



Nick Metropolis



Mici Teller

Edward Teller



Arianna Rosenbluth

Marshall Rosenbluth

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, only two-body forces are considered, and the potential field of a molecule is assumed spherically symmetric. These are the usual assumptions made in theories of liquids. Subject to the above assumptions, the method is not restricted to any range of temperature or density. This paper will also present results of a preliminary two-dimensional calculation for the rigid-sphere system. Work on the two-dimensional case with a Lennard-Jones potential is in progress and will be reported in a later paper. Also, the problem in three dimensions is being investigated.

* Now at the Radiation Laboratory of the University of California, Livermore, California.

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† containing N particles. In order to minimize the surface effects we suppose the complete substance to be periodic, consisting of many such squares, each square containing N particles in the same configuration. Thus we define d_{AB} , the minimum distance between particles A and B , as the shortest distance between A and any of the particles B , of which there is one in each of the squares which comprise the complete substance. If we have a potential which falls off rapidly with distance, there will be at most one of the distances AB which can make a substantial contribution; hence we need consider only the minimum distance d_{AB} .

† We will use the two-dimensional nomenclature here since it is easier to visualize. The extension to three dimensions is obvious.

An Interview with Bernie Alder

Bernie Alder 1997, from **NERSC**, part of the *Stories of the Development of Large Scale Scientific Computing at Lawrence Livermore National Laboratory* series.

It is little known that **THE algorithm** was independently discovered by Bernie Alder, Stan Frankel and Victor Lewinson. Quoting Alder:

..we started out with a configuration, a solid like order configuration, and then jiggled the particles according to the pulse rate distribution. And that is, in fact, known now as the Monte Carlo Method—it was presumably independently developed at Los Alamos by Teller, Metropolis, and Rosenbluth. They actually got all the credit. My guess is we did it first at Cal Tech. It's not that difficult to come up with that algorithm, which, by the way, I think is one of, if not THE, most powerful algorithms

Actually, In a footnote of the original paper by Metropolis *et al.* they credited Alder, Frankel and Lewinson on this, but this fact has been almost forgotten over the years. Furthermore Alder *et al.* did not give a general formulation for the algorithm but only a specialized version for hard spheres.

Radial Distribution Function Calculated by the Monte-Carlo Method for a Hard Sphere Fluid

B. J. Alder, S. P. Frankel and V. A. Lewinson 1955

The Journal of Chemical Physics, 23, 417 (1955)

The paper mentioned above. Quoting Alder:

I was still working on my Ph.D. thesis. He (Frankel) was really well known in computing circles. He actually put the Monte Carlo Method on the FERRANTI Computer and ran it all summer. I think it was before Los Alamos had electronic computers available. Anyway, we ran it and he came back. The thing that happens, Kirkwood did not believe in my boss, my thesis supervisor, he didn't believe in him at the college and, of course, he had communication with Los Alamos. The fact is, we never published—you can't publish something your boss doesn't believe in! In the meantime, Teller, Rosenbluth and Metropolis independently published. There may have been some collusion or communication of ideas that I couldn't recall, but they had the machines, so they published and we published only years later. There is, in fact, a footnote in the Metropolis paper giving us credit of having independently developed it.

Our method in this respect is similar to the cell method except that our cells contain several hundred particles instead of one. One would think that such a sample would be quite adequate for describing any one-phase system. We do find, however, that in two-phase systems the surface between the phases makes quite a perturbation. Also, statistical fluctuations may be sizable.

If we know the positions of the N particles in the square, we can easily calculate, for example, the potential energy of the system,

$$E = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N V(d_{ij}). \quad (1)$$

(Here V is the potential between molecules, and d_{ij} is the minimum distance between particles i and j as defined above.)

In order to calculate the properties of our system we use the canonical ensemble. So, to calculate the equilibrium value of any quantity of interest F ,

$$\bar{F} = \left[\int F \exp(-E/kT) d^{2N}p d^{2N}q \right] / \left[\int \exp(-E/kT) d^{2N}p d^{2N}q \right], \quad (2)$$

where $(d^{2N}p d^{2N}q)$ is a volume element in the $4N$ -dimensional phase space. Moreover, since forces between particles are velocity-independent, the momentum integrals may be separated off, and we need perform only the integration over the $2N$ -dimensional configuration space. It is evidently impractical to carry out a several hundred-dimensional integral by the usual numerical methods, so we resort to the Monte Carlo method.[†] The Monte Carlo method for many-dimensional integrals consists simply of integrating over a random sampling of points instead of over a regular array of points.

Thus the most naive method of carrying out the integration would be to put each of the N particles at a random position in the square (this defines a random point in the $2N$ -dimensional configuration space), then calculate the energy of the system according to Eq. (1), and give this configuration a weight $\exp(-E/kT)$. This method, however, is not practical for close-packed configurations, since with high probability we choose a configuration where $\exp(-E/kT)$ is very small; hence a configuration of very low weight. So the method we employ is actually a modified Monte Carlo scheme, where, instead of choosing configurations randomly, then weighting them with $\exp(-E/kT)$, we choose

[†] This method has been proposed independently by J. E. Mayer and by S. Ulam. Mayer suggested the method as a tool to deal with the problem of the liquid state, while Ulam proposed it as a procedure of general usefulness. B. Alder, J. Kirkwood, S. Frankel, and V. Lewinson discussed an application very similar to ours.

configurations with a probability $\exp(-E/kT)$ and weight them evenly.

This we do as follows: We place the N particles in any configuration, for example, in a regular lattice. Then we move each of the particles in succession according to the following prescription:

$$\begin{aligned} X &\rightarrow X + \alpha \xi_1 \\ Y &\rightarrow Y + \alpha \xi_2, \end{aligned} \quad (3)$$

where α is the maximum allowed displacement, which for the sake of this argument is arbitrary, and ξ_1 and ξ_2 are random numbers§ between (-1) and 1 . Then, after we move a particle, it is equally likely to be anywhere within a square of side 2α centered about its original position. (In accord with the periodicity assumption, if the indicated move would put the particle outside the square, this only means that it re-enters the square from the opposite side.)

We then calculate the change in energy of the system ΔE , which is caused by the move. If $\Delta E < 0$, i.e., if the move would bring the system to a state of lower energy, we allow the move and put the particle in its new position. If $\Delta E > 0$, we allow the move with probability $\exp(-\Delta E/kT)$; i.e., we take a random number ξ_3 between 0 and 1 , and if $\xi_3 < \exp(-\Delta E/kT)$, we move the particle to its new position. If $\xi_3 > \exp(-\Delta E/kT)$, we return it to its old position. Then, whether the move has been allowed or not, i.e., whether we are in a different configuration or in the original configuration, we consider that we are in a new configuration for the purpose of taking our averages. So

$$\bar{F} = (1/M) \sum_{j=1}^M F_j, \quad (4)$$

where F_j is the value of the property F of the system after the j th move is carried out according to the complete prescription above. Having attempted to move a particle we proceed similarly with the next one.

We now prove that the method outlined above does choose configurations with a probability $\exp(-E/kT)$. Since a particle is allowed to move to any point within a square of side 2α with a finite probability, it is clear that a large enough number of moves will enable it to reach any point in the complete square. Since this is true of all particles, we may reach any point in configuration space. Hence, the method is ergodic.

Next consider a very large ensemble of systems. Suppose for simplicity that there are only a finite number of states¶ of the system, and that ν_r is the number of

§ It might be mentioned that the random numbers that we used were generated by the middle square process. That is, if ξ_m is an m digit random number, then a new random number ξ_{m+1} is given as the middle m digits of the complete $2m$ digit square of ξ_m .

¶ In practice it is, of course, not necessary to make enough moves to allow a particle to diffuse evenly throughout the system since configuration space is symmetric with respect to interchange of particles.

¶ A state here means a given point in configuration space.

systems of the ensemble in state r . What we must prove is that after many moves the ensemble tends to a distribution

$$\nu_r \propto \exp(-E_r/kT).$$

Now let us make a move in all the systems of our ensemble. Let the *a priori* probability that the move will carry a system in state r to state s be P_{rs} . [By the *a priori* probability we mean the probability before discriminating on $\exp(-\Delta E/kT)$.] First, it is clear that $P_{rs} = P_{sr}$, since according to the way our game is played a particle is equally likely to be moved anywhere within a square of side 2α centered about its original position. Thus, if states r and s differ from each other only by the position of the particle moved and if these positions are within each other's squares, the transition probabilities are equal; otherwise they are zero. Assume $E_r > E_s$. Then the number of systems moving from state r to state s will be simply $\nu_r P_{rs}$, since all moves to a state of lower energy are allowed. The number moving from s to r will be $\nu_s P_{sr} \exp(-\Delta E/kT)$, since here we must weigh by the exponential factor. Thus the net number of systems moving from s to r is

$$P_{rs}(\nu_s \exp(-(E_r - E_s)/kT) - \nu_r). \quad (5)$$

So we see that between any two states r and s , if

$$(\nu_r/\nu_s) > [\exp(-E_r/kT)/\exp(-E_s/kT)], \quad (6)$$

on the average more systems move from state r to state s . We have seen already that the method is ergodic; i.e., that any state can be reached from any other, albeit in several moves. These two facts mean that our ensemble must approach the canonical distribution. It is, incidentally, clear from the above derivation that after a forbidden move we must count again the initial configuration. Not to do this would correspond in the above case to removing from the ensemble those systems which tried to move from s to r and were forbidden. This would unjustifiably reduce the number in state s relative to r .

The above argument does not, of course, specify how rapidly the canonical distribution is approached. It may be mentioned in this connection that the maximum displacement α must be chosen with some care; if too large, most moves will be forbidden, and if too small, the configuration will not change enough. In either case it will then take longer to come to equilibrium.

For the rigid-sphere case, the game of chance on $\exp(-\Delta E/kT)$ is, of course, not necessary since ΔE is either zero or infinity. The particles are moved, one at a time, according to Eq. (3). If a sphere, after such a move, happens to overlap another sphere, we return it to its original position.

III. SPECIALIZATION TO RIGID SPHERES IN TWO DIMENSIONS

A. The Equation of State

The virial theorem of Clausius can be used to give an equation of state in terms of \bar{n} , the average den-

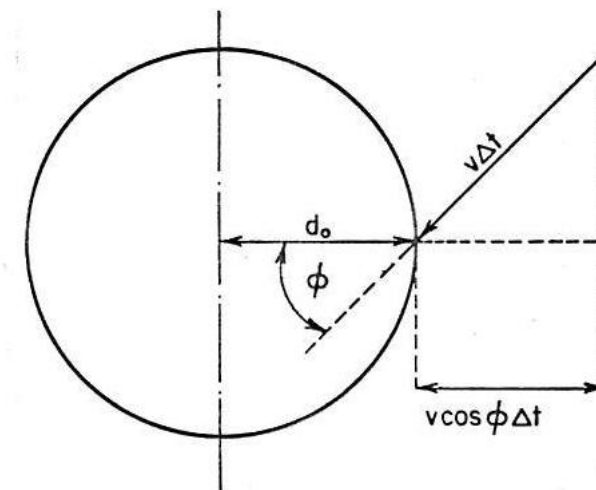


FIG. 1. Collisions of rigid spheres.

sity of other particles at the surface of a particle. Let $X_i^{(tot)}$ and $X_i^{(int)}$ represent the total and the internal force, respectively, acting on particle i , at a position \mathbf{r}_i . Then the virial theorem can be written

$$\langle \sum_i X_i^{(tot)} \cdot \mathbf{r}_i \rangle_{Av} = 2PA + \langle \sum_i X_i^{(int)} \cdot \mathbf{r}_i \rangle_{Av} = 2E_{kin}. \quad (7)$$

Here P is the pressure, A the area, and E_{kin} the total kinetic energy,

$$E_{kin} = Nm\bar{v}^2/2$$

of the system of N particles.

Consider the collisions of the spheres for convenience as represented by those of a particle of radius d_0 , twice the radius of the actual spheres, surrounded by \bar{n} point particles per unit area. Those surrounding particles in an area of $2\pi d_0 v \cos \phi \Delta t$, traveling with velocity v at an angle ϕ with the radius vector, collide with the central particle provided $|\phi| < \pi/2$. (See Fig. 1.) Assuming elastic recoil, they each exert an average force during the time Δt on the central particle of

$$2mv \cos \phi / \Delta t.$$

One can see that all ϕ 's are equally probable, since for any velocity-independent potential between particles the velocity distribution will just be Maxwellian, hence isotropic. The total force acting on the central particle, averaged over ϕ , over time, and over velocity, is

$$\bar{F}_i = m\bar{v}^2 \pi d_0 \bar{n}. \quad (8)$$

The sum

$$\langle \sum_i X_i^{(int)} \cdot \mathbf{r}_i \rangle_{Av}$$

$$- \frac{1}{2} \sum_{i \neq j} \langle \sum_{i,j} r_{ij} F_{ij} \rangle,$$

is with F_{ij} the magnitude of the force between two particles and r_{ij} the distance between them. We see that

Motivating problem

Computation of integrals of the form

$$\mathfrak{J} = \frac{\int F(p, q) \exp\{-E(p, q)/kT\} dp dq}{\int \exp\{-E(p, q)/kT\} dp dq},$$

with energy E defined as

$$E(p, q) = \frac{1}{2} \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N V(d_{ij}),$$

and N number of particles, V a potential function and d_{ij} the distance between particles i and j .

Boltzmann distribution

Boltzmann distribution $\exp\{-E(p, q)/kT\}$ parameterised by temperature T , k being the Boltzmann constant, with a normalisation factor

$$\mathfrak{Z}(T) = \int \exp\{-E(p, q)/kT\} dp dq$$

not available in closed form.

Computational challenge

Since p and q are $2N$ -dimensional vectors, numerical integration is impossible

Plus, standard Monte Carlo techniques fail to correctly approximate \mathfrak{J} : $\exp\{-E(p, q)/kT\}$ is very small for most realizations of random configurations (p, q) of the particle system.

Metropolis algorithm

Consider a random walk modification of the N particles: for each $1 \leq i \leq N$, values

$$x'_i = x_i + \alpha \xi_{1i} \quad \text{and} \quad y'_i = y_i + \alpha \xi_{2i}$$

are proposed, where both ξ_{1i} and ξ_{2i} are uniform $\mathcal{U}(-1, 1)$. The energy difference between new and previous configurations is ΔE and the new configuration is accepted with probability

$$1 \wedge \exp\{-\Delta E/kT\},$$

and otherwise the previous configuration is replicated*

*counting one more time in the average of the $F(p_t, p_t)$'s over the τ moves of the random walk.

Convergence

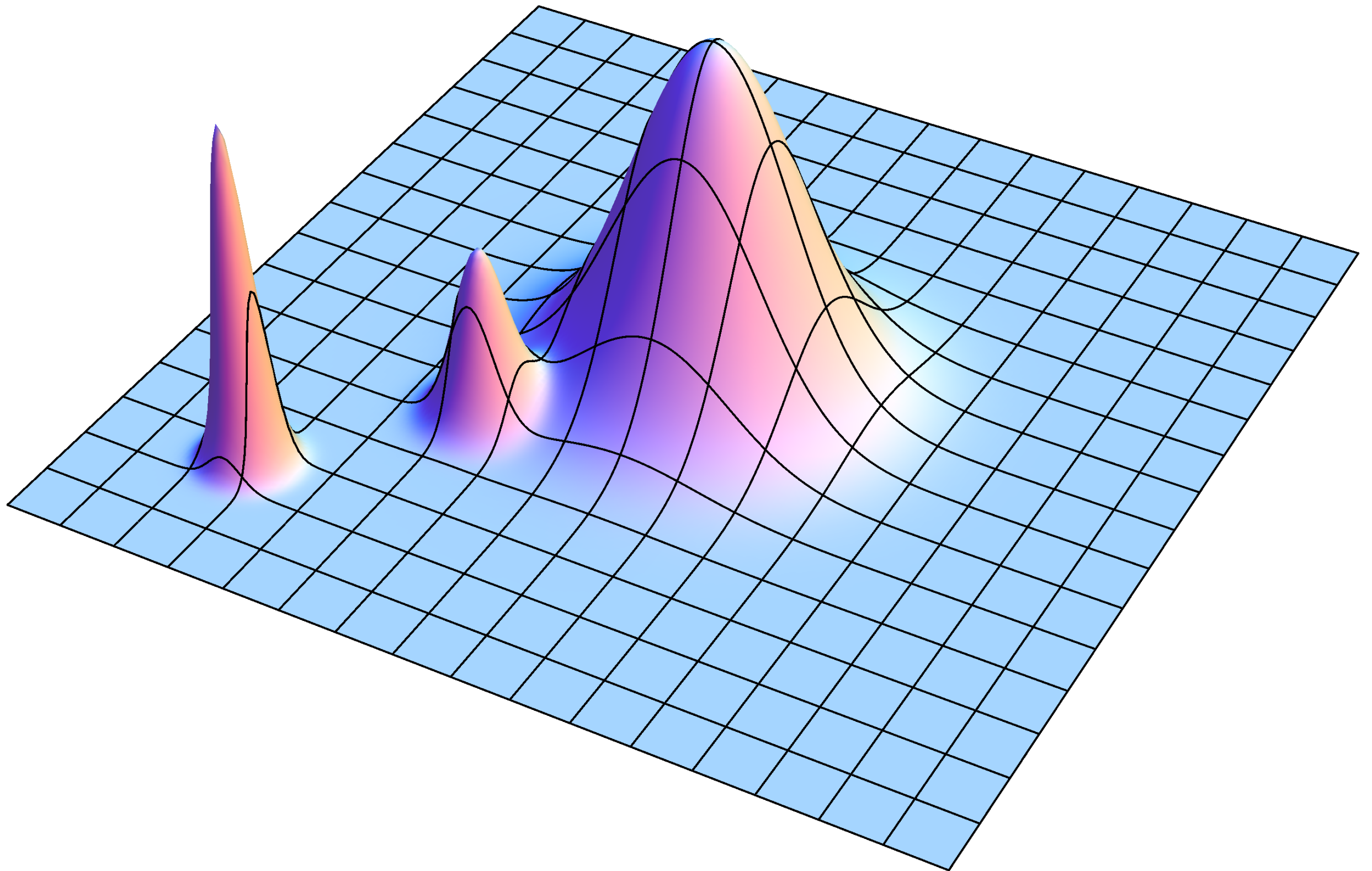
Validity of the algorithm established by proving

1. irreducibility
2. ergodicity, that is convergence to the stationary distribution.

Second part obtained via discretization of the space: Metropolis et al. note that the proposal is reversible, then establish that $\exp\{-E/kT\}$ is invariant.

Application to the specific problem of the rigid-sphere collision model. The number of iterations of the Metropolis algorithm seems to be limited: 16 steps for burn-in and 48 to 64 subsequent iterations (that still required four to five hours on the Los Alamos MANIAC).

$$g(x, y) = e^{-\frac{1}{10}[x^2 - y^2]} + \frac{1}{2}e^{-\frac{1}{20}[(x-20)^2 - (y-20)^2]} + e^{-\frac{1}{200}[(x-40)^2 - (y-40)^2]}$$



```

# mcmc driver function
def mcmc(n=1000, delta=[0.1,0.1], domain=[[-30,100],[-30,100]],startval = [0.0,0.0]):
    oldx = np.array(startval)
    accept = []
    record = []
    for ni in range(n):
        newx = proposal(oldx,delta,domain) # (1)
        a,b = metropolis_accept(newx,oldx) # (2)
        accept.append(a)
        record.append(b) # (3)
        oldx = b
    aa = Counter(accept)
    return [aa[True]/float(n),record]

```

- (1) Propose a new state
- (2) Accept or Reject new state
- (3) Record the state

```

# acceptance rejection step (2)
def metropolis_accept(newx,oldx):
    #print newx,oldx
    nom = f(newx)
    denom = f(oldx)
    #print "nom,denom",nom,denom
    r = random.uniform(0.0,1.0)
    if r < nom/denom:
        return [True,newx]
    else:
        return [False,oldx]

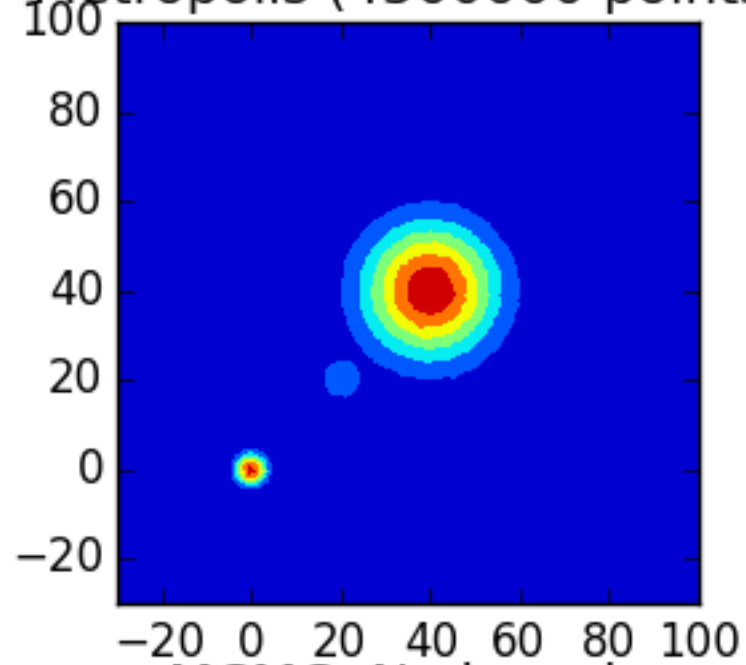
```

```

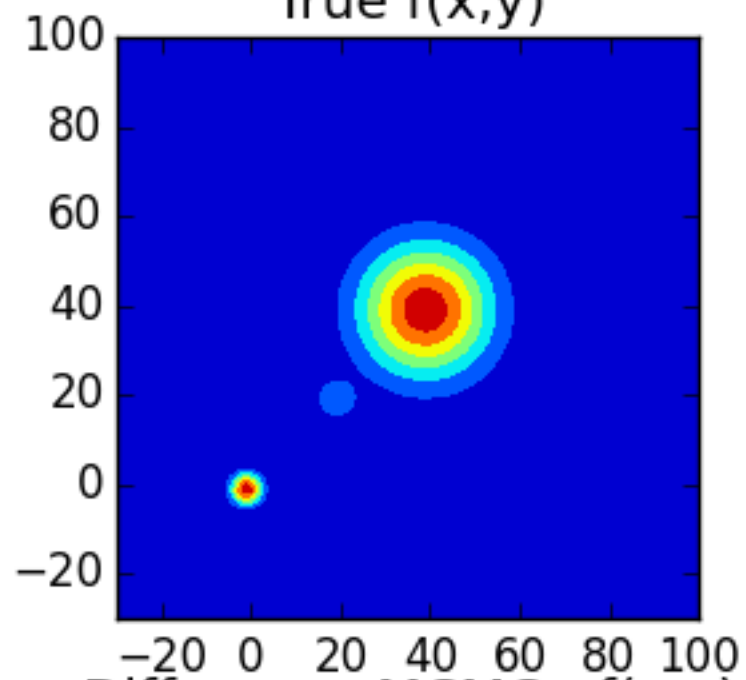
# proposal function (1)
def proposal(x,delta,domain):
    x2=[]
    for xi,di,doi in zip(x,delta,domain):
        dx = di * random.uniform(-0.5,0.5)
        while not( doi[0] < xi+dx < doi[1]):
            dx = di * random.uniform(-0.5,0.5)
        x2.append(xi + dx)
    return np.array(x2)

```

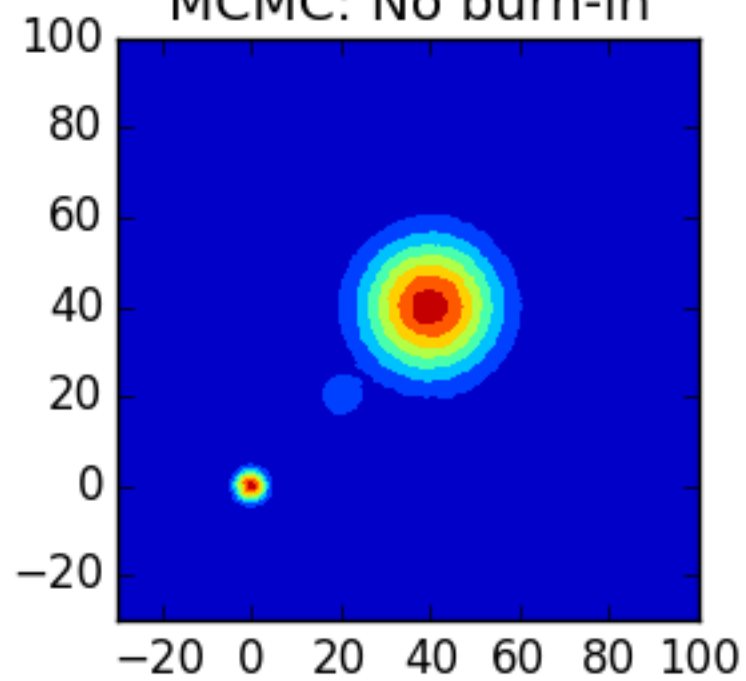

Metropolis (4500000 points)



True $f(x,y)$



MCMC: No burn-in



Difference MCMC - $f(x,y)$

