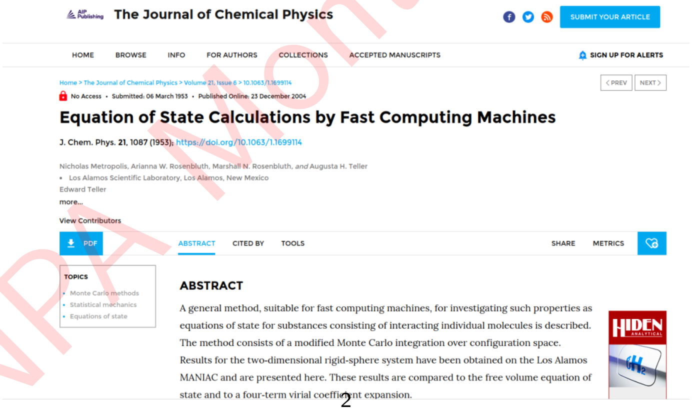
Markov Chain Monte Carlo: Metropolis-Hasting Algorithm

Prof. Dr. Narayan Prasad Adhikari
Central Department of Physics
Tribhuvan University Kirtipur, Kathmandu, Nepal

April 6, 2025

Introduction

- The original paper by Metropolis et al. (1953) deals with the calculation of properties of chemical substances and was published in the Journal of Chemical Physics. Nevertheless, it later proved itself to have a great impact in Statistics and Simulation.



The Journal of Chemical Physics

Home > The Journal of Chemical Physics > Volume 21, Issue 6 > 10.1063/1.1699114

No Access • Submitted: 06 March 1953 • Published Online: 23 December 2004

Equation of State Calculations by Fast Computing Machines

J. Chem. Phys. 21, 1087 (1953); https://doi.org/10.1063/1.1699114

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, *and* Augusta H. Teller

- Los Alamos Scientific Laboratory, Los Alamos, New Mexico

Edward Teller

more...

View Contributors

TOPICS

- Monte Carlo methods
- Statistical mechanics
- Equations of state

ABSTRACT

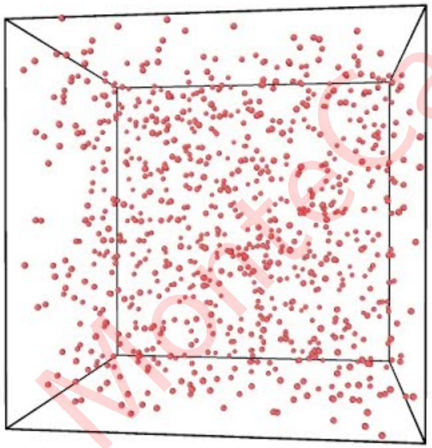
A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional 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.

- Consider a substance with d molecules positioned at $\theta = (\theta_1, ..., \theta_d)'$ In this case, the component θ_i is formed by the bidimensional vector of positions in the plane of the ith molecule. From Statistical Mechanics, the density of these positions is given by Equation

$$f(x_1, ...x_d) \;\propto \exp\left(-E(x_1, ..., x_d)/kT\right) \quad (1)$$

  vectors where a potential V between molecules can be defined. The potential energy of the substance is then given by $E(\theta) = \sum_{i,j} V(\theta_i, \theta_j)/2$.

- The calculation of the equilibrium value of any chemical property is given by the expected value of this property with respect to the distribution of the vector of positions. Direct calculation of the expectation is not feasible for d large and is replaced by a Monte Carlo estimate.



- Metropolis et al. (1953) suggested a method to deal with the difficult problem of sampling from this density.

■Metropolis Algorithm

1. Start with any initial configuration $\theta^{(0)} = (\theta_1^{(0)}, ..., \theta_d^{(0)})'$ and set the iteration counter j = 1.
2. Move the particles from previous positions $\theta^{(j-1)} = (\theta_1^{(j-1)}, ..., \theta_d^{(j-1)})'$ according to a uniform distribution centered at these positions in order to obtain new positions $\phi = (\phi_1, ..., \phi_d)'$.
3. Calculate the change ΔE in the potential energy caused by the move. The move in step 2 is accepted with probability min $\{1, \exp(-\Delta E)/kT\}$, with If the move is accepted, $\theta^{(j)} = \phi$ Otherwise, $\theta^{(j)} = \theta^{(j-1)}$
4. Change the counter from j to $j+1$ and return to step 2 until convergence is reached.

After convergence, the vector of positions generated by the method has density according to probability density given by equation 1.

■Metropolis Algorithm

- It is evident that the above method defines a Markov chain as the transitions depend only on the positions at the previous stage. However, it is not obvious that the method converges to an equilibrium distribution and that this distribution is given by equation 1.

Figure: Nicholas Metropolis who invented Metropolis Algorithm

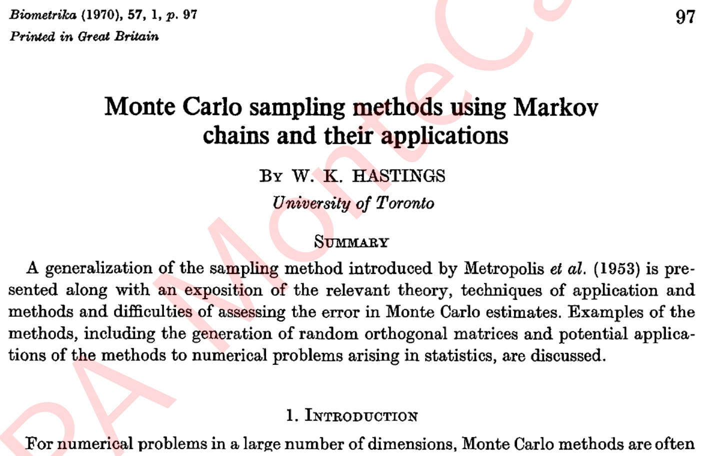
■Metropolis Algorithm

Figure: Wilfred Keith Hastings (July 21, 1930 – May 13, 2016[1]) was a statistician. He was noted for his contribution to the Metropolis–Hastings algorithm, the most commonly used Markov chain Monte Carlo method (MCMC)

■Metropolis Algorithm

- Metropolis and coworkers present a heuristic proof (non-formal a simple argument) of this result. The same proof is valid for the case where the moves to ϕ are made according to any symmetric distribution centered at previous positions. This defines a transition kernel q that depends on (θ, ϕ) through $(|\theta - \phi|)$.
- Hastings in 1970 referred to the above algorithm in this extended form as the Metropolis method.

■Hastings Algorithm



Biometrika (1970), 57, 1, *p.* 97
Printed in Great Britain

Monte Carlo sampling methods using Markov chains and their applications

BY W. K. HASTINGS
University of Toronto

SUMMARY

A generalization of the sampling method introduced by Metropolis *et al.* (1953) is presented along with an exposition of the relevant theory, techniques of application and methods and difficulties of assessing the error in Monte Carlo estimates. Examples of the methods, including the generation of random orthogonal matrices and potential applications of the methods to numerical problems arising in statistics, are discussed.

1. INTRODUCTION

For numerical problems in a large number of dimensions, Monte Carlo methods are often

■Metropolis Algorithm

- Note that the above algorithm includes an additional step that was not present in the chains previously presented. The transition mechanism now depends on a proposed transition q and a subsequent step of evaluation of this proposal. Also note that the proposed positions are completely unrelated from the equilibrium distribution but this is represented in the overall transition through the acceptance probability because

$$\frac{\pi(\phi)}{\pi(\theta^{(j-1)})} = \frac{\exp\left(-E(\phi)/kT\right)}{\exp\left(-E(\theta^{(j-1)})/kT\right)} = \exp\left(-\Delta E/kT\right) \qquad (2)$$

■Metropolis Algorithm

- Another important point is that the resulting chain may remain in a low energy (or equivalently, high density) position for many iterations. In this case, it is likely that the proposal will lead to very high energy (very low density) points and $\Delta E >>> 0$ forcing an acceptance probability very close to 0. Computationally, this is not desirable and transition kernels must be carefully chosen to avoid such low acceptance rates.

■Definition and Properties

- Consider a distribution π from which a sample must be drawn via Markov chains. Again, it is worth stressing that this task will only make sense if the non-iterative generation of π is very complicated or expensive. In this case, a transition kernel $p(\theta, \phi)$ must be constructed in a way such that π is the equilibrium distribution o f the chain. A simple way to do this is to consider reversible chains where the kernel p satisfies

$$\pi(\theta)p(\theta, \phi) = \pi(\phi)p(\phi, \theta) \quad \forall(\theta, \phi) \tag{3}$$

  As previously seen, this is the reversibility condition of the chain. It is also called *detailed balance equation.*

- Even though this is not a necessary condition for convergence, it is a sufficient condition in order that π be the equilibrium distribution of the chain.

■Definition and Properties

- The kernel $p(\theta, \phi)$ consists of 2 elements: (i) an arbitrary transition kernel $q(\theta, \phi)$ and a probability $\alpha(\theta, \phi)$ such that

$$p(\theta, \phi) = q(\theta, \phi)\alpha(\theta, \phi), \text{if } \theta \neq \phi \tag{4}$$

So, the transition kernel defines a density $p(\theta, \cdot)$ for every possible value of the parameter different from θ. Consequently, there is a positive probability left for the chain to remain at θ given by

$$p(\theta, \theta) = 1 - \int q(\theta, \phi)\alpha(\theta, \phi)d\phi \tag{5}$$

These two forms can be grouped in the general expression

$$p(\theta, A) = \int_A q(\theta, \phi)\alpha(\theta, \phi)d\phi + I(\theta \in A)\left[1 - \int q(\theta, \phi)\alpha(\theta, \phi)d\phi\right] \tag{6}$$

for any subset A of the parameter space.

■Definition and Properties

- So, the transition kernel defines a mixed distribution for the new state ϕ of the chain. For $\theta \neq \phi$, this distribution has a density and for $\phi = \theta$, this distribution has a probability atom.

- Hastings proposed to define the acceptance probability in such a way that when combined with the arbitrary transition kernel, it defines a reversible chain. The expression most commonly cited for the acceptance probability is

$$\alpha(\theta, \phi) = min\left\{1, \frac{\pi(\phi)q(\phi, \theta)}{\pi(\theta)q(\theta, \phi)}\right\} \tag{7}$$

Algorithms based on chains with transition kernel given by equation 6 and acceptance probability equation 7 will be referred to as Metropolis-Hastings algorithms. Hastings referred to the ratio appearing in eq. 7 as the test ratio.

■Justification/Proof of Metropolis-Hastings Algorithm

- The purpose of the Metropolis–Hastings algorithm is to generate a collection of states according to a desired distribution $p(\theta)$. To accomplish this, the algorithm uses a Markov process, which asymptotically reaches a unique stationary distribution $\pi(\theta)$ such that $\pi(\theta) = p(\theta)$
- A Markov process is uniquely defined by its transition probabilities $p(\phi|\theta)$, the probability of transitioning from any given state θ to any other given state ϕ. It has a unique stationary distribution $\pi(\theta)$ when the following two conditions are met.

■Justification/Proof of Metropolis-Hastings Algorithm

1. *Existence of stationary distribution: there must exist a stationary distribution $\pi(\theta)$. A sufficient but not necessary condition is detailed balance, which requires that each transition $\theta \to \phi$ is reversible: for every pair of states θ, ϕ, the probability of being in state θ and transitioning to state ϕ must be equal to the probability of being in state ϕ and transitioning to state θ,*

$$\pi(\theta)p(\phi|\theta) = \pi(\phi)p(\theta|\phi)$$

2. *Uniqueness of stationary distribution:the stationary distribution $\pi(\theta)$ must be unique. This is guaranteed by ergodicity of the Markov process, which requires that every state must (1) be aperiodic—the system does not return to the same state at fixed intervals; and (2) be positive recurrent—the expected number of steps for returning to the same state is finite.*

■Justification/Proof of Metropolis-Hastings Algorithm

The Metropolis–Hastings algorithm involves designing a Markov process (by constructing transition probabilities) that fulfills the two above conditions, such that its stationary distribution $\pi(\theta)$ is chosen to be $p(\theta)$. The derivation of the algorithm starts with the condition of detailed balance:

$$p(\theta)p(\phi|\theta) = p(\phi)p(\theta|\phi)$$

which is written as

$$\frac{p(\phi|\theta)}{p(\theta|\phi)} = \frac{p(\phi)}{p(\theta)}. \tag{8}$$

■Justification/Proof of Metropolis-Hastings Algorithm

The approach is to separate the transition in two sub-steps; the proposal and the acceptance-rejection. The proposal distribution $q(\phi \mid \theta)$ is the conditional probability of proposing a state ϕ given θ , and the acceptance distribution $\alpha(\phi, \theta)$ is the probability to accept the proposed state ϕ. The transition probability can be written as the product of them:

$$p(\phi|\theta) = q(\phi \mid \theta)\alpha(\phi, \theta)$$

Inserting this relation in the previous equation, we get

$$\frac{\alpha(\phi, \theta)}{\alpha(\theta, \phi)} = \frac{p(\phi)q(\theta \mid \phi)}{p(\theta)q(\phi \mid \theta} \tag{9}$$

■Justification/Proof of Metropolis-Hastings Algorithm

The next step in the derivation is to choose an acceptance ratio that fulfills the condition above. One common choice is the Metropolis choice:

$$\alpha(\phi, \theta) = min\left\{1, \frac{p(\phi)q(\theta \mid \phi)}{p(\theta)q(\phi \mid \theta}\right\} \tag{10}$$

For this Metropolis acceptance ratio α, either $\alpha(\phi, \theta) = 1$ or $\alpha(\theta, \phi) = 1$ and, either way, the condition is satisfied.

■ Metropolis-Hastings Algorithm

In practical terms, simulation of a draw from n using the Markov chain defined by the transition given by equation 6 can be set up as follows:

1. Initialize the iteration counter $j = 1$ and set an arbitrary initial value $\theta^{(0)}$.
2. Move the chain to a new value ϕ generated from the density $q(\theta^{(j-1)}, \cdot)$
3. Evaluate the acceptance probability of the move $\alpha(\theta^{(j-1)}, \phi)$ given by equation 7. If the move is accepted, set $\theta^{(j)} = \phi$. If it is not accepted, $\theta^{(j)} = \theta^{(j-1)}$ and the chain does not move.
4. Change the counter from j to $j + 1$ and return to step 2 until convergence is reached.

■ Metropolis-Hastings Algorithm

Step 3 is performed after the generation of an independent uniform quantity u - a random number (in practice). If $u < \alpha$, the move is accepted and if $u > \alpha$ the move is not allowed. The transition kernel q defines only a possible move that can be confirmed according to the value of α. For that reason, q is generally referred to as the proposal kernel or proposal (conditional) density when looked upon as a (conditional) density $q(\theta, \cdot)$. Other terms sometimes used are probing kernel or density.

In any of the forms of the Metropolis algorithm, q defines a symmetric transition around the previous positions of the molecules. Therefore,

$$q(\theta, \phi) = q(\phi, \theta)$$

■ Metropolis-Hastings Algorithm

for every (θ, ϕ) and the acceptance probability becomes

$$\alpha(\theta, \phi) = min\left\{1, \frac{\pi(\phi)}{\pi(\theta)}\right\} \tag{11}$$

depending only on a simplified test ratio $\frac{\pi(\phi)}{\pi(\theta)}$, the ratio of the posterior density values at the proposed and previous positions of the chain.

Note also that the chain may remain in the same state for many iterations. A useful monitoring device of the method is given by the average percentage of iterations for which moves are accepted. Hastings suggests that this acceptance rate should always be computed in practical applications.

■ Metropolis-Hastings Algorithm

It is also crucial that the proposal kernels are easy to draw from as the method replaces the difficult generation of π by many generations proposed from q. Another less obvious but equally important requirement to be met by q is the correct tuning of the moves it proposes to ensure that moves covering the parameter space can be made and accepted in real computing time.

The test ratio can be rewritten as

$$\frac{\pi(\phi)/q(\theta,\phi)}{\pi(\theta)//q(\phi,\theta)} \tag{12}$$

Acceptance of proposed values is based on the ratio of target and proposed density. So, there is a connection here with the resampling schemes. In resampling schemes, the proposal density q was to be chosen as similar as possible to n to increase acceptance rates but the methods were not iterative. Also, for the rejection method, the rejection probability depended only on the numerator in 12.

■ Metropolis-Hastings Algorithm

The target distribution π enters the algorithm through the test ratio $\pi(\phi)/\pi(\theta)$ in the form of the ratio as in the resampling methods. So again, the complete knowledge of π is not required. In particular, proportionality constants are not needed. When π is a posterior density, even though its functional form is always known, the value o f the proportionality constant is rarely known. So, the algorithm is particularly useful for applications to Bayesian inference.

Many o f the comments made about Gibbs sampling in the previous chapter are also valid for the Metropolis-Hastings algorithm. So, the discussion about single long against multiple chains is just as relevant here.

■ Metropolis-Hastings Algorithm

Formal and informal convergence techniques described in **Gibbs Sampling** can all be used here. The exception is made up of those based on complete knowledge of conditional densities. Typically, but not necessarily, Metropolis-Hastings algorithms are used when these are not completely known and hence difficult to sample from. W hen the complete conditional densities are known, Gibbs sampling is generally used.

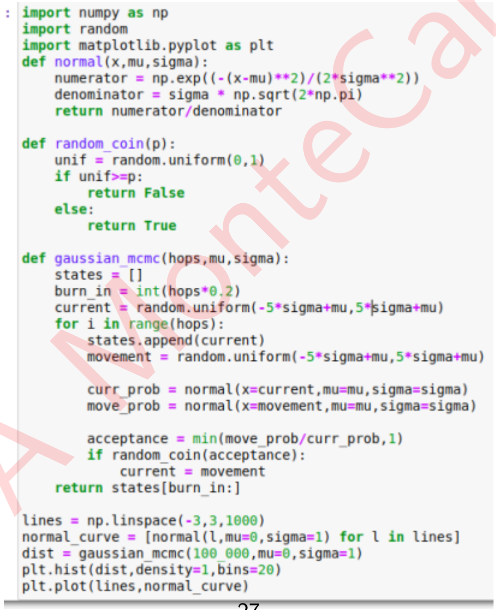
The algorithm has the interesting feature of not needing the value of the normalization constant in the probability density function. This is the most interesting feature of Metropolis-Hastings Algorithm.

■ Metropolis-Hastings Algorithm: Example

We define three functions: (1) Normal, which evaluates the probability density of any observation given the parameters mu and sigma. (2) $Random_coin$. And (3) $Gaussian_mcmc$, which samples executes the algorithm as described.

We're not calling any Gaussian or normal function from numpy, scipy, etc. In the third function, we initialize a current sample as an instance of the uniform distribution (where the lower and upper boundaries are $+/$- 5 standard deviations from the mean.) Likewise, movement is defined in the same way. Lastly, we move (or stay) based on the random event's observed value in relation to acceptance, which is the probability density comparison discussed at length elsewhere.

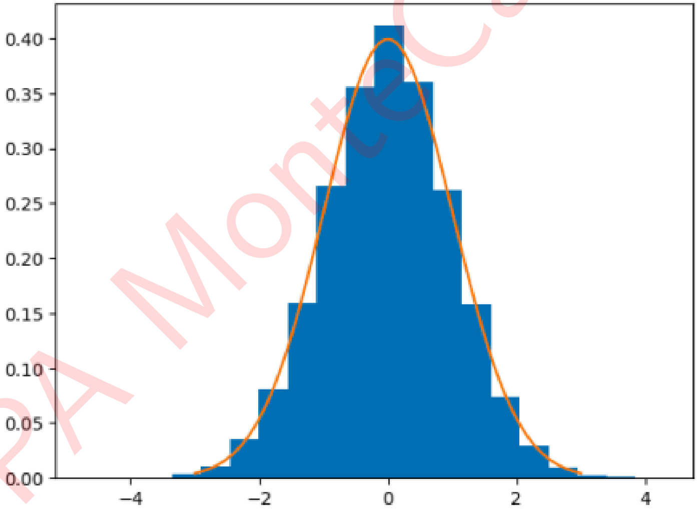
■ Metropolis-Hastings Algorithm: Code



```python
import numpy as np
import random
import matplotlib.pyplot as plt
def normal(x,mu,sigma):
    numerator = np.exp((-(x-mu)**2)/(2*sigma**2))
    denominator = sigma * np.sqrt(2*np.pi)
    return numerator/denominator

def random_coin(p):
    unif = random.uniform(0,1)
    if unif>=p:
        return False
    else:
        return True

def gaussian_mcmc(hops,mu,sigma):
    states = []
    burn_in = int(hops*0.2)
    current = random.uniform(-5*sigma+mu,5*sigma+mu)
    for i in range(hops):
        states.append(current)
        movement = random.uniform(-5*sigma+mu,5*sigma+mu)

        curr_prob = normal(x=current,mu=mu,sigma=sigma)
        move_prob = normal(x=movement,mu=mu,sigma=sigma)

        acceptance = min(move_prob/curr_prob,1)
        if random_coin(acceptance):
            current = movement
    return states[burn_in:]

lines = np.linspace(-3,3,1000)
normal_curve = [normal(l,mu=0,sigma=1) for l in lines]
dist = gaussian_mcmc(100_000,mu=0,sigma=1)
plt.hist(dist,density=1,bins=20)
plt.plot(lines,normal_curve)
```

■ Metropolis-Hastings Algorithm: Code

```
Out[8]: [<matplotlib.lines.Line2D at 0x7f975a1e1af0>]
```


0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−4
−2
0
2
4