HPCSE II

Markov chain Monte Carlo

Review:
The Metropolis Algorithm

Statistical mechanics and the Boltzmann weight

- At a fixed temperature *T* the average of a physical observable *A* can be calculated as a sum over all configurations *c*

$$\langle A \rangle = \frac{1}{Z} \sum_c A_c \exp(-\beta E_c)$$

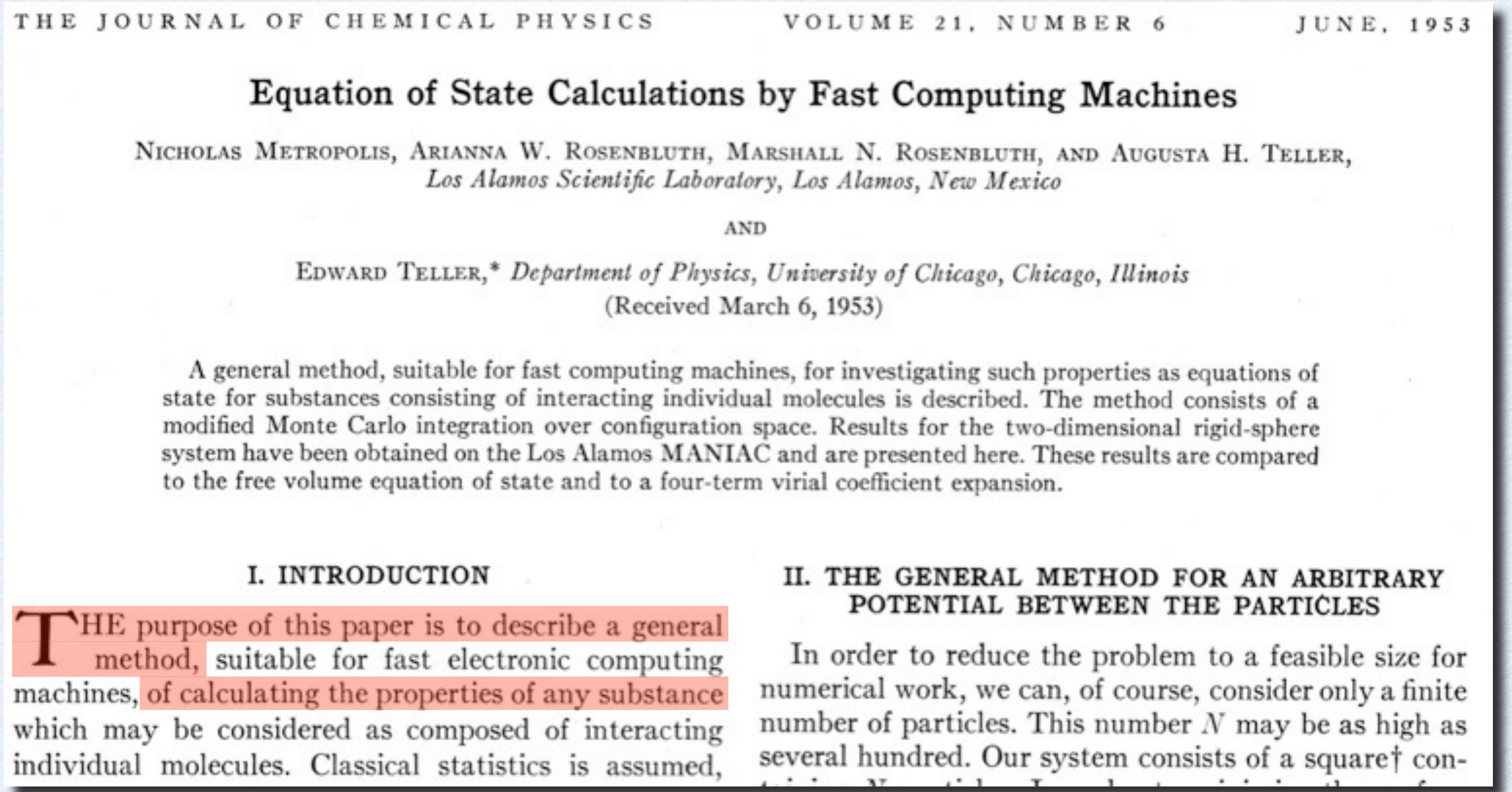
where

c	configuration
E_c	energy of a configuration
A_c	value of the observable for a configuration
T	temperature
$\beta = \frac{1}{k_B T}$	inverse temperature
$Z = \sum_c \exp(-\beta E_c)$	partition function (normalization)

- This is ideal for importance sampling with the Boltzmann weight

$$p_c = \frac{1}{Z} \exp(-\beta E_c)$$

The Metropolis Algorithm (1953)



THE JOURNAL OF CHEMICAL PHYSICS VOLUME 21, NUMBER 6 JUNE, 1953

Equation of State Calculations by Fast Computing Machines

NICHOLAS METROPOLIS, ARIANNA W. ROSENBLUTH, MARSHALL N. ROSENBLUTH, AND AUGUSTA H. TELLER,
Los Alamos Scientific Laboratory, Los Alamos, New Mexico

AND

EDWARD TELLER,* *Department of Physics, University of Chicago, Chicago, Illinois*

(Received March 6, 1953)

A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional rigid-sphere system have been obtained on the Los Alamos MANIAC and are presented here. These results are compared to the free volume equation of state and to a four-term virial coefficient expansion.

I. INTRODUCTION

THE purpose of this paper is to describe a general method, suitable for fast electronic computing machines, of calculating the properties of any substance which may be considered as composed of interacting individual molecules. Classical statistics is assumed,

II. THE GENERAL METHOD FOR AN ARBITRARY POTENTIAL BETWEEN THE PARTICLES

In order to reduce the problem to a feasible size for numerical work, we can, of course, consider only a finite number of particles. This number N may be as high as several hundred. Our system consists of a square† con-

Markov chain Monte Carlo

- Instead of drawing independent samples c_i we build a Markov chain

$$c_1 \to c_2 \to \ldots \to c_i \to c_{i+1} \to \ldots$$

- Transition probabilities $W_{x,y}$ for transition $x \to y$ need to satisfy:
  - **Normalization:** $$\sum_y W_{x,y} = 1$$
  - **Ergodicity:** any configuration reachable from any other

$$\forall x,y \; \exists n \; : \; \left(W^n\right)_{x,y} \neq 0$$

  - **Balance:** the distribution should be stationary
    - change in distribution in one step: $$p_y^{(n+1)} = \sum_x W_{x,y} p_x^{(n)}$$
    - stationarity condition: $$p_y^{(n+1)} = p_y^{(n)} \Rightarrow p_y = \sum_x W_{x,y} p_x$$
  - Detailed balance is sufficient but not necessary for balance

$$\frac{W_{x,y}}{W_{y,x}} = \frac{p(y)}{p(x)}$$

The Metropolis algorithm

- Teller's proposal was to use rejection sampling:
    - Propose a change with an a-priori proposal rate $A_{x,y}$
    - Accept the proposal with a probability $P_{x,y}$
    - The total transition rate is $W_{x,y} = A_{x,y}\, P_{x,y}$
- The choice

$$P_{x,y} = \min\left[1, \frac{A_{y,x} p_y}{A_{x,y} p_x}\right]$$

satisfies detailed balance and was first proposed by Metropolis *et al*

Sampling N-body states

- **Simple sampling**
  - draw random configurations and calculate their energy and weight
  - measure: $\langle A \rangle \approx \sum_i A_i \exp(-\beta E_i) \Big/ \sum_i \exp(-\beta E_i)$
  - problem: we will never hit low-energy configurations (e.g. a crystal)!
- **Importance sampling by Markov chains**
  - Start from a suitable initial condition, e.g. a perfect crystal
  - Then do the following updates:
    - choose a random particle
    - choose a random direction and distance, e.g. by Gaussian distribution with sensible parameters
    - Accept/reject with Boltzmann weight and Metropolis sampling
    - Measure $\langle A \rangle \approx \frac{1}{N} \sum_{i=1}^{N} A_i$
    - Exercise: proof detailed balance and implement this multi-threaded

Monte Carlo Error Analysis

The dogs & fleas model

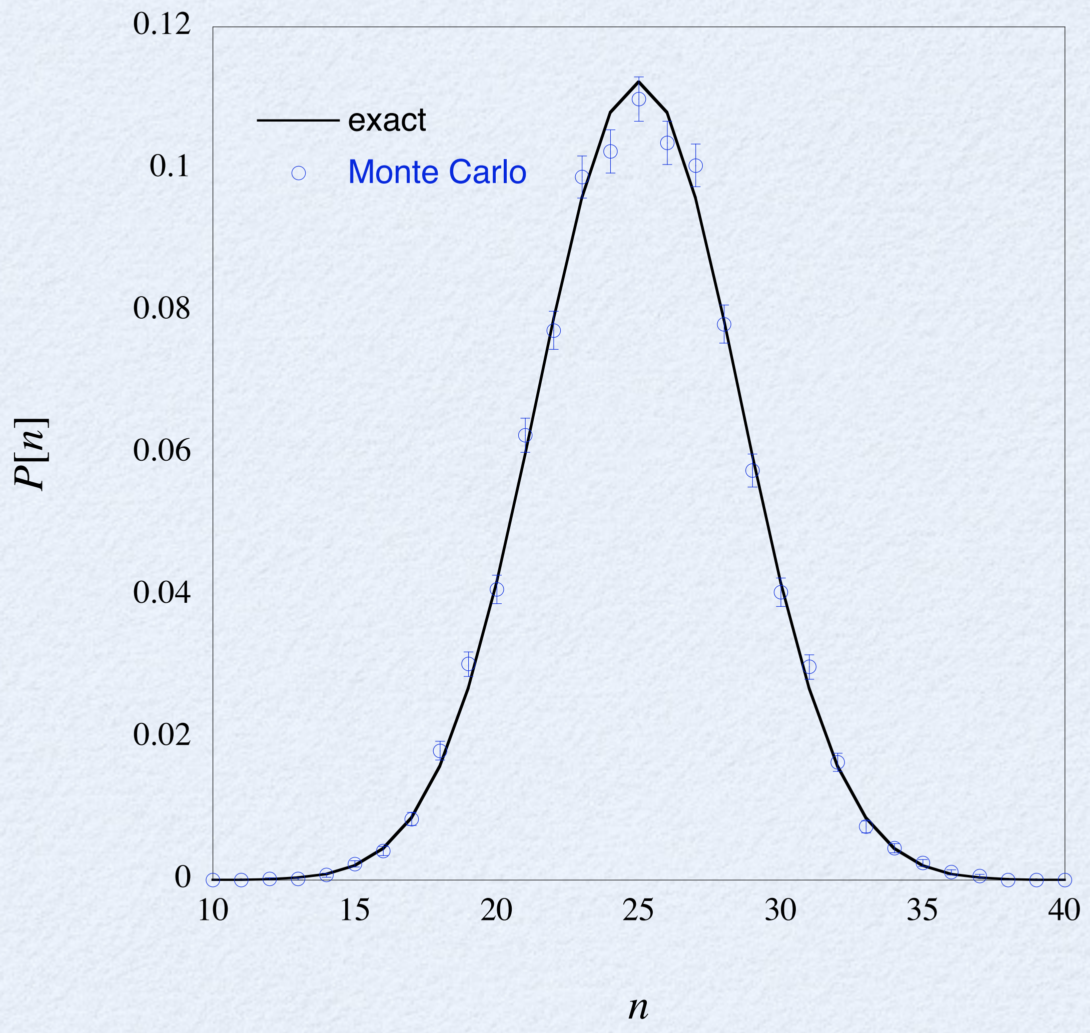
- Two dogs play:
  - Anick has 50 fleas
  - Burnside has no fleas

- During play fleas jump from one dog to the other
- What is the distribution of fleas after they played?

- Vinay Ambegaokar and Matthias Troyer
  American Journal of Physics **78**, 150 (2010)

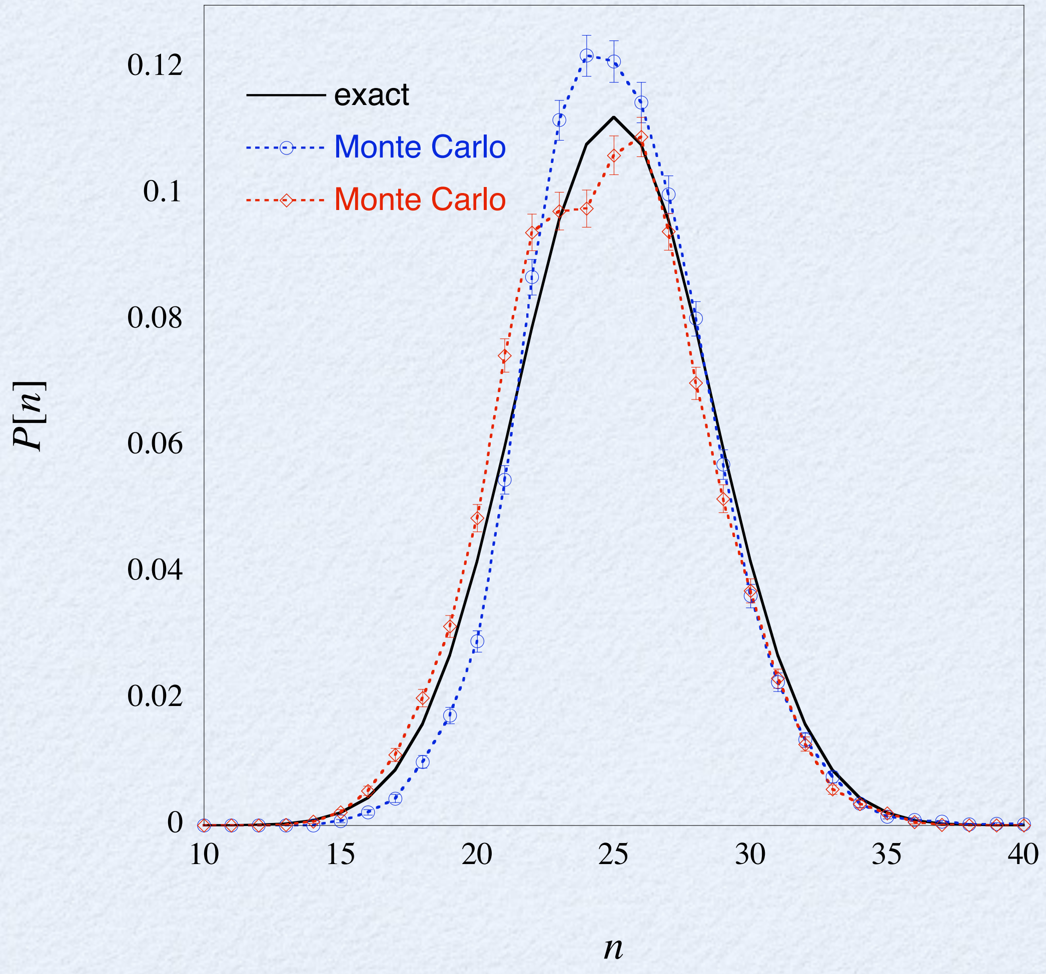
Dogs and fleas: direct sampling

- We pick a dog for each flea: direct sampling as done last week



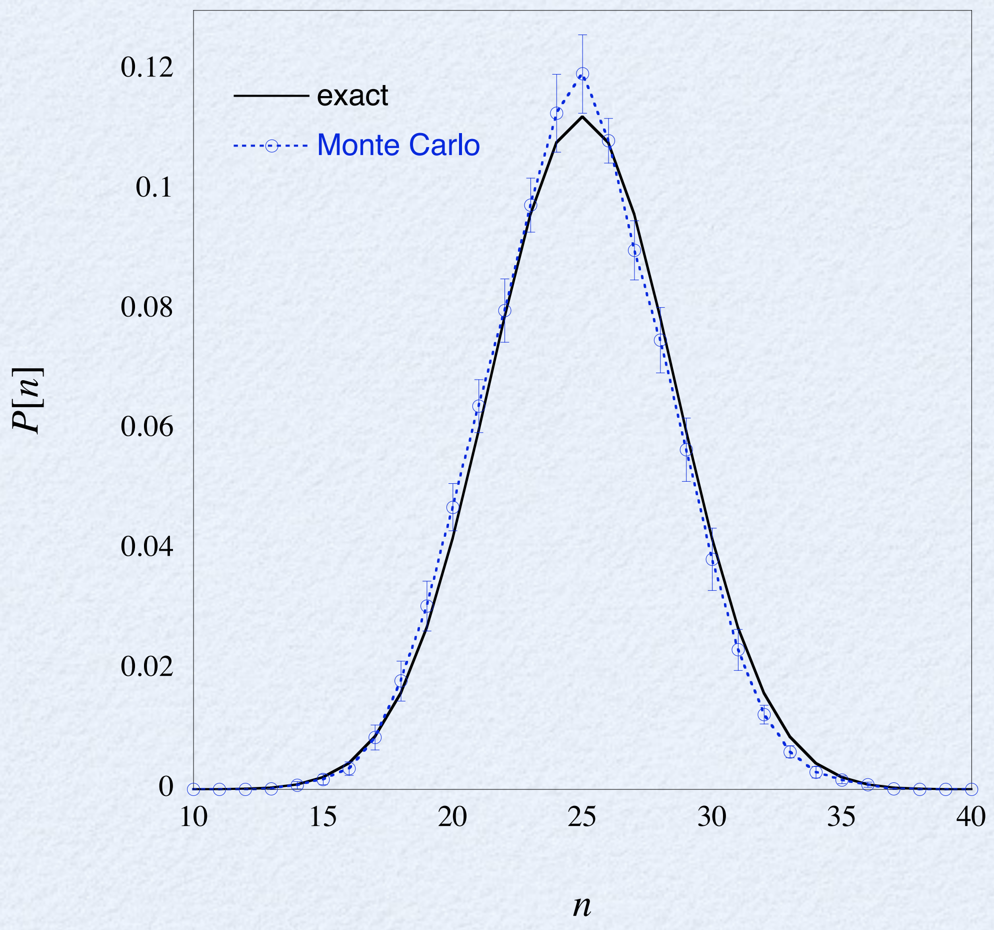
Dogs and fleas: MCMC with naïve errors

- MCMC: pick a flea and let it jump to the other dog
- estimate errors using the standard equation



Dogs and fleas: uncorrelated samples

- One flea hop does not change much: the results are correlated,
- Measure not after every flea hop but only after a few hundred hops



Recall: estimating the error

- The sampling error is the rms (root mean square) deviation

$$(\Delta X)^2 = E\left[\left(\bar{X} - E[X]\right)^2\right] = E\left[\left(\frac{1}{N}\sum_{i=1}^{N}(X_i - E[X])\right)^2\right]$$

$$= E\left[\frac{1}{N^2}\sum_{i,j=1}^{N}(X_i - E[X])\left(X_j - E[X]\right)\right]$$

$$= \frac{1}{N^2}\sum_{i,j=1}^{N}\left(E\left[X_i X_j\right] - E[X]^2\right)$$

$$= \frac{1}{N^2}\sum_{i=1}^{N}\left(E\left[X_i^2\right] - E[X]^2\right) = \frac{1}{N}\left(E\left[X^2\right] - E[X]^2\right) = \frac{\operatorname{Var} X}{N}$$

- We used that samples are uncorrelated:

$$E\left[X_i X_j\right] = E\left[X_i\right]E\left[X_j\right] \text{ for } i \neq j$$

Recall: estimating the error

- Now include correlations:

$$
\begin{aligned}
(\Delta X)^2 &= \frac{1}{N^2}\sum_{i,j=1}^{N}\left(E\left[X_i X_j\right] - E[X]^2\right) \\
&= \\
&= \frac{\operatorname{Var} X}{N} + \frac{1}{N^2}\sum_{i\neq j}\left(E\left[X_i X_j\right] - E[X]^2\right) \\
&= \frac{\operatorname{Var} X}{N} + \frac{2}{N^2}\sum_{i=1}^{N}\sum_{t}\left(E\left[X_i X_{i+t}\right] - E[X]^2\right) \\
&\equiv \frac{\operatorname{Var} X}{N}\left(1 + 2\tau_X\right)
\end{aligned}
$$

- where we defined the integrated autocorrelation time as

$$
\tau_X = \frac{\sum_{t}\left(E\left[X_i X_{i+t}\right] - E[X]^2\right)}{\operatorname{Var} X}
$$

Binning analysis

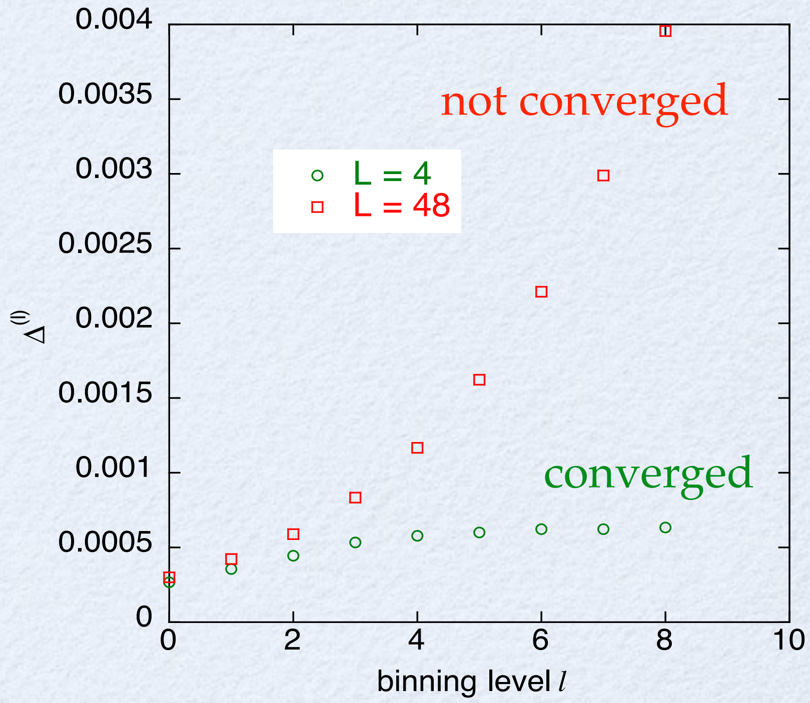
- Take averages of consecutive measurements: averages become less correlated and naive error estimates converge to real error

$$\begin{tikzcd}
A_1 \arrow[d] & A_2 \arrow[dl] & A_3 \arrow[d] & A_4 \arrow[dl] & A_5 \arrow[d] & A_6 \arrow[dl] & A_7 \arrow[d] & A_8 \arrow[dl] & A_9 \arrow[d] & A_{10} \arrow[dl] & A_{11} \arrow[d] & A_{12} \arrow[dl] & A_{13} \arrow[d] & A_{14} \arrow[dl] & A_{15} \arrow[d] & A_{16} \arrow[dl] \\
A_1^{(1)} \arrow[d] & & A_2^{(1)} \arrow[dll] & & A_3^{(1)} \arrow[d] & & A_4^{(1)} \arrow[dll] & & A_5^{(1)} \arrow[d] & & A_6^{(1)} \arrow[dll] & & A_7^{(1)} \arrow[d] & & A_8^{(1)} \arrow[dll] & \\
A_1^{(2)} \arrow[d] & & & & A_2^{(2)} \arrow[dllll] & & & & A_3^{(2)} \arrow[d] & & & & A_4^{(2)} \arrow[dllll] & & & \\
A_1^{(3)} & & & & & & & & A_2^{(3)} & & & & & & &
\end{tikzcd}$$

$$A_i^{(l)} = \frac{1}{2}\left(A_{2i-1}^{(l-1)} + A_{2i}^{(l-1)}\right)$$

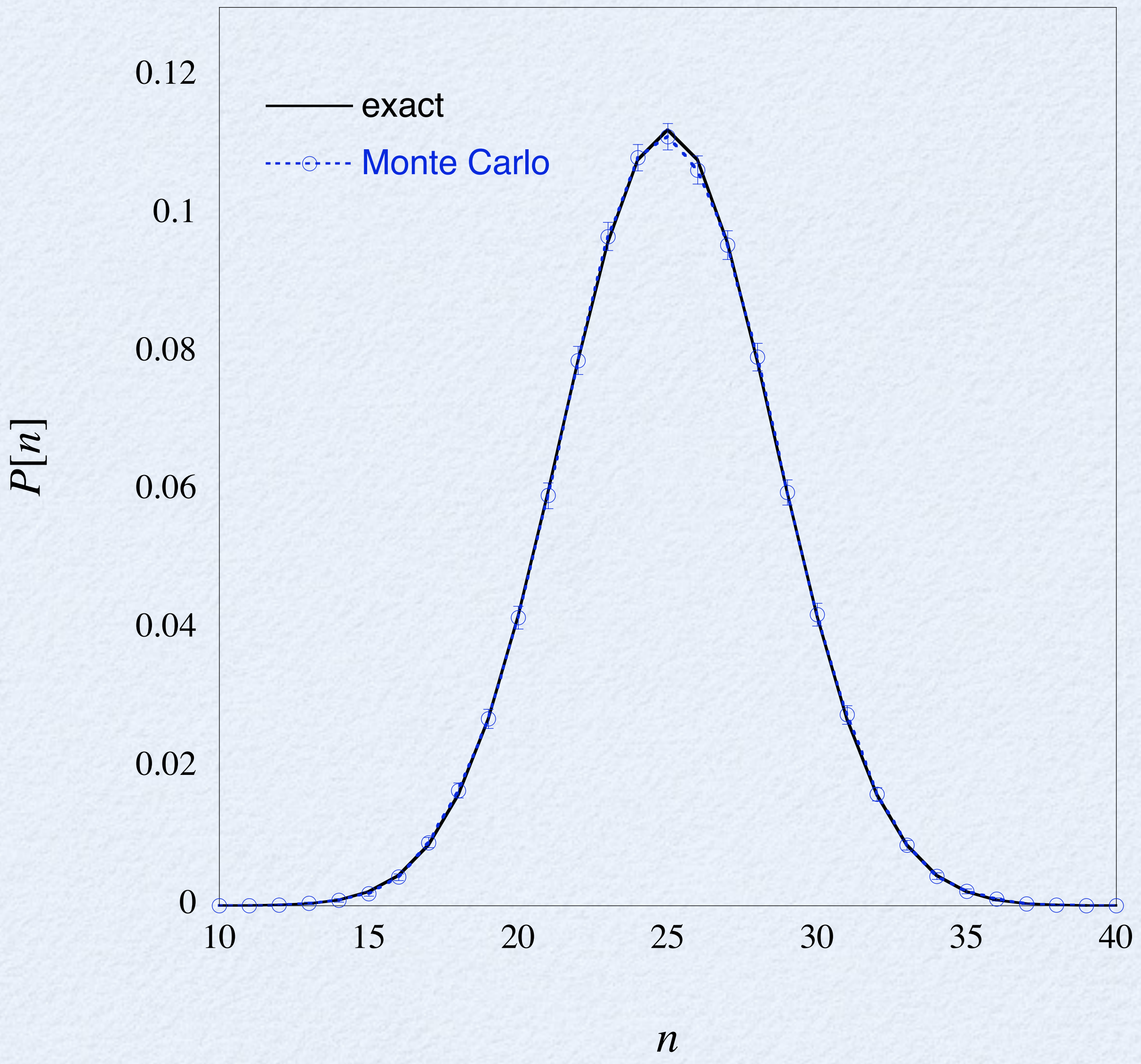
$$\Delta^{(l)} = \sqrt{\mathrm{Var}\,A^{(l)}/M^{(l)}} \xrightarrow{l\to\infty} \Delta = \sqrt{(1+2\tau_A)\,\mathrm{Var}\,A/M}$$

$$\tau_A = \lim_{l\to\infty}\frac{1}{2}\left(\frac{2^l\,\mathrm{Var}\,A^{(l)}}{\mathrm{Var}\,A^{(0)}} - 1\right)$$



a smart implementation needs only O(log(N)) memory for N measurements

Dogs and fleas: binning analysis

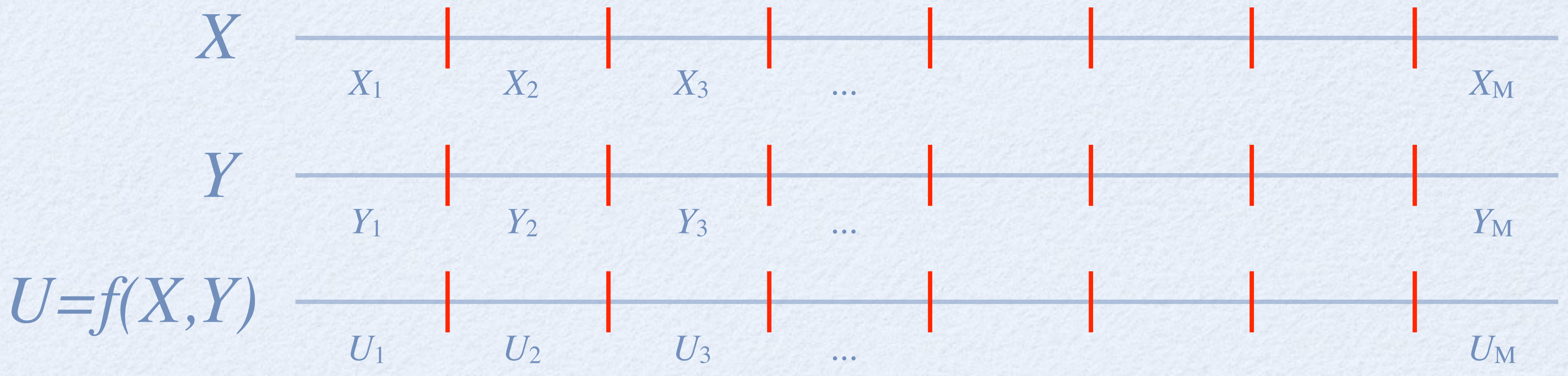


Correlated quantities

- How do we calculate the errors of functions of correlated measurements?
    - specific heat
      $$c_V = \frac{\langle E^2\rangle - \langle E\rangle^2}{k_B T^2}$$
    - Expectation values of weighted samples in direct sampling
      $$\langle A\rangle = \frac{\left\langle \sum_c A_c \exp(-\beta E_c))\right\rangle}{\left\langle \sum_c \exp(-\beta E_c)\right\rangle}$$
- The naïve way of assuming uncorrelated errors is wrong!
- It is not even enough to calculate all crosscorrelations due to nonlinearities except if the errors are tiny!

Splitting the time series

Simplest idea: split the time series and evaluate for each segment



$$\langle U \rangle \approx \overline{U} = \frac{1}{M}\sum_{i=1}^{M} U_i$$

$$\Delta U \approx \sqrt{\frac{1}{M(M-1)}\sum_{i-1}^{M}\left(U_i - \overline{U}\right)^2}$$

Problem: can be unstable and noisy for nonlinear functions such as X/Y

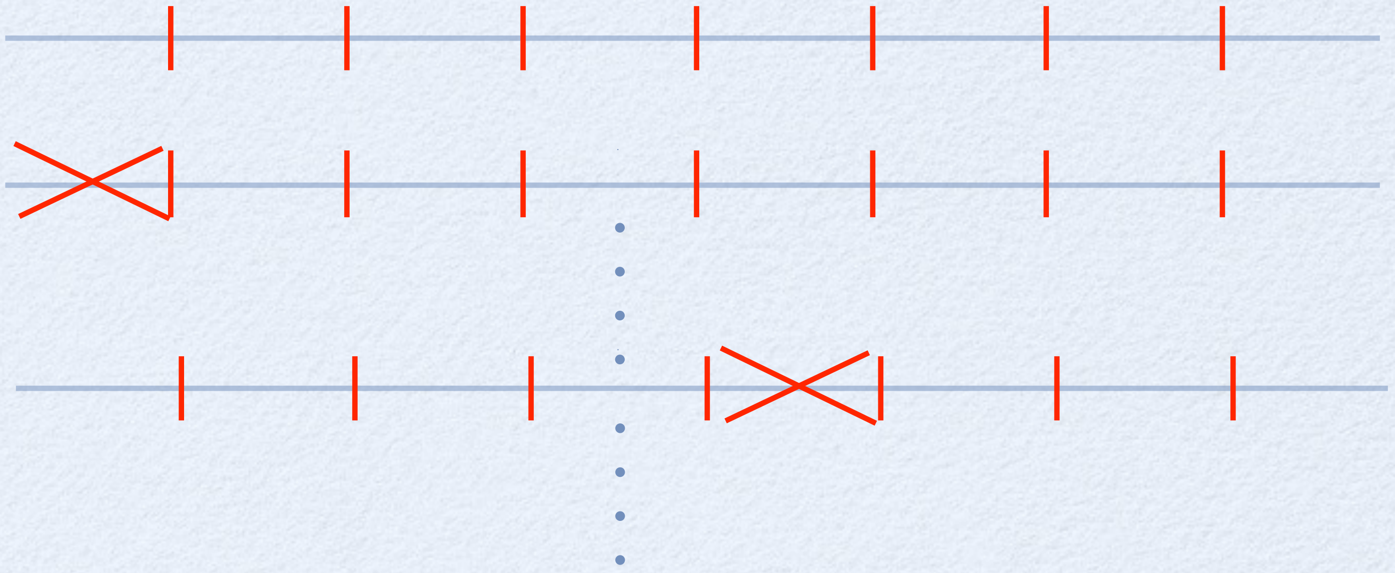
Jackknife-analysis

Evaluate the function on all and all but one segment

$$U_0 = f\left(\frac{1}{M}\sum_{i=1}^{M} X_i, \frac{1}{M}\sum_{i=1}^{M} Y_i\right)$$

$$U_1 = f\left(\frac{1}{M-1}\sum_{i=2}^{M} X_i, \frac{1}{M-1}\sum_{i=2}^{M} Y_i\right)$$

$$U_j = f\left(\frac{1}{M-1}\sum_{\substack{i=1\\ i\neq j}}^{M} X_i, \frac{1}{M-1}\sum_{\substack{i=1\\ i\neq j}}^{M} Y_i\right)$$



$$\langle U\rangle \approx U_0 - (M-1)(\overline{U} - U_0)$$

$$\overline{U} = \frac{1}{M}\sum_{i=1}^{M} U_i$$

$$\Delta U \approx \sqrt{\frac{M-1}{M}\sum_{i-1}^{M}\left(U_i - \overline{U}\right)^2}$$

Analyzing parallel MC simulations

- The error analysis depends on the parallelization strategy

- Parallelization of a single Markov chain, e.g. by multi-threading the energy evaluation
  - Use the binning analysis to calculate errors of the measurements
  - Use jackknife to calculate means and errors of functions of the measurements

- Parallelization by launching multiple independent Markov chains
  - Run a single Markov chain to calculate the autocorrelation time using the binning analysis and then choose a good distance (number of updates) between measurements. A distance comparable to the autocorrelation time is ideal.
  - Then run a parallel simulation and store only the mean for each Markov chain
  - Calculate the overall mean and its error from the mean values of each Markov chain using the simple error formula for independent measurements
  - Use jackknife to calculate means and errors of functions of the measurements