator is simply the average value of the measurements

$$\mu = \frac{1}{\mathcal{T}} \sum_{t=1}^{\mathcal{T}} x_t \ .$$

Prove that in this case, the error resulting from the jackknife method, given by the equation (20.12), is identical to the one we obtain by computing the standard deviation of the measurements, as in the expression (20.11).

Creating a code that computes the error with the jackknife method is easy. We simpy need to write two functions, respectively, returning the value of μ and μ_j. Actually, we can write one single function which, called with a parameter $j \in [0, \mathcal{T}-1]$, computes the average without the jth measure, and returns the total average, instead, if $j \notin [0, \mathcal{T}-1]$. The drawback of this approach is the number of operations, as the code would contain $\mathcal{T}+1$ function calls performing $\mathcal{O}(\mathcal{T})$ operations each. Performing a number of operations $\mathcal{O}(\mathcal{T}^2)$ is sometimes very heavy (it is not rare to have 10^8 data and 10^{16} operations are many even for a modern PC) and we need more efficient solutions, whenever possible. For example, in case of the Binder parameter, being the magnetization an integer variable, we can use the function given in Listing 20.2. The latter only takes $\mathcal{O}(2\mathcal{T})$, number of operations because the averages μ_j are computed by subtracting each time a measure from the variables `sumM2` and `sumM4` containing the sums of all measures. In case of floating-point variables, the use of expressions like `sumM4 - tmp * tmp`, where a large quantity `sumM4`) is compared to a small one, is risky and requires at least the use of Kahan's algorithm (described in Section 4.5).

The function `binder` given in Listing 20.2 accepts the pointer `M` to the array containing the magnetization measurements and the number `numMeas` of these measures as arguments. The other two arguments, `pMean` and `pError`, are pointers to the memory locations where the function saves the value of the Binder parameter and its error. The latter is computed with the jackknife method assuming the measures are uncorrelated.

```
void binder(int *M, int numMeas, double *pMean, double *pError) {
  unsigned long long int tmp, sumM2 = 0, sumM4 = 0;
  double reducedMean;
  int i, *pM;

  for (i = 0, pM = M; i < numMeas; i++, pM++) {
    tmp = (*pM) * (*pM);
    sumM2 += tmp;
    sumM4 += tmp * tmp;
  }
  *pMean = 0.5 * (3.0 - (double)numMeas * sumM4 / sumM2 / sumM2);
  *pError = 0.0;
  for (i = 0, pM = M; i < numMeas; i++, pM++) {
    tmp = (*pM) * (*pM);
    reducedMean = 0.5 * (3.0 - (double)(numMeas - 1) *
      (sumM4 - tmp * tmp) / (sumM2 - tmp) / (sumM2 - tmp));
    *pError += (reducedMean - *pMean) * (reducedMean - *pMean);
  }
  *pError = sqrt(*pError * (numMeas - 1) / numMeas);
}
```

Listing 20.2 A function computing the error on the estimate of the Binder parameter with the jackknife method.

Hands on 4 - Error of the Binder parameter with the jackknife

Consider once more the magnetization data obtained when studying the Ising model in 2 and/or 3 dimensions. Combine them in bins of an appropriate size in order to eliminate the correlations. Next, estimate the average value of the Binder parameter and the corresponding error with the jackknife method. Compare this error with the one obtained by the simpler (but less reliable) error propagation.

20.3.3 *Scaling laws and critical exponents*

Very often a Monte Carlo simulation is used to solve a problem with a given number of variables. For example, optimizing a given cost function or computing the average value in a model accurately describing a specific problem. Nevertheless, there exist situations in which the interesting result is obtained in the limit in which the number of variables of the model becomes extremely large (in statistical mechanics formally we consider the thermodynamic limit, $N \to \infty$). Think, for example, to those phenomena in