

## Marshall Rosenbluth and the Metropolis algorithm<sup>a)</sup>

J. E. Gubernatis<sup>b)</sup>

*Los Alamos National Laboratory, MS B262, P.O. Box T-11, Los Alamos, New Mexico 87545*

(Received 29 November 2004; accepted 14 February 2005; published online 28 April 2005)

The 1953 publication, “Equation of State Calculations by Very Fast Computing Machines” by N. Metropolis, A. W. Rosenbluth and M. N. Rosenbluth, and M. Teller and E. Teller [J. Chem. Phys. **21**, 1087 (1953)] marked the beginning of the use of the Monte Carlo method for solving problems in the physical sciences. The method described in this publication subsequently became known as the Metropolis algorithm, undoubtedly the most famous and most widely used Monte Carlo algorithm ever published. As none of the authors made subsequent use of the algorithm, they became unknown to the large simulation physics community that grew from this publication and their roles in its development became the subject of mystery and legend. At a conference marking the 50th anniversary of the 1953 publication, Marshall Rosenbluth gave his recollections of the algorithm’s development. The present paper describes the algorithm, reconstructs the historical context in which it was developed, and summarizes Marshall’s recollections. © 2005 American Institute of Physics. [DOI: 10.1063/1.1887186]

### I. INTRODUCTION

Just before Marshall Rosenbluth embraced a career in plasma physics, he was an author of what is arguably the most significant publication in the history of computational physics. This famous paper,<sup>1</sup> published in June, 1953, was entitled “Equation of State Calculations by Fast Computing Machines,” and was coauthored by Nick Metropolis, Marshall and his first wife Arianna, and Edward Teller and his wife Augusta (nicknamed Mici). It presented what is now known as the Metropolis algorithm. The use of this algorithm and its variants has spread from its original equation of state application in chemical physics to problems ranging from calculating hadronic spectra in high energy physics, polymer growth and folding in polymer physics, itinerant ferromagnetism and superconductivity in condensed matter physics, atomic and molecular spectra in quantum chemistry, as a few examples. It is also the object of study in mathematics and statistics in the theory of Markov chains. It is even used in the social sciences as a tool for simulating financial markets and the self-organization and resolution of social and political conflicts. In bioinformatics, it is being used to refine cancer treatments. The Metropolis algorithm is the most famous and widely used Monte Carlo method ever published.

Perhaps, somewhat ironically, this algorithm is not widely used in plasma physics. (It is not a useful tool for simulating nonlinear dynamics.) Thus, while most plasma physicists might likely be surprised about Marshall’s role in an important computational event, they also might likely be surprised by his role becoming known to the algorithm’s users only within the past one and one-half years. In fact, the roles of the all the coauthors only became clearer in the past one and one-half years. This new clarity was largely provided by Marshall<sup>2</sup> at a 2003 conference organized to cel-

brate the 50th anniversary of the publication of the algorithm. I will now recount his revelations and other facts about the history of this famous publication.<sup>3</sup>

### II. BACKGROUND

The equation of state work by Metropolis *et al.* was one of the first scientific uses of a computer Metropolis had just finished building. This computer, given the incongruous acronym MANIC (mathematical analyzer numerical integrator and calculator), was based on what today is called the von Neumann architecture where the program is stored in the computer’s memory. This architecture was a major advance in power, operation, and use over existing electronic computers and is the grandfather architecture of computers we use today.

The MANIAC was not the first of such computers but was the first at Los Alamos. More importantly, it captured the imaginations of the scientists there, particularly the imagination of Teller who was eager to use it. Teller did equation of state work during the development of the first atomic and hydrogen bombs, and Marshall and Arianna worked with Teller on the hydrogen bomb. While Metropolis worked with Teller on the atomic bomb, he was now mainly interested in how his computer would perform for a variety of scientific calculations. Its instruction set was designed to be optimal for hydrodynamic calculations, and the equation of state work was one of the first projects to test the MANIAC’s the performance on other types of problems. To do the equation of state work, the authors discovered that they had to develop a new computational method. It would be simple to say the rest became history, but there has never been any known verbal or written history of the algorithm’s development.

The lack of such an account has always seemed mysterious, particularly in light of the first sentence of the paper:<sup>1</sup>

“The purpose of this paper is to describe a general method, suitable for fast electronic computing ma-

<sup>a)</sup>Paper H11 1, Bull. Am. Phys. Soc. **49**, 173 (2004).

<sup>b)</sup>Invited speaker.

chines, of calculating the properties of any substance which may be considered as composed of interacting particles,”

which suggests the authors knew they were potentially doing something significant. In statistical mechanics, all substances are considered as composed of interacting particles, and the use of “general” and “any” in this context translates this sentence into saying that the paper is presenting a numerical method which in principle solves all problems in equilibrium statistical mechanics. What this sentence makes mysterious is none of the authors ever using the algorithm again despite being aware of its potential significance. We now know that shortly after the paper’s publication the spouses focused on raising families, Metropolis was already designing his next computer, Teller helped to set up the Lawrence Livermore National Laboratory, and Marshall began his career in plasma physics.

Another mystery has been the lack of information about the individual roles and contributions of the various coauthors. Teller was the only one generally known to the community of the algorithm’s users. Marshall’s stature and presence in plasma physics was underappreciated. Metropolis, outside of Los Alamos, was also unknown. The roles of Arianna and Mici were even less appreciated. The most famous story, and it is surprising how popular it is, says that the algorithm was worked out on a cocktail napkin at a party one evening by Edward, Nick, and Marshall, and the names of the spouses were placed on the paper to placate them for having to spend an evening bored by all the technical talk.

The truth is a bit different than this. In the first place, by 1953 Mici was an experienced computer programmer and operator. She and Edward were in the first group of people to arrive at Los Alamos during World War II. The cloistering of the community as a secret city fostered a tradition of the spouses helping with the administrative and technical tasks. Mici helped execute computations for the design of the atomic bomb, first on mechanical calculators. As the sophistication of calculating machines grew so did her skills. Arianna has a Ph.D. in physics from Harvard under the mentorship of the Nobel Laureate van Vleck. On completing her degree, she received an Atomic Energy Commission post-doctoral fellowship to Stanford where she met Marshall. When Teller summoned Marshall, his former Ph.D. student, to Los Alamos to work on the hydrogen bomb, she also came to Los Alamos and assisted in executing the complex calculations for this bomb on the electronic computers of the time. Adriana told me<sup>4</sup> that Mici started the computer code for the equation of state work, but she took it over and wrote from scratch the one used.

With these bits of history the roles of the coauthors are clearer but still incompletely defined. Better definition came at the 50th anniversary conference. As part of the conference presenting a historical perspective of the evolution of the algorithm, one member of the organizing committee, I believe Daan Frenkel, suggested answering once and for all the question about how the algorithm got started before it was too late. We discussed inviting Teller to speak (we knew who he was) but soon realized that his age and health made his

participation unlikely. Then, the suggestion was made to invite Marshall about whom we knew considerably less. How active had he been since the publication of the algorithm? What kind of research has he been doing? Another member of the organizing committee, David Ceperley, said he had met him several years ago while visiting San Diego and spoke with him briefly about the Metropolis algorithm. He felt Marshall might have something significant to say. The invitation to speak was extended to Marshall with an unexpected email reply:

“I fear that 50 years may be too long to wait for such an anniversary. As you know Nick M. and Mici Teller are dead, Edward is unable to travel and has little remaining memory. I have terminal cancer and it would be a 10 sigma event were I to be alive next June.”

Imagine our shock. Further communication however revealed that Marshall wanted to speak. He was told we wanted to hear what he had to say. While we did not know until about a week before the conference that he definitely would attend, we were all grateful that he in fact did.

### III. REVELATIONS

Marshall’s recounting of the development of the Metropolis algorithm first of all made it very clear that Metropolis played no role in its development other than providing computer time.

Marshall acknowledged several helpful conversations with von Neumann. Von Neumann and Ulam were the inventors of the Monte Carlo method in 1946, and Von Neumann pointed out to Marshall that the Monte Carlo method could be used for numerical integration. Marshall’s exposure to Monte Carlo simulations being performed at Los Alamos at the time was likely to just ones simulating radiation transport through fissionable materials, something quite a bit different from numerical integration.

According to Marshall, Teller made the crucial suggestion, pointing out that statistical mechanical averages could be performed by ensemble averaging instead of time averaging. Time averaging required following the detailed kinematics of the interacting particles through the time integration of Newton’s laws. Marshall comments that Metropolis’s computer (as the others of the time) was not yet big enough and fast enough to do this. (If it were, Marshall would have been a developer to the molecular dynamics method<sup>5</sup> which is another exceptionally important simulation method.) An alternative approach was necessary. This was the value of Teller’s suggestion and the value of von Neumann’s observation was in implementing Teller’s suggestion.

Lastly, Marshall commented that he and Arianna did all the work.

These revelations help clarify the history of the development of the Metropolis algorithm and the roles of the coauthors. It is unlikely that more will be said. All but Arianna have now passed on. It would have been best to have the perspectives of all the developers. It is unfortunate that we had to wait 50 years for even this piece of the history. While

Marshall's revelations resolved many of the mysteries, they do not explicitly define who had the key as opposed to the crucial idea. What is the key idea? It will unfold as I will now describe the algorithm and its novel features.

#### IV. THE ALGORITHM

The basic problem of ensemble averaging in classical equilibrium statistical mechanics is calculating averages (in the canonical ensemble, for example) via

$$\bar{F} = \frac{\int F \exp(-E/kT) d^{3N}p d^{3N}q}{\int \exp(-E/kT) d^{3N}p d^{3N}q},$$

where  $F$  is any physical observable,  $p$  and  $q$  are the momentum and position of a particle,  $E$  is the energy of the system of interacting particle,  $N$  is the number of particles, and finally  $kT$  is the Boltzmann constant  $k$  times the temperature  $T$ . Instead of constructing a uniform grid in phase space, if the trapezoid method, for example, were used to do the integrations numerically, the common Monte Carlo method of the day would sample points in phase space randomly, but on the average uniformly. This procedure would reduce the ensemble averaging to

$$\bar{F} \approx \sum_i F_i \frac{\exp(E_i/kT)}{\sum_j \exp(E_j/kT)} = \sum_i F_i p_i.$$

For cases with just more than a few particles, most configurations however have the probability  $p_i$  associated with configuration  $i$  being vanishing small. Accordingly, such a Monte Carlo computation would be highly inefficient. It would be like searching for a needle in a haystack. The Metropolis algorithm replaces the numerical integration by

$$\bar{F} \approx \frac{1}{M} \sum_m F_m,$$

that is, it chooses configurations in phase space  $m$  with the proper Boltzmann probability.

In the conference proceedings, Marshall describes the algorithm in the following beautifully concise and clear manner:<sup>2</sup>

"... A simple way to do this [sampling configurations with the Boltzmann weight], as emerged after discussions with Teller, would be to make a trial move: if it decreased the energy of the system, allow it; if it increased the energy, allow it with probability  $\exp(-\Delta E/kT)$  as determined by a comparison with a random number. Each step, after an initial annealing period, is counted as a member of the ensemble, and the appropriate ensemble average of any quantity determined."

How this algorithm was used in the original paper is depicted in Fig. 1. The problem under consideration was the solidification/melting of a collection of particles (the disks). "What features of their interactions are the most important ones?" was the question being asked. The particular model simulated is called the hard disk model where the particles experience no interaction instead of a weak attractive one

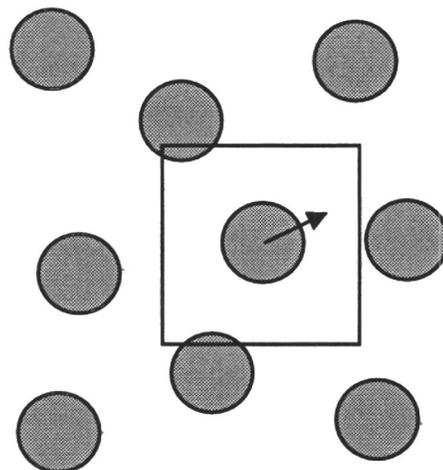


FIG. 1. Monte Carlo simulation of a system of hard disks. Each Monte Carlo step consists of proposing a move of one disk, repositioning its center to some random position inside a small square. If the move is accepted or rejected via the Metropolis algorithm, such movements will eventually sample the Boltzmann distribution. Instead of a square, a circle, for instance, could be also used. The efficiency of the simulation is controlled by the size of this square or circle.

when they do not overlap and experience an infinite instead of a very strong repulsion when they do. By ignoring the attractive part of the interaction a minimal model for solidification/melting was being tested.

The algorithm functioned by serially cycling the list of disks. For each, a trial move was made by repositioning the disk's center from its current position to some position randomly selected inside of a small square centered about the original position. If the new position did not cause any overlaps, the move was accepted, the positions of disks recorded as a member of the ensemble, a move was then attempted for the next disk, and so on. If an overlap occurred, the move was rejected, the disk returned to its original position, the original positions of the disks was repeated as a member of the ensemble, a move was then attempted for the next disk, and so on.

The algorithm is remarkable for its simplicity, and this simplicity is part of the reason for its popularity. The major reason for its popularity, however, was created a few years after its publication when it was realized that the algorithm could be used to sample most distributions  $p(i)$  other than just the Boltzmann distribution. This unleashed power developed when its mathematical connection with the theory of Markov chains was noted by Wood.<sup>6</sup>

A Markov chain is defined by a transition probability  $P(i \rightarrow j)$  from configuration  $i$  to  $j$ . After a number of initial steps, an initial annealing period, the chain starts to sample from a specific probability distribution  $p(i)$ . In this context the Metropolis algorithm is expressed as

$$P(i \rightarrow j) = \min[1, p(i)/p(j)], \quad (1)$$

$$P(i \rightarrow i) = 1 - \sum_{j \neq i} p(j)/p(i). \quad (2)$$

When  $p(i)$  is the Boltzmann distribution,  $p(i)/p(j) = \exp(-\Delta E/kT)$ . Hence, if a move lowers the energy,  $P(i \rightarrow j) = 1$ ;

if it raises it,  $P(i \rightarrow j) = \exp(-\Delta E/kT)$ . The second equation handles the probability of rejecting the move, that is, the move staying put but the configuration being repeated in the ensemble, that is, remaining part of the chain. One should note that the algorithm only depends on the ratio of the values of the distribution for two different configurations; hence, it does not depend on their normalization constant. In statistical mechanics the normalization constant  $\sum_i \exp(-E_i/kT)$  for the Boltzmann distribution is called the partition function which is the generator of all equilibrium statistical mechanics. Not having to compute it in order to compute a thermodynamic average is a useful feature of the algorithm.

The common Monte Carlo simulations of the day, radiation transport through fissionable material, were based on analogs to natural processes that *a priori* defined  $P(i \rightarrow j)$ . The unknown  $p(i)$ , the object of the simulation, for instance, might be the density of radiation of a given type with a given energy reaching a detector at position  $i$ . What the Metropolis algorithm solved was the problem of knowing  $p(i)$  *a priori* but not knowing  $P(i \rightarrow j)$ . A Copernican revolution occurred.

Why is the algorithm valid? A difficult point to understand is placing the reject configuration in the ensemble. The mathematics requires it, see Eq. (2), so not placing the rejected configuration in the ensemble produces an invalid algorithm. The original exposition however was nonmathematical. What insight validated the algorithm? It was conservation of probability, the need to avoid the movement of the particles from piling up probability in phase space incorrectly. From the mathematics, a sufficient condition that the Markov chain eventually samples from  $p(i)$  is

$$p(i)P(i \rightarrow j) = p(j)P(j \rightarrow i),$$

which is called the detailed balance condition. This equation says when the Markov chain is stationary, that is, when probability is being conserved, a connection between the movement from  $i \rightarrow j$  and  $j \rightarrow i$  can be inferred. Equations (1) and (2) are the Metropolis algorithm and identify what is now known to be one of the many possible  $P(i \rightarrow j)$  that satisfy this condition. While not explicitly stated in the original paper, detailed balance is clearly at the core of the proof of the validity of the algorithm given in the paper. It is the key idea.

While the paper had a proof of validity of the algorithm, it was initially greeted with some reservation by Kirkwood, a very prominent statistical physicist and a Los Alamos consultant.<sup>7</sup> In response to Kirkwood's concerns, Marshall worked out a detailed proof, but Kirkwood converted on his own so Marshall merely recorded the proof as an Atomic Energy Commission report. I stumbled upon a reference to this report while scanning a bibliography of a conference on Monte Carlo methods held in 1956.<sup>8</sup> Our library located it, and it now appears in the conference proceedings.<sup>9</sup> In preparing his contribution to the proceedings, Marshall, not remembering the report, reconstructed the proof. Both are done via phase space arguments for the Boltzmann (canonical) distribution instead of today's more general probabilistic treatments.

Detailed balance is the seed from which modern developments grew. While the Metropolis algorithm is efficient or

at least adequate for a large number of simulations, research typically sits in problems where the algorithm is highly inefficient. Two prominent difficult problems are sampling very close to a phase transition and simulating systems, such as glasses, that have rough energy landscapes. For some problems modern algorithms are often as much as a factor of a million more efficient than the Metropolis algorithm. This efficiency is often achieved by changing multiple variables of the configuration simultaneously instead of just one at a time *à la* Metropolis *et al.* The developers of these modern algorithms however do, just as Metropolis *et al.* did, prove the validity of their algorithm by showing it satisfies detailed balance.

## V. CLOSURE

Marshall's presence at the conference was its highlight, not only for his revelations but also for his insightful questions and comments. He only had energy and time to attend several sessions but he dominated them. At these sessions he was witnessing for the first time the products of his algorithm for physical problems outside his main interests. He wanted to understand this "new" physics, why things were as they were, and even why some of the results in the original paper required further simulations. (The MANIAC did not quite have enough power for the challenging problem selected.) His questions were always well directed and tenacious. What impressed all was the enthusiasm he displayed towards the new things. He was really enjoying himself. Outside of the conference's organizing committee few knew or could sense that he was ill. One member of the organizing committee remarked to me, "If I were facing what he is facing, I doubt if any of this would still interest me."

When I learned of Marshall's death, I reflected and felt grateful that the conference gave me a chance to meet and know him. It is unfortunate that our Monte Carlo community waited 50 years to have this pleasure. More unfortunate is that we will not have the opportunity to know him better.

Obviously, Marshall's revelations prompted hallway and diner discussions about whether the Metropolis algorithm should be called the Rosenbluth or at least the Rosenbluth-Teller algorithm. One of the organizing committee, Rajan Gupta, whose curiosity got the best of him, privately asked Marshall his opinion about the name of the algorithm. Marshall replied:<sup>10</sup>

"... life has been good to me. I feel rewarded in knowing that this algorithm will allow scientists to solve problems ranging from fluid flow to social dynamics to elucidating the nature of elementary particles."

The original name will stick.

Three times during the conference, Marshall walked up to me and said, "It hard to explain how exciting it was to be at Los Alamos during those times. To be able to interact with Teller and von Neumann was very important to me." Even heroes have their heroes.

**ACKNOWLEDGMENT**

This work was supported by the U.S. Department of Energy.

<sup>1</sup>N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *J. Chem. Phys.* **21**, 1087 (1953).

<sup>2</sup>M. N. Rosenbluth, in *The Monte Carlo Method in the Physical Sciences*, edited by J. E. Gubernatis (American Institute of Physics, New York, 2003), p. 22.

<sup>3</sup>Some pre- and post-history to the algorithm are discussed by J. E. Gubernatis, in *The Monte Carlo Method in the Physical Sciences*, edited by J. E. Gubernatis (American Institute of Physics, New York, 2003), p. 3.

<sup>4</sup>A. W. Rosenbluth (private communication).

<sup>5</sup>B. J. Alder and T. E. Wainwright, *J. Chem. Phys.* **27**, 1208 (1957).

<sup>6</sup>W. W. Wood and J. D. Jacobson, *J. Chem. Phys.* **27**, 1207 (1957).

<sup>7</sup>See W. W. Wood, in *The Monte Carlo Method in the Physical Sciences*, edited by J. E. Gubernatis (American Institute of Physics, New York, 2003), p. 39, and references therein.

<sup>8</sup>*Symposium on Monte Carlo Methods*, edited by H. A. Meyer (Wiley, New York, 1956).

<sup>9</sup>M. N. Rosenbluth, in *The Monte Carlo Method in the Physical Sciences*, edited by J. E. Gubernatis (American Institute of Physics, New York, 2003), p. 32.

<sup>10</sup>R. Gupta, in *The Monte Carlo Method in the Physical Sciences*, edited by J. E. Gubernatis (American Institute of Physics, New York, 2003), p. 110.