Estimating errors reliably in Monte Carlo simulations of the Ehrenfest model

Vinay Ambegaokar
Laboratory of Atomic and Solid State Physics, Cornell University, Ithaca, New York 14853

Matthias Troyer
Theoretische Physik, ETH Zurich, 8093 Zurich, Switzerland

(Received 4 June 2009; accepted 23 September 2009)

We use the Ehrenfest urn model to illustrate the subtleties of error estimation in Monte Carlo simulations. We discuss how the smooth results of correlated sampling in Markov chains can fool one's perception of the accuracy of the data and show via numerical and analytical methods how to obtain reliable error estimates from correlated samples. © *2010 American Association of Physics Teachers.*
[DOI: 10.1119/1.3247985]

I. INTRODUCTION AND SUMMARY

The Ehrenfest urn model[1] is sometimes picturesquely described as fleas jumping between dogs. One imagines a system of N numbered fleas residing on dog A or dog B, each jumping from one dog to the other when its number is called. This model has been used to illuminate thermodynamic equilibration and equilibrium.[2–4] Monte Carlo simulations of the process are particularly instructive. One of us noticed,[5] but let pass without investigation, the errors associated with such simulations. This neglect is remedied in the present paper, raising issues known to specialists but perhaps not widely enough appreciated. Despite its importance, the topic is not covered in depth in most text books.[6,7] The tutorial exposition given here may therefore be of general interest.

Although the message of this paper is that single flea hops are an inefficient way of sampling the steady state, the process is ideally suited to understanding thermodynamically irreversible transitions from unlikely to likely configurations,[2–5] as well as fluctuations in equilibrium, which in typical physical situations also proceed with small local changes similar to the flea hops in our model.

The paper is organized as follows. Sec. II contains a brief description of the essence of the Monte Carlo method. Although the procedure is useful in cases where an enumeration of possibilities is prohibitively difficult, the urn model is simple enough to allow explicit analysis. In the main body of the paper we exploit only the fact that the steady state probability of n fleas on dog A is a binomial distribution and use this analysis as a check for various numerical simulations. In Sec. III we show that trials of N-flea configurations yield good results with expected errors. We then simulate the single flea transfer used in Refs. 3–5 and encounter the apparent inaccuracies we have mentioned. In Sec. IV correlations between successive samples and their effect in reducing the number of independent trials are studied, and a numerical method ("binning analysis") is used to illuminate and eliminate the problem, leading to the conclusions in Sec. V. In the Appendix the Markov (that is, memory less) random process underlying single flea transfers is treated analytically using methods similar to those in Ref. 2, revealing nice features of the approach to equilibrium and the autocorrelation problem.

II. THE MONTE CARLO METHOD

It is said that Stanislav Ulam[8] invented the Monte Carlo method in the 1940s when playing Solitaire while lying sick in bed. He wanted to know the probability of winning in Solitaire but was faced with the problem that with $52! \approx 10^{68}$ different ways of arranging the cards, he could not exactly calculate the chance of winning. He realized that by just playing 100 games and counting the number of wins, he could get a pretty good estimate.

This insight suggested a way of tackling the problem caused by the exponential growth with size in the number of states of a statistical system. In a general statistical context, we might wish to calculate weighted averages over configurations. However, even in our simple model the number of ways of distributing fleas between the two dogs is 2^N. These configurations may be denoted by N-dimensional vectors $\vec{x}$ of which each element x_n, $1 \leq n \leq N$, can take on two values. If each configuration is assigned a normalized weight $p(\vec{x})$, $\Sigma_{\vec{x}} p(\vec{x}) = 1$, the weighted mean of an arbitrary function $A(\vec{x})$ is

$$\langle A \rangle \equiv \sum_{\vec{x}} A(\vec{x}) p(\vec{x}). \tag{1}$$

An exact summation over all states is usually impossible for $N > 40$ even on the most powerful supercomputers. The Monte Carlo method,[6,7] which Ulam named after the famous casinos in Monaco,[9] estimates such sums by a partial sum over a sample of only $M \ll 2^N$ configurations $\vec{x}_i$,

$$\bar{A} \equiv \frac{1}{M} \sum_{i=1}^{M} A_i, \tag{2}$$

where the configurations $\vec{x}_i$ are chosen randomly with the correct probability $p(\vec{x})$, and we have introduced the shorthand notation $A_i \equiv A(\vec{x}_i)$.

Choosing the sample randomly and with the correct probabilities is as crucial here as in opinion polls before presidential elections: only a truly random and representative sample will give meaningful results.

The estimate $\bar{A}$ of the true expectation value $\langle A \rangle$ is a fluctuating quantity that will deviate from the true value. According to the central limit theorem, $\bar{A}$ is normally distributed around $\langle A \rangle$ with a standard deviation Δ_A, which we shall calculate in the following.

As a warm-up we show that the expectation value of $\bar{A}$ is $\langle A \rangle$,

$$\langle \bar{A} \rangle = \left\langle \frac{1}{M} \sum_{i=1}^{M} A_i \right\rangle = \frac{1}{M} \sum_{i=1}^{M} \langle A_i \rangle \tag{3a}$$

$$= \frac{1}{M} \sum_{i=1}^{M} \langle A \rangle = \langle A \rangle. \tag{3b}$$

In going from Eq. (3a) to Eq. (3b), we have used the linearity of the expectation value. We also have made use of the fact that the samples $\vec{x}_i$ are all chosen from the same distribution $p(\vec{x})$ so that the A_i's have the expectation value given by Eq. (1).

Similar reasoning allows the calculation of the average of the square of the sample mean,

$$\langle \bar{A}^2 \rangle = \left\langle \left(\frac{1}{M} \sum_{i=1}^{M} A_i \right)^2 \right\rangle = \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \langle A_i A_j \rangle \tag{4a}$$

$$= \frac{1}{M^2} \sum_{i=1}^{M} \langle A_i^2 \rangle + \frac{M-1}{M} \langle A \rangle^2 \tag{4b}$$

$$= \frac{1}{M} \langle A^2 \rangle + \frac{M-1}{M} \langle A \rangle^2, \tag{4c}$$

where we have inserted the definition of the average (2), used the linearity of the expectation value, and also exploited the fact that for independent samples $\vec{x}_i$ and $\vec{x}_j$, the expectation value for $i \neq j$ factorizes as

$$\langle A_i A_j \rangle = \langle A_i \rangle \langle A_j \rangle = \langle A \rangle^2. \tag{5}$$

The statistical error Δ_A, the root-mean-square deviation of the sample mean $\bar{A}$ from the true expectation value $\langle A \rangle$, is thus given by

$$\Delta_A^2 \equiv \langle (\bar{A} - \langle A \rangle)^2 \rangle \tag{6a}$$

$$= \frac{1}{M^2} \sum_{i=1}^{M} \langle A_i^2 \rangle - \frac{1}{M} \langle A \rangle^2 \tag{6b}$$

$$= \frac{1}{M} (\langle A^2 \rangle - \langle A \rangle^2) \equiv \frac{1}{M} \mathrm{Var}\, A, \tag{6c}$$

which is the basis of the central limit theorem. It is, however, more useful to express the error in terms of the sampled A_i's. A naïve guess would be to estimate the variance as $\overline{A^2} - \bar{A}^2$, where

$$\overline{A^2} \equiv \frac{1}{M} \sum_{i=1}^{M} A_i^2. \tag{7}$$

If we calculate the expectation values via Eq. (4c), we obtain

$$\langle \overline{A^2} - \bar{A}^2 \rangle = \frac{M-1}{M} \mathrm{Var}\, A. \tag{8}$$

The true estimator is thus

$$\mathrm{Var}\, A \approx \frac{M}{M-1} (\overline{A^2} - \bar{A}^2), \tag{9}$$

where the (small) fluctuations in the right-hand side of Eq. (9) have been ignored. Taking the square root, we obtain the final result

$$\Delta_A = \sqrt{\frac{\mathrm{Var}\, A}{M}} \approx \sqrt{\frac{\overline{A^2} - \bar{A}^2}{M-1}}. \tag{10}$$

The -1 in the denominator, which is irrelevant for the large values of M in the simulations, reflects the loss of one piece of information in calculating the sample mean.

III. DOGS AND FLEAS

After these preliminaries, let us consider the fleas on two dogs game as played in Refs. 5 and 4. The game starts with two dogs—a flea-ridden dog B (Burnside) with $N=50$ fleas and a clean dog A (Anik). Once per time step a randomly chosen flea hops from one dog to the other so that asymptotically the probability of a flea being on one of the dogs is 1/2. In this simple case it is possible to analytically calculate the probability distribution $P[n]$ of having n of the N fleas on one dog. It is the binomial distribution

$$P_{\mathrm{eq}}[n] = \frac{1}{2^N} \binom{N}{n} = \frac{1}{2^N} \frac{N!}{n!\,(N-n)!}. \tag{11}$$

This exact solution will be very useful as a test for our Monte Carlo simulations.

A. Direct sampling

Our first Monte Carlo simulation does not follow the above game but will directly sample the asymptotic distribution. For each sample, we loop over all fleas and draw a uniformly distributed random binary integer $u \in \{0,1\}$. If $u=0$, the flea is positioned on Anik, otherwise on Burnside. To estimate the distribution $P[n]$ for the number of fleas n on Anik, it is sufficient to record the histogram $H[n]$, which counts how often n fleas are on Anik. From a histogram built from M repetitions, we can compute an estimate for $P[n]$ as

$$\overline{P[n]} = \frac{1 \times H[n] + 0 \times (M - H[n])}{M} = \frac{H[n]}{M} \tag{12}$$

because our estimator is one whenever there were n fleas on Anik and zero otherwise. Because $1^2=1$ and $0^2=0$, we obtain the same estimator for the square,

$$\overline{P[n]^2} = \frac{1^2 \times H[n] + 0^2 \times (M - H[n])}{M} = \frac{H[n]}{M}, \tag{13}$$

from which we obtain the error estimate

$$\Delta_{P[n]} \approx \sqrt{\frac{H[n]/M - H[n]^2/M^2}{M-1}}. \tag{14}$$

In Fig. 1 we compare the exact solution to the Monte Carlo solution for $M=10\,000$ samples and find that, as expected from the normal distribution, the exact solution lies within the error bars about 2/3 of the time. The Monte Carlo simulation is working well!

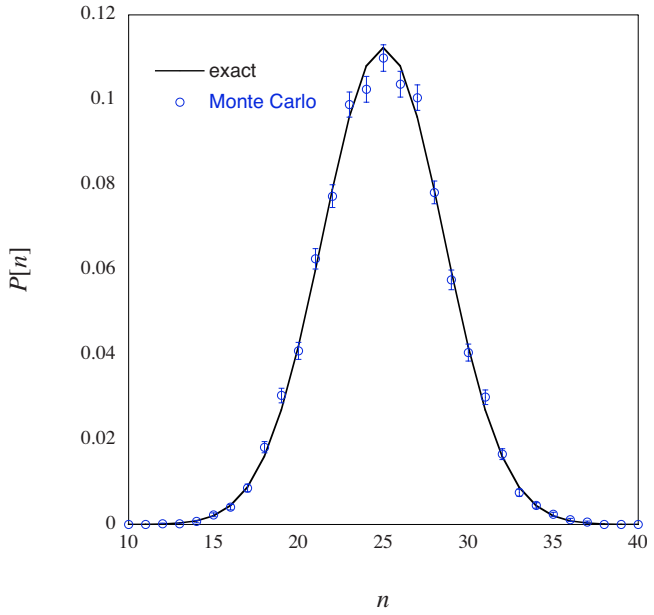

Fig. 1. Comparison of the flea distribution $P[n]$ obtained in a direct Monte Carlo simulation with the exact asymptotic result. $M=10\,000$ samples were recorded.

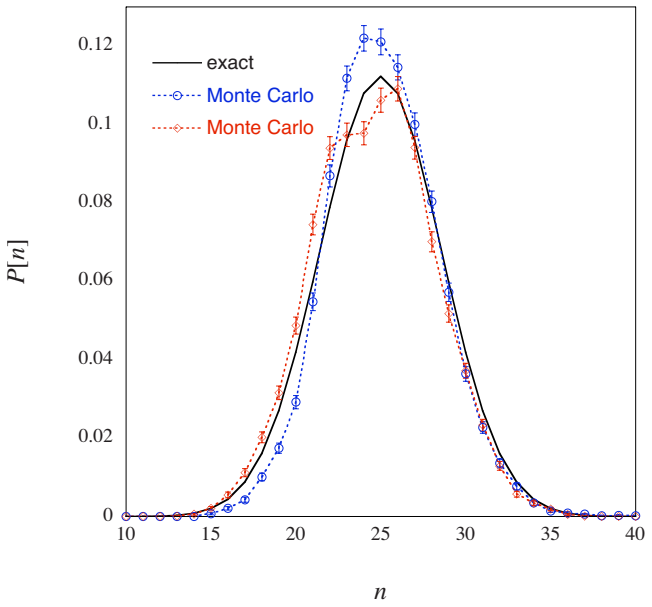

Fig. 2. Comparison of the flea distribution $P[n]$ obtained in Monte Carlo simulations of the original dog and fleas game with the exact asymptotic result. Two different random seeds were used; $M=10\,000$ and $M/5$ steps were used for equilibration. Something is wrong because the exact results are significantly outside the error bars, not even the two simulations agree, and the asymmetric shape cannot be right.

B. The dogs and fleas simulation

We next implement the simulation of the dog and fleas game. We start with all $N=50$ fleas on Burnside and hence $n=0$. In each simulation step we then choose one of the N fleas at random by drawing a uniform integer random number u between 1 and N and move that flea to the other dog. In practice, we label the fleas so that fleas $1,\ldots,n$ are on Anik and fleas $n+1,\ldots,N$ are on Burnside. Hence if $u\leq n$, we move a flea from Anik to Burnside and decrease n by 1; otherwise we move a flea in the opposite way and increase n by 1.

In our simulation we need to wait a while until the distribution of fleas has equilibrated, and we can expect to observe the asymptotic distribution. We thus perform $M/5$ flea hops for equilibration without recording any measurements. Only then do we start with the actual simulation and perform M flea hops, recording the histogram $H[n]$.

In simple examples like this simulation, we might actually be able to guess the number of steps needed for equilibration. As we show in the Appendix, only about 50 hops are needed to reach equilibrium. Why then did we throw away 20%, or 2000 samples? The reason is that in more complex cases we often have no idea of the actual equilibration times. It is then strongly recommended to err on the side of throwing away too many samples rather than too few. By throwing away the first 20% of our samples, we increase our statistical error by only about 10% (remember the inverse square root scaling of the error with the number of samples), which is a small price to pay to be on the safe side regarding equilibration.

In Fig. 2 we compare $P[n]$ to the exact solution and observe deviations that have been noted earlier.[5] At first sight, the deviations are puzzling because the curves look smooth. However, the asymmetric shapes cannot be correct, and the errors bars calculated using Eq. (10) with $M=8000$ are too small. That these features are general can be seen by repeating the simulations with different random seeds. Sometimes the results look mostly right, but often they are just plainly wrong as in Fig. 2. The large variations observed also confirm that something is wrong with the error estimates.

A little further thought suggests the reason. Equation (14) is an estimate for the relative deviation from the mean of M trials of a binomial process with a success probability estimate $P[n]$. But M single flea hops is not the same as M trials of the whole distribution as performed to obtain Fig. 1.

IV. AUTOCORRELATION EFFECTS AND ERROR ESTIMATES

We need to reconsider the derivation of the errors in Eqs. (6a)–(6c) and (7)–(10). The only assumption, besides a finite variance, in Eq. (5) was the independence of samples $\vec{x}_i$ and $\vec{x}_j$ for $i\neq j$. Although this independence is clear in the direct simulation—at least as long as we use independent random numbers to create the flea distributions—it is not true of the original dogs and fleas simulation in which subsequent samples differ only by a single flea. The samples form a "Markov chain." As remarked, this method of sampling explores the space of states much less efficiently than the calculation of Fig. 1 in which every flea is addressed at every trial. Equation (5) and thus also the error estimate (10) are not valid for correlated samples from a Markov chain. The correlation between samples is also responsible for the smooth shape of the results, which fools our intuition about the errors of the results.

In the following we will discuss two methods for obtaining reliable errors of a Monte Carlo simulation.

A. Error estimates from independent simulations

The easiest way of obtaining reliable error estimates is to create independent samples. To obtain them we perform the simulation multiple times with different random seeds. In each simulation we record an estimate for $P[n]$. Then we

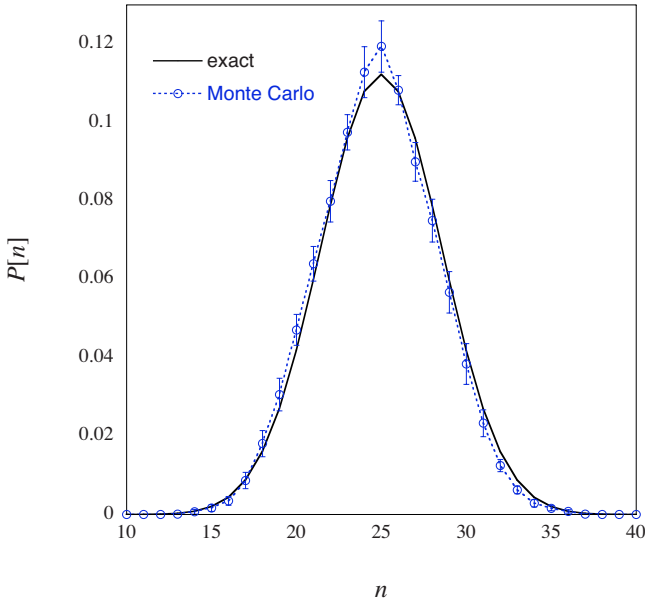

Fig. 3. Comparison of the flea distribution $P[n]$ obtained in Monte Carlo simulations of the original dog and fleas game with the exact asymptotic result. Now $L=10$ independent simulations were performed for a total of $M=10\,000$ measurements. Each simulation performed M/L measurements, and was equilibrated for $M/5$ steps. Now the error bars, estimated from the $L=10$ independent simulations are much larger and agree with the exact result, but at the cost of having to equilibrate L simulations.

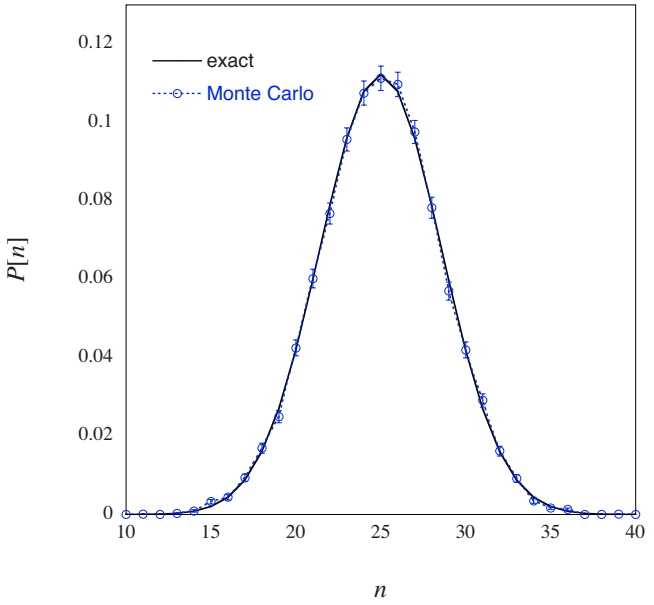

Fig. 4. Comparison of the flea distribution $P[n]$ obtained in Monte Carlo simulations of the original dog and fleas game with the exact asymptotic result. Now $N_{\text{hop}}=99$ flea hops were made between the $M=10\,000$ measurements and $M/5$ steps were used for equilibration. $N_{\text{hop}}=99$ seem to be enough hops to decorrelate the samples and give reliable error estimates.

obtain a final estimate for $P[n]$ by averaging the $P[n]$ obtained in the individual simulations and an error estimate by applying Eq. (14) to the $P[n]$ obtained from these independent simulations.

We show results from performing $L=10$ simulations of $M/L=1000$ measurements each in Fig. 3. Although we still see large deviations and the maximum appears too high as in Refs. 4 and 5, the error bars are now much larger and appear correct—they include the correct value most of the time.

Although performing L independent simulations gives reliable error bars, we pay the price that each of the L simulations needs to be equilibrated independently so that in our case we performed $LM/5=20\,000$ equilibration steps in addition to $M=10\,000$ measurement steps.

B. Error estimates from uncorrelated samples

Another way of obtaining reliable errors is not to measure after every flea hop but to let many fleas hop before performing a measurement. In Fig. 4 we show the results from a simulation performing $N_{\text{hop}}=99$ flea hops[10] between each of the $M=10\,000$ measurements. Now the Monte Carlo results agree with the exact results, and the error bars are again much smaller but at the cost of having to perform $N_{\text{hop}}=99$ times more flea hops and also losing all of the potentially useful information between measurements. In addition, we have no way of knowing whether $N_{\text{hop}}=99$ hops between measurements are sufficient to create uncorrelated samples for which Eq. (5) holds, or whether a much smaller suffices or a much larger number is needed.

C. Error estimates for correlated samples

The previous analysis clearly demonstrates that the correlation between samples $\vec{x}_i$ and $\vec{x}_j$ is the origin of our problems. Instead, let us correct the error estimate Eq. (10) for the case of correlated samples by including the terms that we omitted under the assumption of independence (5) to obtain

$$\Delta_A^2=\frac{\text{Var}\,A}{M}+\frac{1}{M^2}\sum_{i\neq j=1}^{M}(\langle A_iA_j\rangle-\langle A\rangle^2). \tag{15}$$

Previously we had assumed that due to independence the second term is zero. Let us now replace the assumption of independence by a rapid decay as $|i-j|\to\infty$ (Ref. 11) and rewrite the second term as

$$\frac{1}{M^2}\sum_{i\neq j=1}^{M}(\langle A_iA_j\rangle-\langle A\rangle^2)=\frac{2}{M^2}\sum_{i<j=1}^{M}(\langle A_iA_j\rangle-\langle A\rangle^2) \tag{16a}$$

$$=\frac{2}{M^2}\sum_{i=1}^{M}\sum_{t=1}^{M-i}(\langle A_iA_{i+t}\rangle-\langle A\rangle^2) \tag{16b}$$

$$=\frac{2}{M}\sum_{t=1}^{M-1}(\langle A_1A_{1+t}\rangle-\langle A\rangle^2) \tag{16c}$$

$$\approx\frac{2}{M}\sum_{t=1}^{\infty}(\langle A_1A_{1+t}\rangle-\langle A\rangle^2) \tag{16d}$$

$$\equiv\frac{2}{M}(\text{Var}\,A)\,\tau_A. \tag{16e}$$

To go from the right-hand side of Eq. (16a) to Eq. (16b), we relabeled the indices. In Eq. (16c) we used the identical distributions to limit the sum over i to the first index; in Eq. (16d) we extended the sum over t to infinity because the correlations are expected to decay fast enough; and in

Eq. (16e) we used the definition of the integrated autocorrelation time τ_A of A,

$$\tau_A \equiv \frac{\sum_{t=1}^{\infty}(\langle A_1 A_{1+t}\rangle - \langle A\rangle^2)}{\langle A^2\rangle - \langle A\rangle^2}. \tag{17}$$

We substitute Eq. (16e) into Eq. (15) and obtain the final error estimate

$$\Delta_A = \sqrt{\frac{\mathrm{Var}\, A}{M}(1+2\tau_A)} \tag{18}$$

and see that due to correlation effects, the error is increased by a factor of $\sqrt{1+2\tau_A}$. Equation (18) gives the effective number of uncorrelated samples as $[M/(1+2\tau_A)]<M$.

Although this equation explains the failures of the simple error estimate (10), it does not help us much because the estimation of τ_A via Eq. (17) is expensive and cumbersome. A fast and easy way of estimating errors is explained in the following, and an exact calculation of the autocorrelation time for this model is presented in the Appendix.

D. Error estimates from a binning analysis

The binning analysis is a method of analyzing Monte Carlo data based on Eq. (18). It provides both an estimate for the error Δ_A and for the integrated autocorrelation time τ_A. Starting from the original series of measurements $A_i^{(0)}=A_i$, we iteratively create "binned" series by averaging over two consecutive entries,

$$A_i^{(l)} := \tfrac{1}{2}(A_{2i-1}^{(l-1)} + A_{2i}^{(l-1)}) \tag{19}$$

for $i=1,\ldots,M_l \equiv M/2^l$.

Every entry in this new and shorter time series is the average of two adjacent values in the original one. The mean of the new binned time series is the same as the original time series. The averaged values are, however, less correlated than the original ones. The (incorrect) error estimates using Eq. (10) for uncorrelated samples gives errors,

$$\Delta_A^{(l)} \approx \sqrt{\frac{1}{M_l(M_l-1)}\sum_{i=1}^{M_l}(A_i^{(l)} - \overline{A^{(l)}})^2}, \tag{20}$$

which increase as a function of bin size 2^l. These errors converge to the correct error estimate

$$\Delta_A = \lim_{l\to\infty} \Delta_A^{(l)} \tag{21}$$

when the bins become uncorrelated for sizes $2^l \gg \tau_A$.

It is best to stop this binning procedure when the number of bins is still larger than 30, which statisticians consider a large sample. Below that the sampling errors may become too big.

This binning analysis thus gives a reliable recipe for estimating errors and autocorrelation times. One has to calculate the error estimates for different bin sizes l and check if they converge to a limiting value. If convergence is observed, the limit Δ_A is a reliable error estimate, and τ_A can be obtained from Eq. (18) as

$$\tau_A = \frac{1}{2}\left[\left(\frac{\Delta_A}{\Delta_A^{(0)}}\right)^2 - 1\right]. \tag{22}$$

If no convergence of the $\Delta_A^{(l)}$ is observed, we know that τ_A is longer than the simulation time, and we have to perform much longer simulations to obtain reliable error estimates. This is illustrated in Fig. 5.

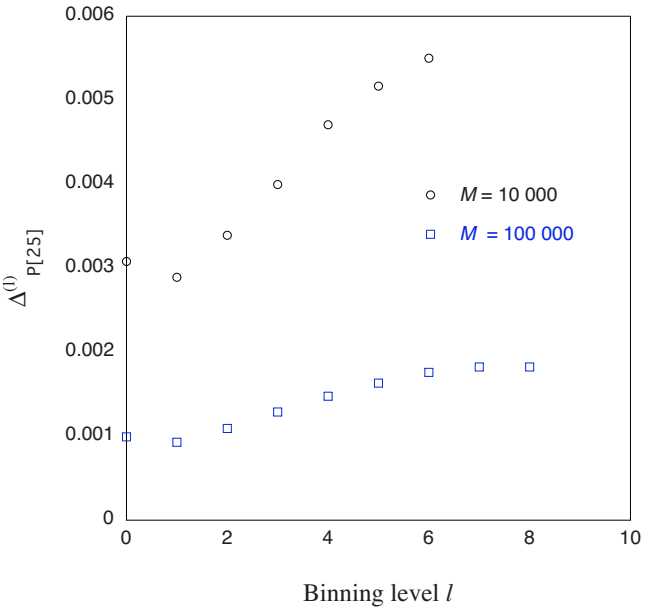

Fig. 5. Binning analysis of the error $\Delta_{P[25]}$ of the central value $P[25]$ of the distribution. It is seen that for $M=10\,000$ samples, the errors have not yet converged, while for $M=100\,000$ samples, convergence starts to be seen. At least $M=100\,000$ samples have to be taken to get reliable results.

Let us redo one of the simulations in Sec. III B and perform a binning analysis. In Fig. 6 we show our results for $M=100\,000$ measurements, calculating the errors using the binning analysis. Now everything is consistent.

Note that the autocorrelation time depends on the variable being sampled. For example, calculating this quantity for the number n of fleas yields 24.0, which is larger than the value obtained from Fig. 5 for the peak of the histogram.

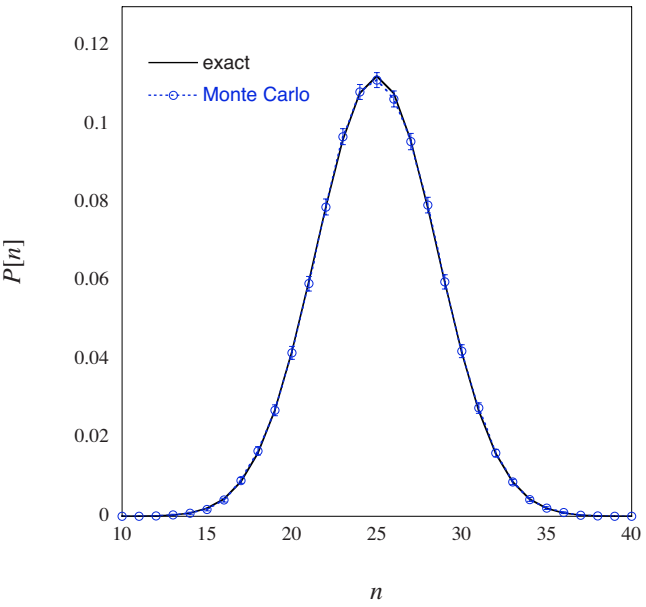

Fig. 6. Comparison of the flea distribution $P[n]$ obtained in Monte Carlo simulations of the original dog and fleas game with the exact asymptotic result. This time $M=100\,000$ correlated measurements were taken, and the errors were calculated using a binning analysis.

To implement the binning analysis, it is not necessary to store the full time series. Instead, the memory of $2\log_2 M$ numbers is sufficient. Interested readers are encouraged to look at the implementation in the file src/alps/alea/simplebinning.h of the ALPS libraries.[12]

V. CONCLUSIONS: LESSONS LEARNED

In the discussion of the dogs and fleas simulation, we have seen some of the subtleties and pitfalls in estimating reliable errors for results of Monte Carlo simulations. Correlation effects make it necessary to perform a binning analysis instead of using Eq. (10), which is valid only for independent samples.

We have not touched on the issue of cross-correlations between different quantities that influence error estimates, for example, the specific heat $C_V=(\langle E^2\rangle-\langle E\rangle^2)/k_BT^2$. To calculate such errors a bootstrap or jackknife method[13] is required in addition to a binning analysis.

An important lesson learned is that a reliable analysis of errors of a simulation can be much more difficult than performing the simulation but is an essential part for any numerical project. We might have drawn incorrect conclusions and conjectured a new physical phenomenon based on our too small error bars.

We have also seen that using improved methods, such as the direct sampling of the distribution in Fig. 1, smaller errors and more reliable results can be obtained. Unfortunately direct sampling is impossible in all but the simplest models—but improved algorithms are the key to reliable large scale simulations. It is interesting that over the past three decades progress in algorithms for the simulation of the Ising model has outperformed Moore's law. Running modern algorithms on 30 year old computers would be faster than running 30 year old algorithms on the fastest supercomputers of today![14]

All of the programs used to produce the data in this paper are included in the example/sampling directory of the latest release of the ALPS libraries.[12]

ACKNOWLEDGMENTS

V.A. thanks his colleague Erich Mueller for suggesting the generating function method used in the Appendix and Cornell graduate students Frank Petruzielo, Bryan Daniels, and Turan Birol for computer instruction. M.T. acknowledges support of the Aspen Center for Physics.

APPENDIX: EXACT CALCULATION OF THE EQUILIBRATION AND AUTOCORRELATION TIMES

The evolution of the number of fleas on Anik in our Monte Carlo simulation is done probabilistically using a Markov process. Let $P_i[n], n=0,1,\ldots N$, be the ith update of the probability of n fleas on Anik. Then[2,4,5]

$$P_{i+1}[n]=\frac{N-n+1}{N}P_i[n-1]+\frac{n+1}{N}P_i[n+1] \tag{A1a}$$

$$=\frac{1}{N}\sum_{n'=0}^{N}T[n,n']P_i[n'], \tag{A1b}$$

where the coefficients are the relative probabilities for a flea to hop on or off Anik, written in the last line in terms of a $(N+1)\times(N+1)$ tridiagonal matrix T with the entries $N,N-1,N-2,\ldots,2,1$ on the subdiagonal, $1,2,3\ldots,N-1,N$ on the superdiagonal, and zero elsewhere,

$$T\equiv\begin{pmatrix}0 & 1 & 0 & \ldots & 0 & 0 & 0\\ N & 0 & 2 & \ldots & 0 & 0 & 0\\ 0 & N-1 & 0 & \ldots & 0 & 0 & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots\\ 0 & 0 & 0 & \ldots & 0 & N-1 & 0\\ 0 & 0 & 0 & \ldots & 2 & 0 & N\\ 0 & 0 & 0 & \ldots & 0 & 1 & 0\end{pmatrix}. \tag{A2}$$

The $N+1$ eigenvalues λ and right eigenvectors $r[n]$ of this matrix can be obtained from the generating function

$$f(u,v)=\sum_{n=0}^{N}u^nv^{N-n}r[n]. \tag{A3}$$

When used with the eigenvalue equation $\Sigma_{n'}T[n,n']r[n']=\lambda r[n]$, f is seen to obey the differential equation

$$\lambda f=\left[u\frac{\partial}{\partial v}+v\frac{\partial}{\partial u}\right]f, \tag{A4}$$

whose solution in the required form $f(u,v)=v^Nh(u/v)$ is

$$f_\lambda=K_\lambda(v+u)^{(N+\lambda)/2}(v-u)^{(N-\lambda)/2}, \tag{A5}$$

where K_λ is independent of u and v. The series in u and v must terminate, requiring the two exponents in Eq. (A5) to be non-negative integers, thus implying that the eigenvalues are $\pm N,\pm(N-2),\ldots,\pm 1$ (for odd N) or zero (for even N).

The left (dual) eigenvectors $l[n]$ are generated by

$$g(u,v)=\sum_{n=0}^{N}u^nv^{N-n}\binom{N}{n}l[n]. \tag{A6}$$

The coefficient is the combinatorial coefficient defined in Eq. (11). When used with the eigenvalue equation $\Sigma_{n'}l[n']T[n',n]=\lambda l[n]$, g is seen to obey the identical differential equation as f, namely, Eq. (A4). The constants in the solution Eq. (A5) determine the normalization. One choice is to take $K_\lambda=1/2^N$ for f_λ and

$$K_\lambda=\binom{N}{\frac{N+\lambda}{2}} \tag{A7}$$

for g_λ, whereupon $r_N[n]$ is given by $P_{\mathrm{eq}}[n]$ in Eq. (11), the stationary normalized solution of Eq. (A1b), and $l_N[n]=1$ for every n. That this choice also achieves the completeness relation for orthonormal eigenvectors,

$$\sum_\lambda r_\lambda[n]l_\lambda[n']=\delta_{n,n'}, \tag{A8}$$

can be seen from Eqs. (A3), (A5), and (A6).

These considerations facilitate analysis of the approach to equilibrium. The initial condition of flealess Anik may be written using Eq. (A8) as

$$P_0[n] = \delta_{n,0} = \sum_\lambda r_\lambda[n] l_\lambda[0], \tag{A9}$$

where after t hops Eq. (A1b) yields

$$P_t[n] = \left(\frac{T}{N}\right)^t \sum_\lambda r_\lambda[n] l_\lambda[0] = \sum_\lambda \left(\frac{\lambda}{N}\right)^t r_\lambda[n] l_\lambda[0]. \tag{A10}$$

The moments of this distribution may now be calculated. If we compare partial derivatives with respect to u in Eqs. (A3) and (A5), we find

$$\sum_n n r_\lambda[n] = \frac{N}{2}\delta_{\lambda,N} - \frac{1}{2}\delta_{\lambda,N-2}, \tag{A11}$$

and

$$\sum_n n(n-1) r_\lambda[n] = \frac{N(N-1)}{4}\delta_{\lambda,N} - \frac{(N-1)}{4}\delta_{\lambda,N-2} + \frac{1}{2}\delta_{\lambda,N-4}. \tag{A12}$$

Because $l_\lambda[0]$ can be seen to be equal to the K_λ in Eq. (A7), we deduce that

$$\sum_n n P_t[n] \equiv N\mu(t) = \frac{N}{2}\left[1 - \left(1 - \frac{2}{N}\right)^t\right] \tag{A13a}$$

$$\to \frac{N}{2}[1 - e^{-2t/N}], \tag{A13b}$$

showing that the mean number of fleas approaches equal partitioning exponentially with an equilibration time $N/2$. The decay as $(1-(2/N))^t = \lambda_2^t$ is a general result: in any Markov process the equilibration is controlled asymptotically by the second largest eigenvalue λ_2.

Similarly, it is seen using Eqs. (A11) and (A12) that the mean square fluctuation in the number at time step t is given within the exponential approximation of Eq. (A13b) by

$$\sum_n (n - \mu(t))^2 P_t[n] = N\mu(t)(1 - \mu(t)). \tag{A14}$$

Equation (A14) shows that the relation between the mean and width of a binomial distribution for the probabilities associated with tossing a biased coin is preserved during stages of the evolution long before equilibrium is reached.

These methods also permit the exact calculation of the integrated autocorrelation time τ_A defined in Eqs. (16) and (17) for this simple model. As an example, we consider the number n of fleas on Anik and calculate the corresponding autocorrelation time τ_n. We need to calculate the average $\langle n'n \rangle$, where n' is the number of fleas a given number of hops later than an n-flea state. For t hops, this average is

$$C_t \equiv \sum_{n,n'} n' \frac{T^t[n',n]}{N^t} n P_{\text{eq}}[n], \tag{A15}$$

where T is given in Eq. (A2) and P_{eq} is the equilibrium distribution in Eq. (11). In Eq. (A15), n is picked at random from the known correct distribution, and n' is correlated with n via the conditional probability for t hops.

Now T can be represented in terms of its eigenvalues and eigenvectors as

$$T[n',n] = \sum_\lambda r_\lambda[n'] \lambda l_\lambda[n], \tag{A16}$$

and it follows using the orthonormality relation $\Sigma_n l_\lambda[n] r_{\lambda'}[n] = \delta_{\lambda,\lambda'}$ that

$$\frac{T^t[n',n]}{N^t} = \sum_\lambda r_\lambda[n'] \left(\frac{\lambda}{N}\right)^t l_\lambda[n]. \tag{A17}$$

The contribution of the highest eigenvalue $\lambda = N$ to this sum, obtained from the eigenvectors given above Eq. (A8), is independent of n and equal to $P_{\text{eq}}[n']$ for any t. This convenient fact leads to the identity

$$\frac{T^t[n',n]}{N^t} - P_{\text{eq}}[n'] = \sum_{\lambda \neq N} r_\lambda[n'] \left(\frac{\lambda}{N}\right)^t l_\lambda[n]. \tag{A18}$$

If the right-hand side of Eq. (A18) is substituted into Eq. (A15), we encounter the average given in Eq. (A11) and also the average

$$\sum_n n P_{\text{eq}}[n] l_\lambda[n] = \binom{N}{\frac{N+\lambda}{2}} \left[\frac{N}{2}\delta_{\lambda,N} - \frac{1}{2}\delta_{\lambda,N-2}\right], \tag{A19}$$

which has been evaluated via a partial derivative with respect to u of the generating function g in Eq. (A6). Thus only the eigenvalue $\lambda = N-2$ contributes to the simple result

$$\sum_{t=1}^{\infty} [C_t - \langle n \rangle^2] = \sum_{\lambda \neq N} \sum_{n,n'} n' r_\lambda[n'] \frac{\lambda/N}{1 - \lambda/N} l_\lambda[n] n P_{\text{eq}}[n] \tag{A20a}$$

$$= \binom{N}{N-1} \frac{1 - 2/N}{2/N} \left(\frac{1}{2}\right)^2 \tag{A20b}$$

$$= \frac{N(N-2)}{8}. \tag{A20c}$$

Because the equilibrium variance of n is $N/4$, we see by comparison with Eq. (17) that the integrated autocorrelation time for sampling N fleas one at a time is

$$\tau_n(N) = \frac{N-2}{2} \tag{A21}$$

so that M single hops are equivalent to only

$$M_{\text{eff}} = \frac{M}{2\tau_n + 1} = \frac{M}{N-1} \tag{A22}$$

trials of N-flea configurations. In our simulation of $N=50$ fleas, we determined $\tau_n = 24.0$ using the binning analysis, in perfect agreement with the prediction $\tau_n = (N-2)/2 = 24$.

It is gratifying that these learned considerations show that randomizing the number of "heads" among N coins by arbitrarily choosing and turning over one is $1/(N-1)$ times as effective as tossing all N at once. As mentioned in Sec. I, it is the less efficient processes that are typically at work in physical situations.

[1] P. Ehrenfest and T. Ehrenfest, "Über zwei bekannte Einwände gegen das Boltzmannsche H-theorem," Phys. Z. **8**, 311–314 (1907).
[2] M. Kac, "Random walk and the theory of Brownian motion," Am. Math. Monthly **54**, 369–391 (1947). Reprinted in Mark Kac, *Probability, Number Theory and Statistical Physics, Selected Papers*, edited by K. Baclawski and D. Donsker (MIT Press, Cambridge, MA, 1979), pp. 240–262.
[3] G. Emch and C. Liu, *The Logic of Thermostatic Physics* (Springer, Berlin, 2000), pp. 106–112.
[4] V. Ambegaokar and A. Clerk, "Entropy and time," Am. J. Phys. **67**, 1068–1073 (1999).
[5] V. Ambegaokar, *Reasoning About Luck: Probability and Its Uses in Physics* (Cambridge U. P., Cambridge, 1996), p. 174.
[6] An exception is H. Gould, J. Tobochnik, and W. Christian, *Introduction to Computer Simulation Methods*, 3rd ed. (Pearson/Addison-Wesley, San Francisco, 2007), p. 427.
[7] For text books on the Monte Carlo method, see D. P. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics* (Cambridge U. P., Cambridge, 2005); W. Krauth, *Statistical Mechanics: Algorithms and Computations* (Oxford U. P., Oxford, 2006); M. E. J. Newman and G. T. Barkema, *Monte Carlo Methods in Statistical Physics* (Oxford U. P., Oxford, 1999).
[8] R. Eckhardt, "Stan Ulam, John von Neumann, and the Monte Carlo method," Los Alamos Sci. **15**, 125–130 (1987).
[9] If the method had been invented only recently, the name would most likely have been "Las Vegas method."
[10] An odd number N_{hop} is important because n is even after an even number of flea hops and odd after an odd number. Choosing an even value of N_{hop}, we would record only even entries for $P[n]$.
[11] In the Appendix we explicitly calculate the autocorrelation function for this model and show that it decays exponentially.
[12] A. F. Albuquerque *et al.*, "The ALPS project release 1.3: Open-source software for strongly correlated systems," J. Magn. Magn. Mater. **310**, 1187–1193 (2007); F. Alet *et al.*, "The ALPS project: Open source software for strongly correlated systems," J. Phys. Soc. Jpn. **74**, 30–35 (2005). The source code can be obtained from ⟨alps.comp-phys.org/⟩.
[13] See, for example, W. H. Press, S. A. Teukolsky, W. A. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing* (Cambridge U. P., Cambridge, 1992), Chap. 15.6.
[14] D. P. Landau (personal communication).

Superb Calculation of the Pressure in a Fluid
Bob Panoff, Shodor Foundation
Tune: Supercalifragilisticexpialidocious

Bernoulli knew he had a rule he used for wings in air
For fluid incompressible he'd never have a scare.
The density of energy's the same at every spot
A caveat is cavitation in which case it's not!

Oh, Superb calculation of the pressure in a fluid
Is simple so that anyone with any sense can do it.
We all deserve a force conserved among the objects paired.
Just add to pressure rho gee aitch then add half rho vee squared

A water tower tower's o'er a town so water goes
Through every pipe, and when you turn the faucet on it flows.
The pressure head is now instead a steady stream, you see,
The pipe's diameter determines stream velocity.

The sum at every point's a constant, check it if you care
Each term can change within a range for water or for air.
The key's to keep the units straight and don't have any gap
Or else your fluid starts to leak and then you'll just get Oh... .