JAN GUKELBERGER

UNIVERSITÉ DE SHERBROOKE

MONTE CARLO METHODS

AN INTRODUCTION

ACKNOWLEDGEMENT

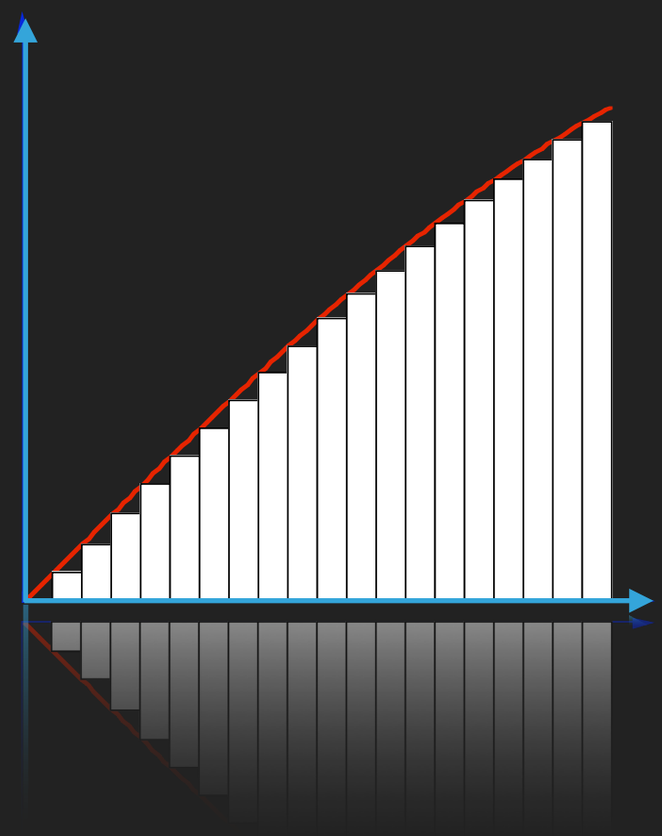
- Slides adapted from
  - Matthias Troyer
  - Werner Krauth

MONTE CARLO INTEGRATION

INTEGRATING A FUNCTION

- convert the integral to a discrete sum

$$\int_a^b f(x)dx = \frac{b-a}{N}\sum_{i=1}^{N} f\left(a + i\frac{b-a}{N}\right) + O(1/N)$$



- higher order integrators
  - trapezoidal rule

$$\int_a^b f(x)dx = \frac{b-a}{N}\left[\frac{1}{2}f(a) + \sum_{i=1}^{N-1} f\left(a + i\frac{b-a}{N}\right) + \frac{1}{2}f(b)\right] + O(1/N^2)$$

  - Simpson rule

$$\int_a^b f(x)dx = \frac{b-a}{3N}\left[f(a) + \sum_{i=1}^{N-1}(3-(-1)^i) f\left(a + i\frac{b-a}{N}\right) + f(b)\right] + O(1/N^4)$$

HIGH-DIMENSIONAL INTEGRALS

- Simpson rule with M points per dimension
  - in one dimension the error is $O(M^{-4})$
  - in d dimensions we need $N=M^d$ points: the error is $O(M^{-4})=O(N^{-4/d})$
- an order-n scheme in one dimension is order-n/d in d dimensions!
- statistical mechanics with N particles requires $6N$-dimensional integrals ($3N$ positions + $3N$ momenta)
- integration becomes extremely inefficient: curse of dimensionality

ULAM: THE MONTE CARLO METHOD

- What is the probability to win in Solitaire?
- Ulam: play it 100 times, count the number of wins and you have a pretty good estimate!

wikipedia.org/wiki/Microsoft_Solitaire

MONTE CARLO INTEGRATION

- consider an integral $\langle f \rangle = \dfrac{\int_\Omega f(\vec{x}) d\vec{x}}{\int_\Omega d\vec{x}}$
- instead of evaluating it at equally spaced points evaluate it at *M* points x_i randomly chosen in Ω

$$\langle f \rangle \approx \frac{1}{M} \sum_{i=1}^{M} f(\vec{x}_i)$$

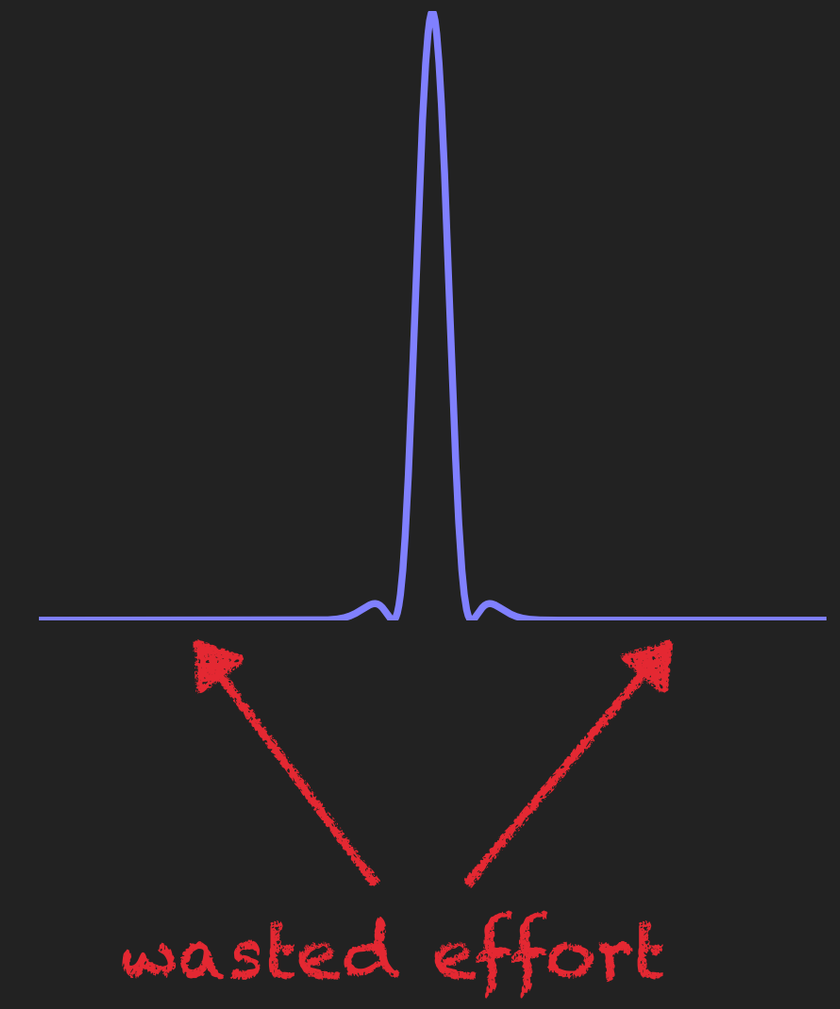
- the error is statistical: $\Delta = \sqrt{\dfrac{\mathrm{Var} f}{M}} \propto M^{-1/2}$

$$\mathrm{Var} f = \langle f^2 \rangle - \langle f \rangle^2$$

- in *d*>*8* dimensions Monte Carlo is better than Simpson!

SHARPLY PEAKED FUNCTIONS

- in many cases the integrand is large only in a tiny region
- lots of time wasted in regions where integrand is small
- sampling error is large because variance is large

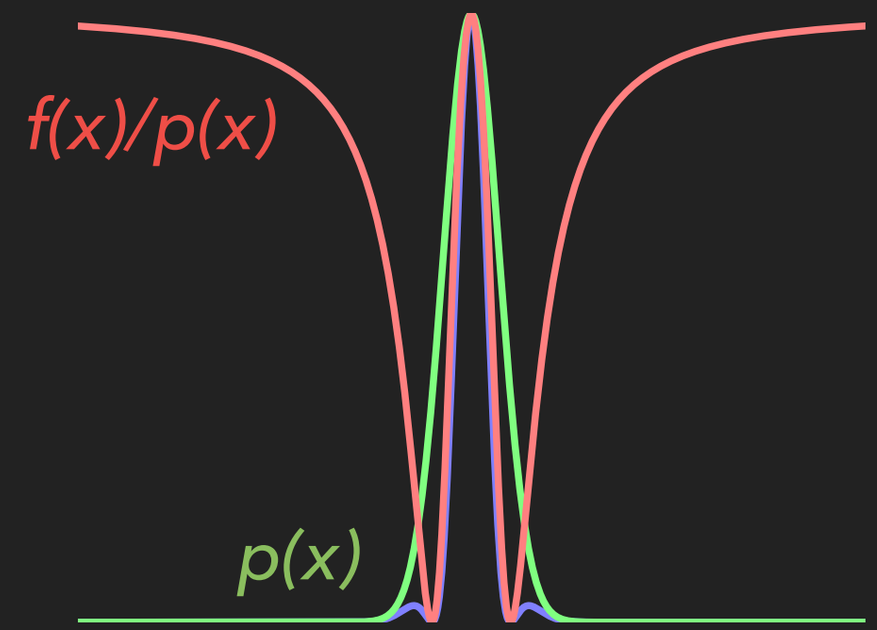


IMPORTANCE SAMPLING

- choose points with probability *p(x)* instead of uniformly

$$\langle f \rangle = \left\langle \frac{f}{p} \right\rangle_p \equiv \int_\Omega \frac{f(\vec{x})}{p(\vec{x})} p(\vec{x}) d\vec{x} \Big/ \int_\Omega d\vec{x}$$

- the error is now determined by $\mathrm{Var}\, f/p$
- find *p* similar to *f* and such that *p*-distributed random numbers are easily available



RANDOM NUMBERS

RANDOM NUMBERS

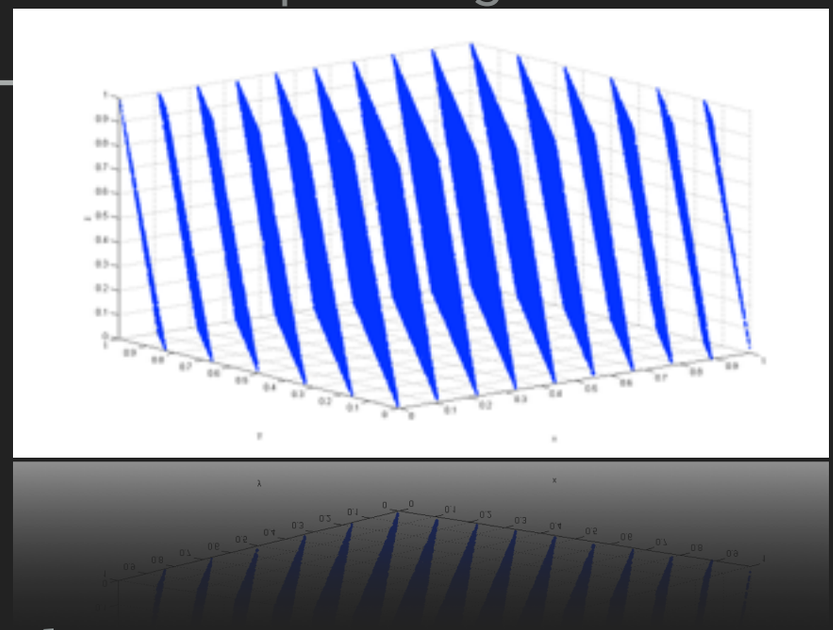
- real random numbers are hard to obtain
  - classical chaos (atmospheric noise)
  - quantum mechanics
- commercial quantum random number generators
  - based on photons and semi-transparent mirrors
  - 4 Mbit/s from a USB device, too slow for most MC simulations

idquantique.com

PSEUDO RANDOM NUMBERS

- are generated by an algorithm
- not random at all, but completely deterministic
- look nearly random though when algorithm is not known
- may be good enough for our purpose
- never trust pseudo-random numbers however!

wikipedia.org/wiki/RANDU



LINEAR CONGRUENTIAL GENERATORS

- are of the simple form $x_{n+1} = f(x_n)$
- e.g. the GGL generator $x_{n+1} = (ax_n + c) \mod m$ with a=16807, c=0, m=2^{31}-1
- quality depends sensitively on a, c, m
- periodicity is a problem with such 32-bit generators
  - sequence repeats identically after 2^{31}-1 iterations
  - with 500 million numbers per second that is just 4 seconds!
- should not be used any more!

LAGGED FIBONACCI GENERATORS

$$x_n = x_{n-p} \otimes x_{n-q} \mod m$$

- good choices are
  - (2281, 1252, +)
  - (9689, 5502, +)
  - (44497, 23463, +)
- seed blocks usually generated by linear congruential
- very long periods due to large block of seeds
- very fast: vectorizes and pipelines well

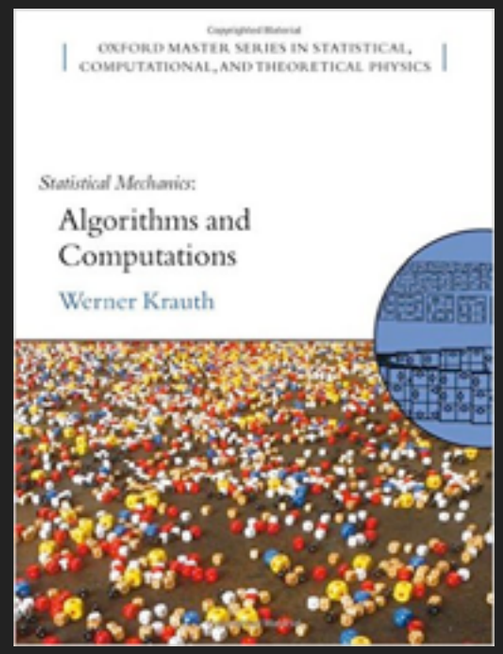
MORE ADVANCED GENERATORS

- as well-established generators fail new tests, better and better generators are developed
  - Mersenne twister (Matsumoto & Nishimura 1997)
  - Well generator (Panneton & L'Ecuyer 2004)
- based on lagged Fibonacci generators, improved with random bit shuffles
- deep number theory enters design of these generators

ARE THESE NUMBERS REALLY RANDOM?

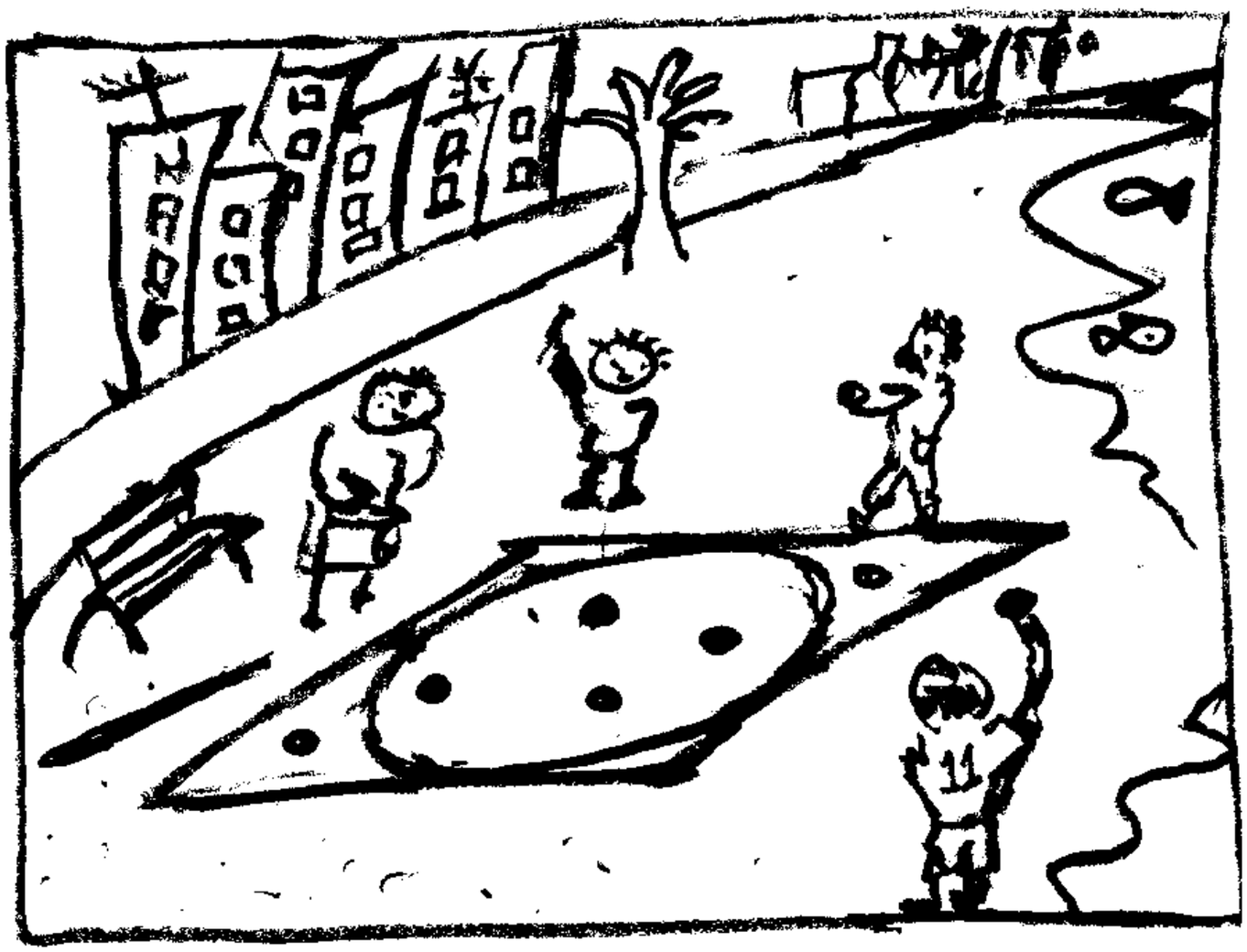
- No!
- Are they random enough?
  - Maybe?
- Statistical tests for distribution and correlations
  - Are these tests enough?
    - No! Your calculation could depend in a subtle way on hidden correlations!
- What is the ultimate test?
  - Run your simulation with various random number generators and compare the results

Werner Krauth:
Statistical Mechanics
– Algorithms and
Computations



CALCULATING Π IN MONACO

CALCULATE Π BY THROWING PEBBLES INTO A CIRCLE (DIRECT SAMPLING)



procedure `direct-pi`

$N_{\text{hits}} \leftarrow 0$ (initialize)

for $i = 1, \ldots, N$ **do**

$$\begin{cases} x \leftarrow \texttt{ran}[-1, 1] \\ y \leftarrow \texttt{ran}[-1, 1] \\ \textbf{if } (x^2 + y^2 < 1)\ N_{\text{hits}} \leftarrow N_{\text{hits}} + 1 \end{cases}$$

output N_{hits}

Five trials with $N = 4000$

run	N_{hits}	estimation
1	3156	3.156
2	3129	3.129
3	3154	3.154
4	3134	3.134
5	3148	3.148

CALCULATING Π IN CANADA (DIRECT SAMPLING)

A Ballistic Monte Carlo Approximation of π

Vincent Dumoulin*
Département d'informatique et de recherche opérationnelle
Université de Montréal

Félix Thouin†
Département de physique
Université de Montréal

(Dated: April 10, 2014)

We compute a Monte Carlo approximation of π using importance sampling with shots coming out of a Mossberg 500 pump-action shotgun as the proposal distribution. An approximated value of 3.131 is obtained, corresponding to a 0.33% error on the exact value of π. To our knowledge, this represents the first attempt at estimating π using such method, thus opening up new perspectives towards computing mathematical constants using everyday tools.

Keywords: shotgun, π, Monte Carlo, importance sampling

I. INTRODUCTION

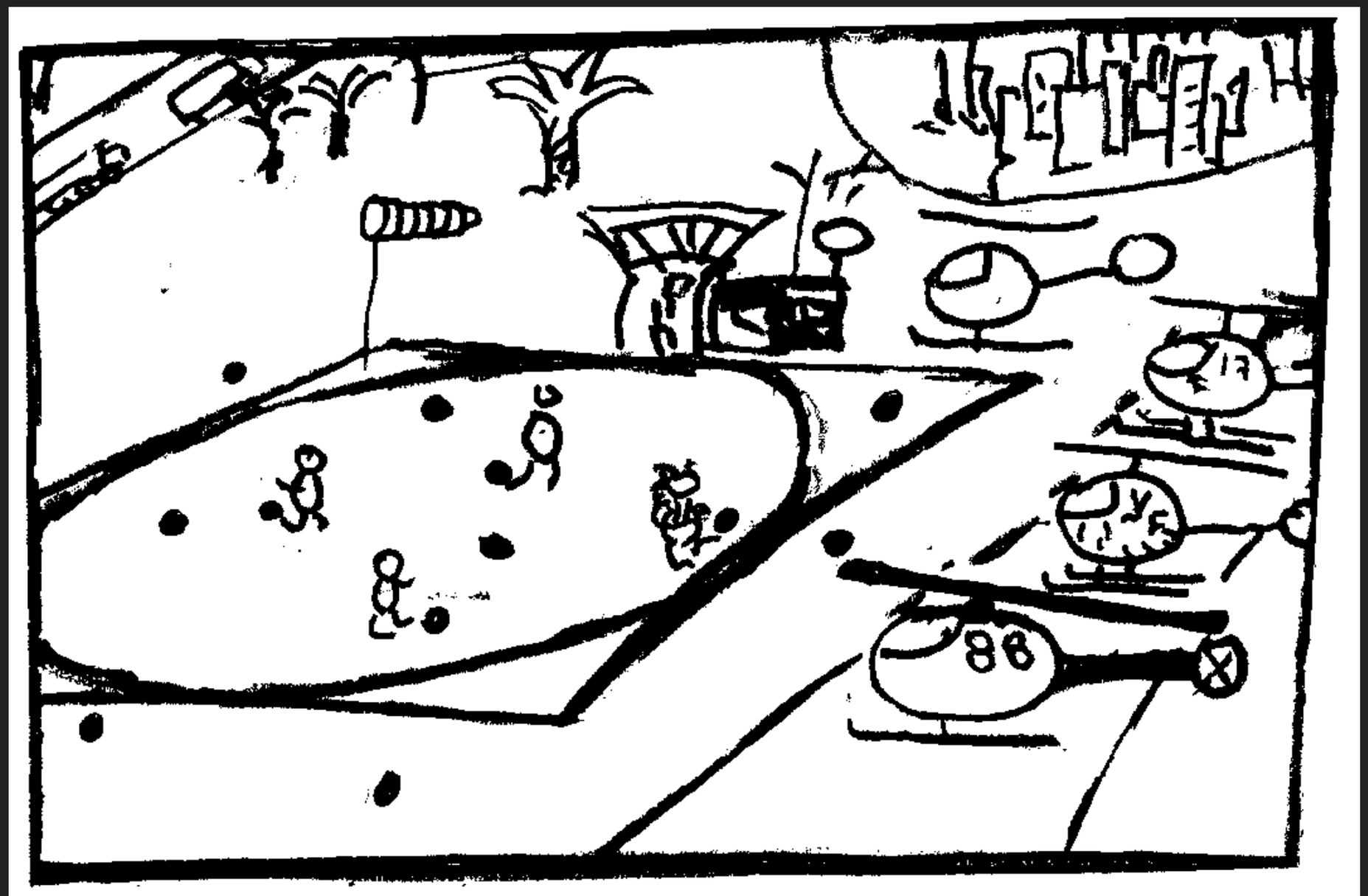
The ratio between a circle's circumference and its diameter, named π, is a mathematical constant of crucial importance to science, yet most scientists rely on precomputed approximations of π for their research. This is problematic, because scientific progress relies on information that will very likely disappear in case of a cataclysmic event, such as a zombie apocalypse. In such case, scientific progress might even stop entirely. This motivates the need for a robust, yet easily applicable method to estimate π.

$$\mathbb{E}_f[g(\mathbf{x})] \approx \hat{\mathbb{E}}_f[g(\mathbf{x})] = \frac{1}{N}\sum_{i=1}^{N} g(\mathbf{x}_i), \quad \mathbf{x}_i \sim f(\mathbf{x}) \qquad (2)$$

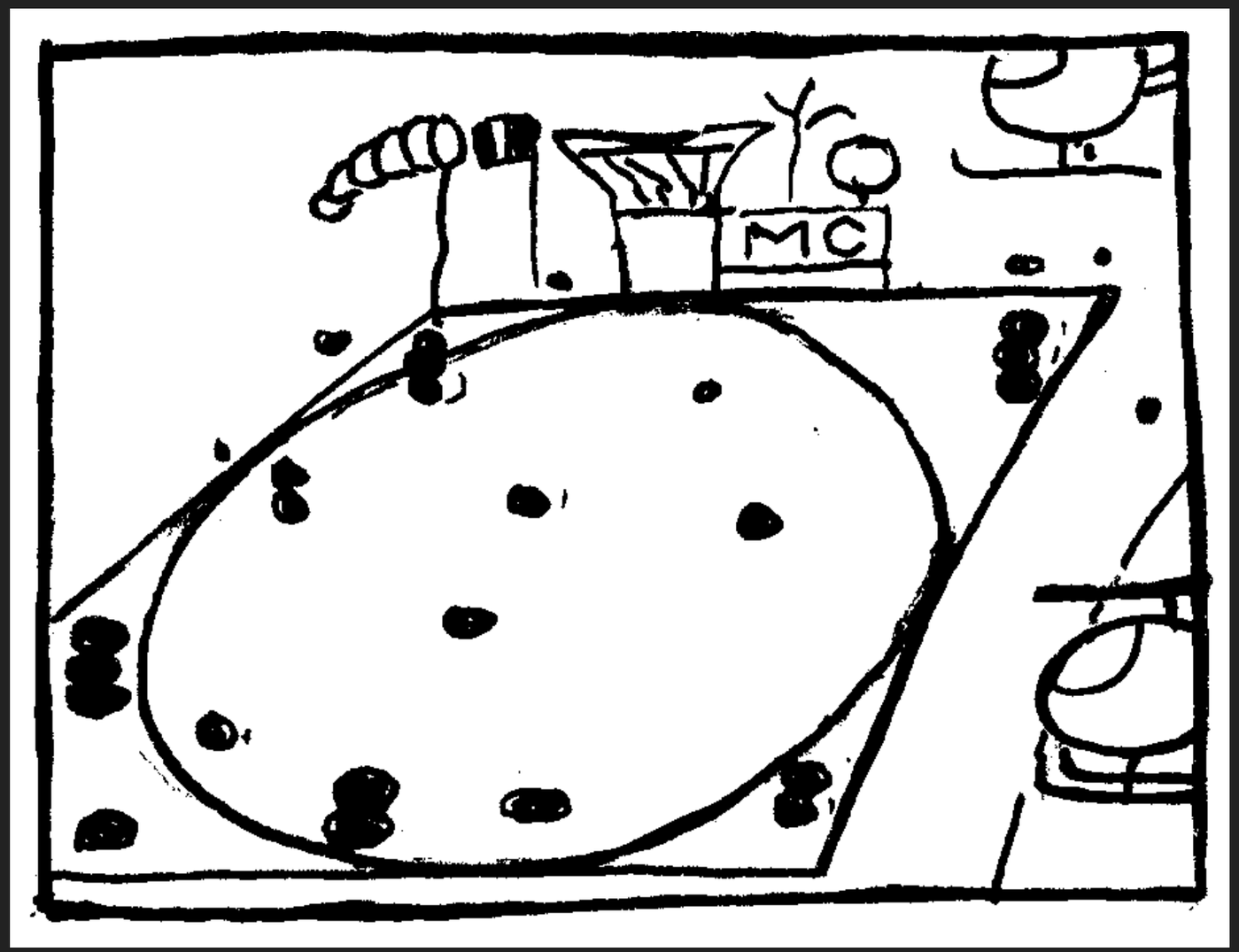
where $\mathbf{x}_i \sim f(\mathbf{x})$ means $\mathbf{x}_i$ is drawn $f(\mathbf{x})$.

Note that $\hat{\mathbb{E}}_f[g(\mathbf{x})]$ is a consistent estimator of $\mathbb{E}_f[g(\mathbf{x})]$, i.e. $\hat{\mathbb{E}}_f[g(\mathbf{x})]$ converges in probability to $\mathbb{E}_f[g(\mathbf{x})]$ as $N \to \infty$. Furthermore, its variance decreases as $\frac{1}{N}$ *independently of the dimensionality of* $\mathbf{x}$. For more details, see [2].

ADULTS GAME (MARKOV CHAIN SAMPLING)



AFTER THE GAME



crucial role of rejections

procedure `markov-pi`

hits $\leftarrow 0$; $x \leftarrow 1$; $y \leftarrow -1$

for $i = 1, \ldots, N$ **do**

$\delta x \leftarrow \texttt{ran}[-\delta, \delta]$

$\delta y \leftarrow \texttt{ran}[-\delta, \delta]$

if $(|x + \delta x| < 1$ and $|y + \delta y| < 1)$ **then**

$x \leftarrow x + \delta x$

$y \leftarrow y + \delta y$

if $(x^2 + y^2 < 1)$ $N_{\text{hits}} \leftarrow N_{\text{hits}} + 1$

output N_{hits}

MARKOV CHAIN MONTE CARLO

MONTE CARLO FOR CLASSICAL SYSTEMS

- evaluate phase space integral by importance sampling

$$\langle A \rangle = \frac{\int_\Omega A(c)p(c)dc}{\int_\Omega p(c)dc} \longrightarrow \langle A \rangle \approx \overline{A} \equiv \frac{1}{M}\sum_{i=1}^{M} A(c_i)$$

- pick configurations with correct Boltzmann weight

$$P(c) = \frac{p(c)}{Z} = \frac{\exp(-\beta E(c))}{Z}$$

- But how do we create configurations with that distribution?

- The key problem in statistical mechanics!

THE METROPOLIS ALGORITHM

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
  - **ergodicity:** any configuration reachable from any other

    $$\forall_{x,y}\exists_n : (W^n)_{x,y} > 0$$

  - **balance:** the distribution is stationary

    $$0 = \frac{d}{dt}p(x) = \sum_y p(y)W_{y,x} - \sum_y p(x)W_{x,y} \implies p(x) = \sum_y p(y)W_{y,x}$$

DETAILED BALANCE

- simplest and most common way to satisfy balance condition

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

- detailed balance is sufficient but not necessary for balance
- global balance algorithms can explore configuration space more efficiently – but are harder to construct

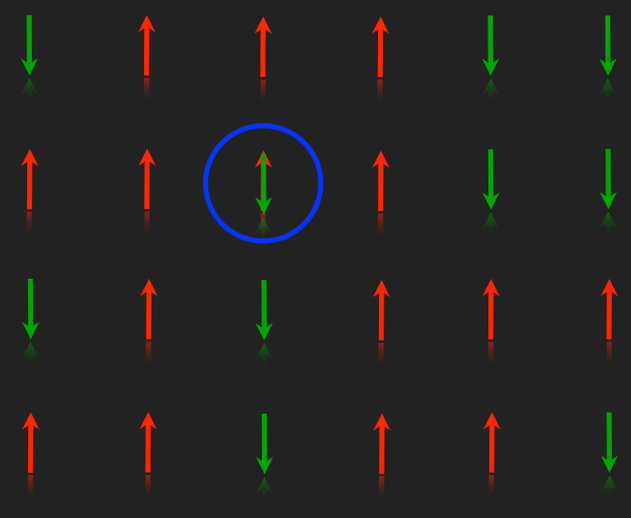
THE METROPOLIS ALGORITHM

- 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 *at al.*

METROPOLIS ALGORITHM FOR THE ISING MODEL



1. pick a random spin and propose to flip it
2. accept the flip with probability $P = \min\left[1, e^{-\beta(E_{\text{new}} - E_{\text{old}})}\right]$
3. perform a measurement – independent of whether the flip was accepted or rejected!

EQUILIBRATION

- starting from a random initial configuration it takes a while to reach the equilibrium distribution
- the desired equilibrium distribution is a left eigenvector with eigenvalue 1 (balance condition)

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

- convergence controlled by second-largest eigenvalue

$$p(x,t) = p(x) + O(e^{-\lambda_2 t})$$

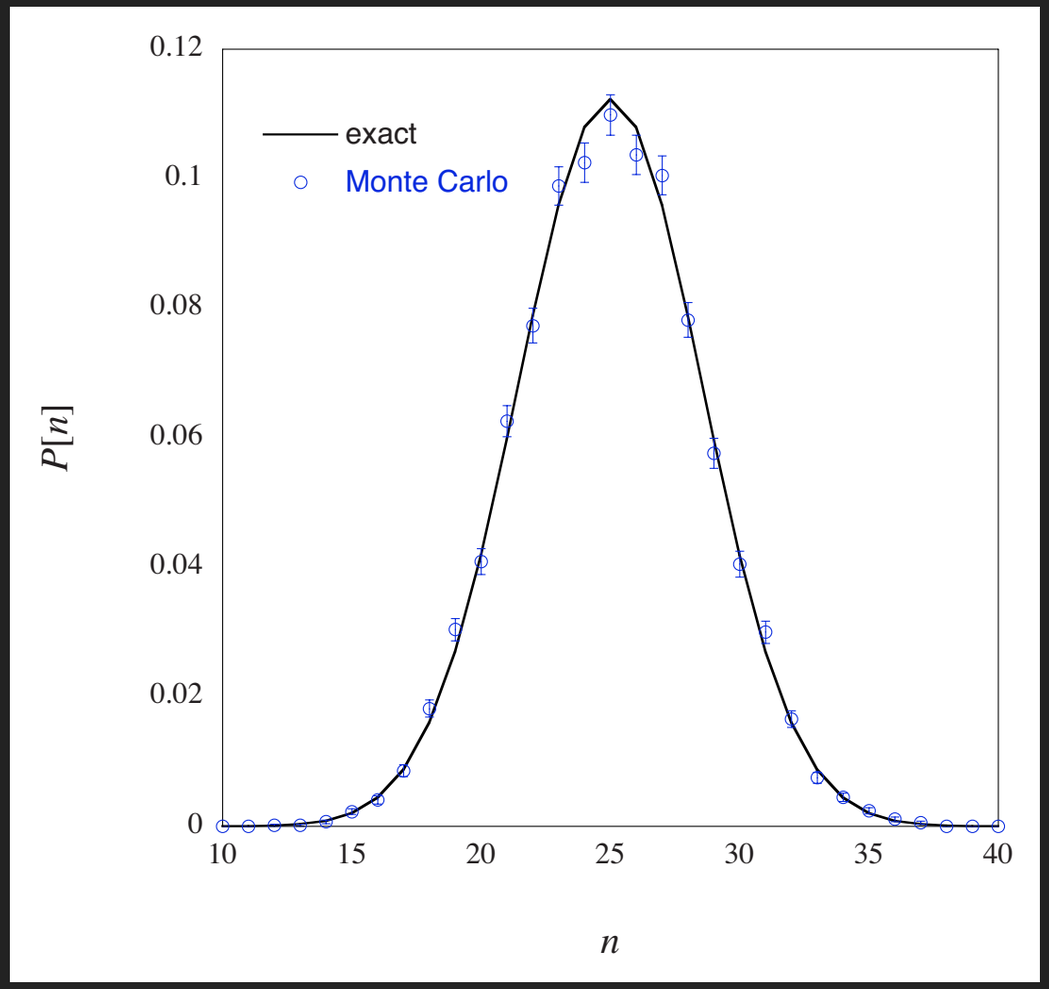
- we need to run the simulation for a while to equilibrate and only then start measuring

MONTE CARLO ERROR ANALYSIS

THE DOGS & FLEAS MODEL

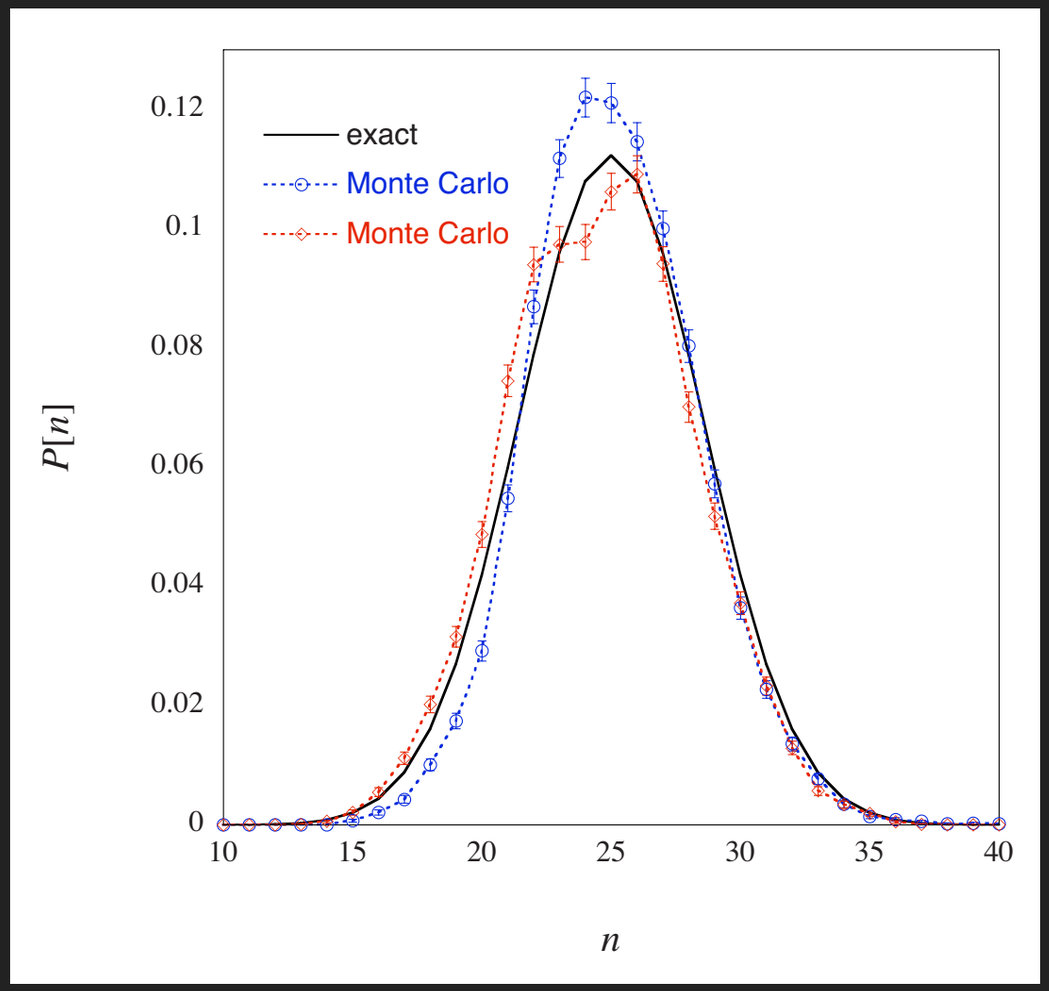
- two dogs play
  - Anik has 50 fleas
  - Burnside has no fleas
- at each time step a random flea jumps to the other dog
- what is the distribution of fleas after they played?
- Vinay Ambegaokar and Matthias Troyer
  American Journal of Physics 78, 150 (2010)

DOGS & FLEAS – DIRECT SAMPLING



$$\Delta_A = \sqrt{\frac{\operatorname{Var} A}{M}} \approx \sqrt{\frac{\overline{A^2} - \overline{A}^2}{M-1}}$$

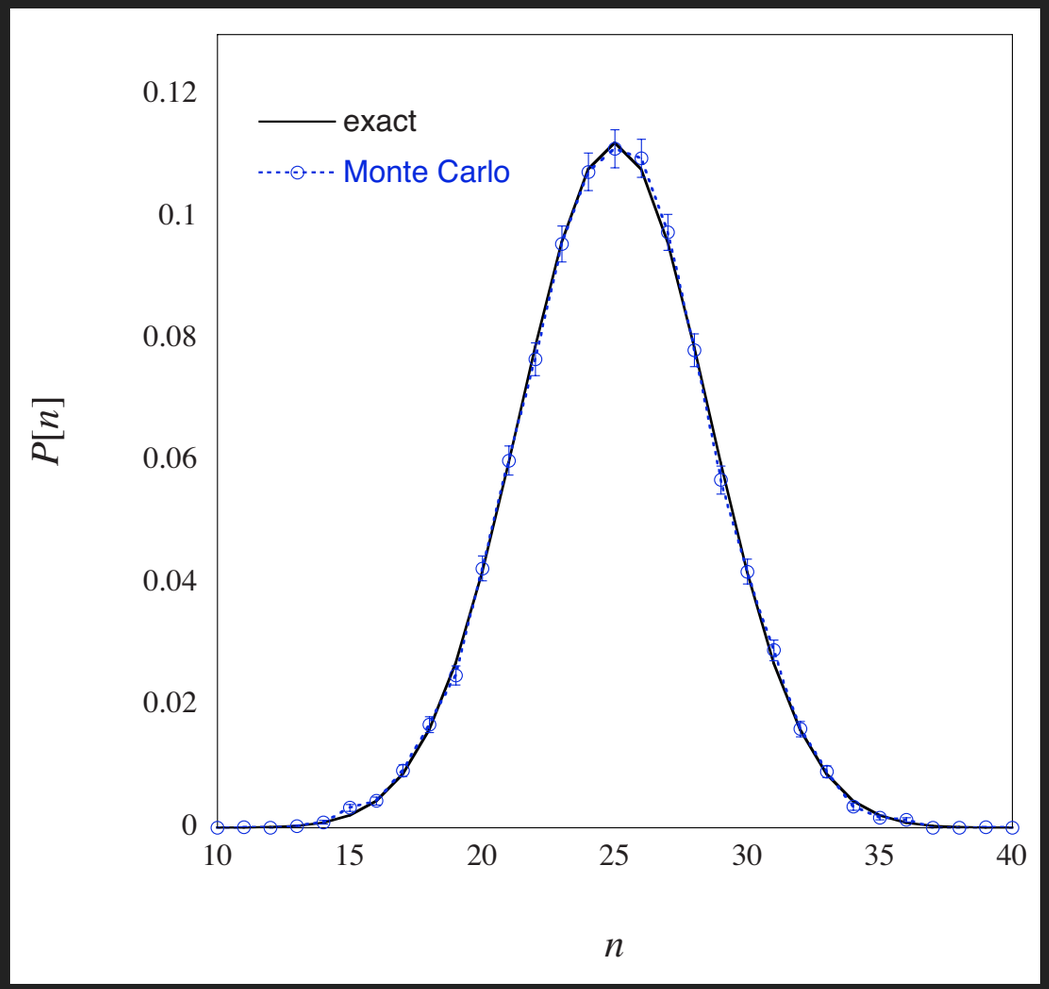
DOGS & FLEAS – NAIVE ERRORS



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

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

DOGS & FLEAS – UNCORRELATED ERRORS



- one flea hop does not change much
- measure only after 99 hops

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

MONTE CARLO ERROR ANALYSIS

- the simple formula $\Delta_A = \sqrt{\frac{\operatorname{Var} A}{M}}$
  is valid only for independent samples
- Metropolis algorithm generates correlated samples!
- number of independent samples is reduced:

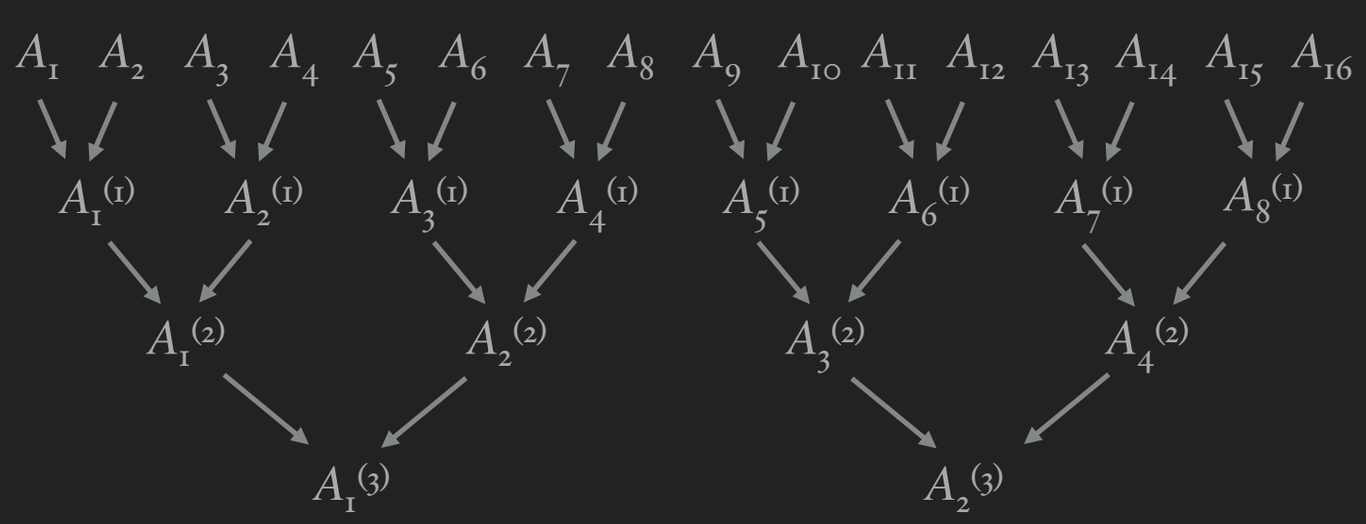
$$\Delta_A = \sqrt{\frac{\operatorname{Var} A}{M}\left(1 + 2\tau_A\right)}$$

- autocorrelation time is defined by

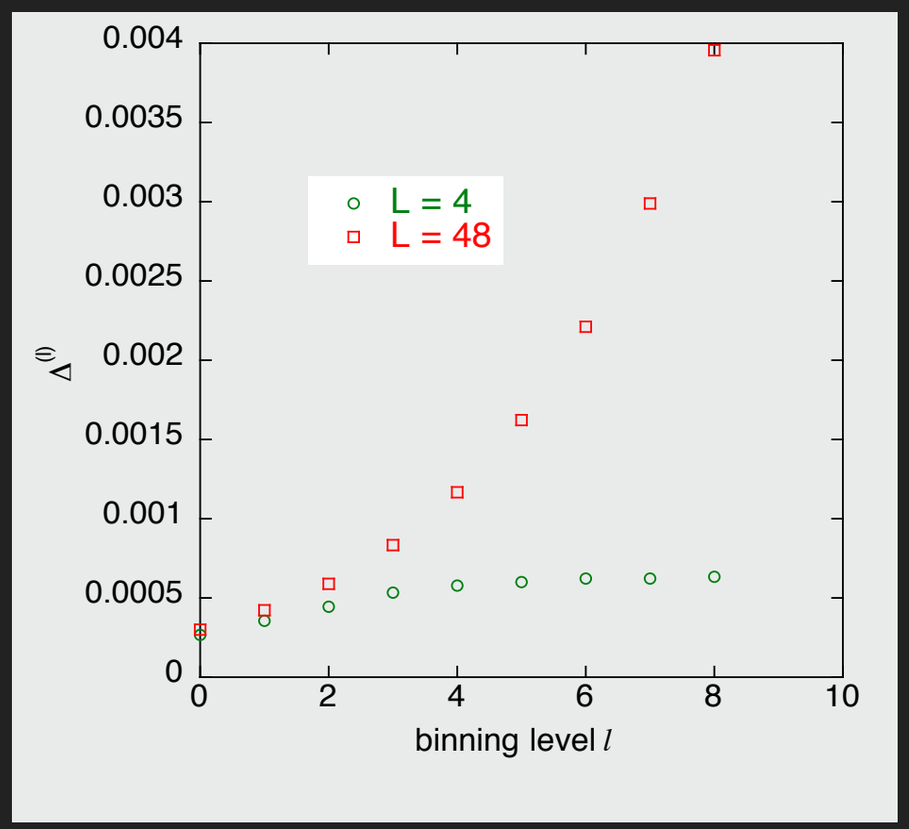
$$\tau_A = \frac{\sum_{t=1}^{\infty}\left(\langle A_{i+t} A_i\rangle - \langle A\rangle^2\right)}{\operatorname{Var} A}$$

BINNING ANALYSIS

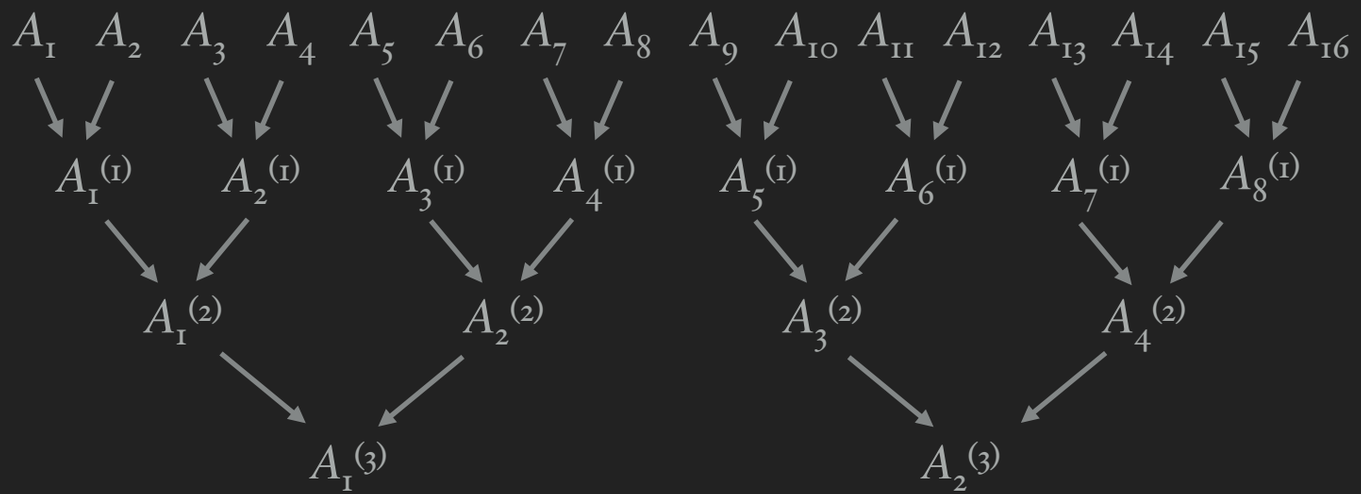
take averages of consecutive measurements: averages become less correlated and naive error estimates converge to real error



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



BINNING ANALYSIS

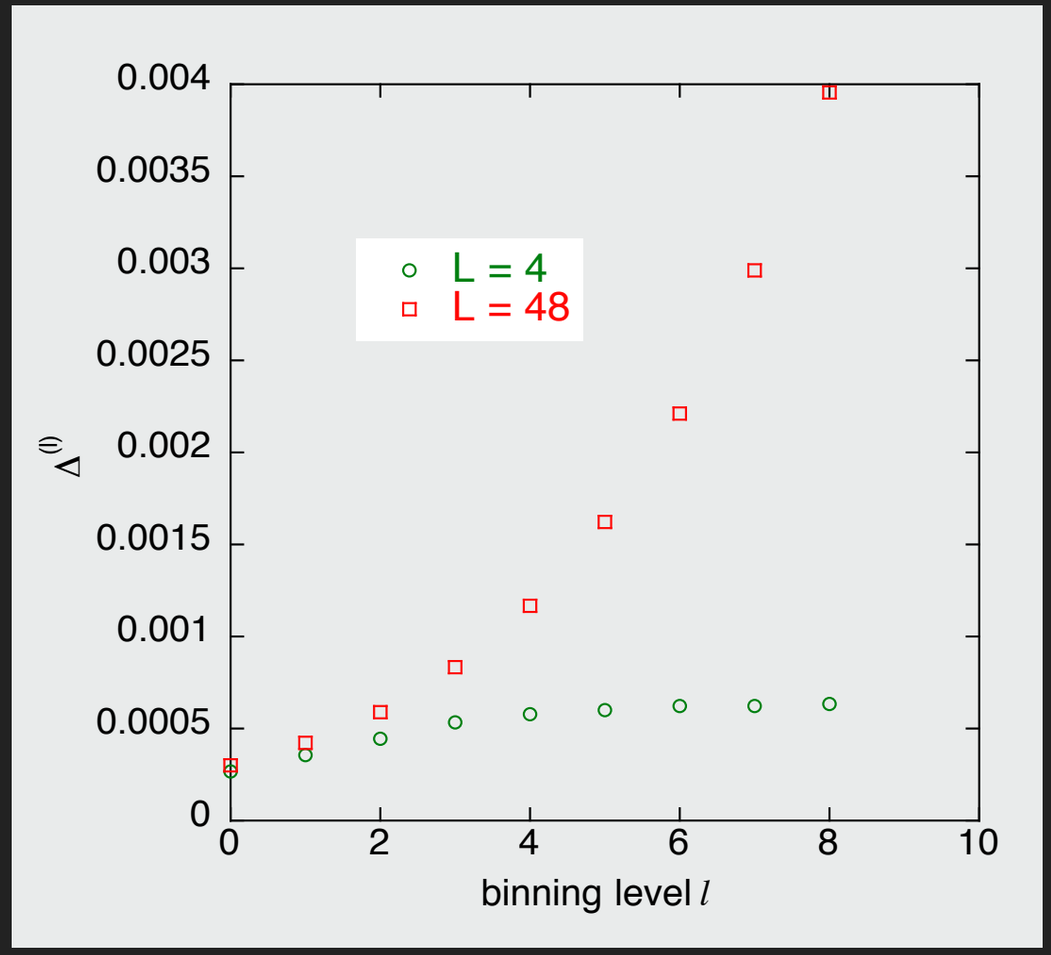


$$\Delta^{(l)} = \sqrt{\mathrm{Var}A^{(l)}/M^{(l)}}$$

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

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

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



CORRELATED QUANTITIES

- how do we calculate the errors of functions of correlated measurements?
  - specific heat $c_V = \dfrac{\langle E^2 \rangle - \langle E \rangle^2}{T^2}$
  - Binder cumulant $U = \dfrac{\langle m^4 \rangle}{\langle m^2 \rangle^2}$
- the naive way of assuming uncorrelated errors is wrong!
- also nonlinearities can lead to biased results!
- jackknife/bootstrap resampling