8. Monte Carlo

Over the years methods have been developed to express the solution of probabilistic problems in analytical terms. With powerful computers available the statistical problems can be modeled directly, reversing the originally intended direction. The solution to any problem that has an analytical expression that "looks like" the solution to a probability problem can be obtained by constructing a realization of the random variable problem.

Calculating π

How to calculate something with random numbers? The first example demonstrates application of Monte Carlo methods to the calculation of multidimensional integrals.

Consider a square with corners at $(\pm 1, \pm 1)$ with a circle inscribed in it; then the area of the square is 4, of the circle π.

Analytically the area of the circle can be calculated by the double integral

$$\pi = \int_{-1}^{1} dx \int_{-\sqrt{1-x^2}}^{\sqrt{1-x^2}} dy \tag{8.1}$$

If we select a number of points at random, distributed uniformly in the square, we know that the probability that they will fall within the circle is equal to the ratio of areas, $\pi/4$. The computer is now used to generate the random numbers. Thus good (pseudo-) random number generator algorithms are crucial for the success of the Monte Carlo method.

A tiny program that determines two pseudo-random numbers x and y in the interval $[-1, 1]$ and counts the number of times c that $x^2 + y^2 < 1$ about $n = 10^8$ times yields for $\pi \approx 4c/n$ (when run repeatedly)

```
Pi is about 3.14144
Pi is about 3.14147
Pi is about 3.1418
Pi is about 3.14114
Pi is about 3.14266
```

This method is easily generalized to higher dimensions; for example in order to compute the ratio of the volume of a sphere to that of a cube, $\pi/6$, one just needs an additional pseudo-random number z and has to count the number of times that $x^2 + y^2 + z^2 < 1$.

This shows that the computational expense of a multidimensional integral with Monte Carlo can be as little as linear in the dimension while conventional methods are exponential in the dimension; the time required to do the calculation is **multiplied** by the number of mesh points necessary for the additional dimension. This illustrates why Monte Carlo is often chosen for multidimensional problems.

Evaluations of integrals by Monte Carlo

Consider an integral of the form

$$\int_a^b dx\, g(x) f(x) \tag{8.2}$$

where $g(x)$ is positive definite.

We can always arrange by normalization that

$$\int_a^b dx\, g(x) = 1 \tag{8.3}$$

so that $g(x)$ can be considered to represent a **probability distribution function (PDF)**. Thus, independent of the original (physical) interpretation of the value of the integral it may be regarded as the solution of a probabilistic problem of computing the average of the distribution function $f(x)$ over a probability distribution function $g(x)$. (This average will sometimes be denoted by $\langle f \rangle$). If we draw a large number of random values of x from the distribution described by $g(x)$ we can directly estimate the average value of $f(x)$ by

$$\langle f \rangle \approx \frac{1}{N} \sum_{i=1}^N f(x_i) \qquad x_i \text{ drawn from } g(x). \tag{8.4}$$

This equation forms the basis of Monte Carlo evaluations of integrals. The set of N values x_i are said to form a **realization of the PDF** $g(x)$. This means that if we consider a range of size dx about the point x the number of points found within this range should be $Ng(x)dx$ within statistical uncertainty. At this point the method for selecting a set of x_i such that they

provide a realization of the PDF $g(x)$ is not obvious. In the simplest case as for the calculation of π above $g(x)$ can be chosen constant: $g(x) \equiv 1$, and the integration can simply be done with uniformly distributed random numbers. The art of "sampling the function $g(x)$" will now be discussed in detail. The sampling process can and should be verified by sorting the values obtained into bins (*i.e.* calculating a histogram) and comparing with the expected number.

The Monte Carlo evaluation also provides a direct estimate of the error involved in the calculation of the integral. The average of the square of the function can be calculated at the same time as the average of the function:

$$\langle f^2 \rangle = \frac{1}{N} \sum_{i=1}^{N} f_i^2 \quad x_i \text{ drawn from } g(x) \tag{8.5}$$

leading to an estimate of the variance of the integrand

$$\sigma^2 = \langle f^2 \rangle - \langle f \rangle^2 \tag{8.6}$$

and the uncertainty of the full calculation will be given by $\sigma/\sqrt{N}$. σ is the variance of the function f and will tend to a fixed number for the limit of large N while the uncertainty of the calculation falls with the square root of the number of trials.

The **cumulative distribution function (CDF)** is defined as

$$F(x) = \int_a^x g(x) \, dx \tag{8.7}$$

where the range of x is assumed to be from a to b. This function plays a central role. Since $g(x) \geqslant 0$, $F(x)$ is a non-decreasing function of x. The normalization of $g(x)$ (defined in the range $[a, b]$) requires $F(b) = 1$. $F(x)$ represents the probability that a random variable drawn from $g(x)$ lies between a and x.

The **characteristic function**

$$\phi(t) \equiv \int dx \, e^{ixt} g(x) \tag{8.8}$$

is very useful in the theory of probability. Here we will use it to derive a formula connecting probability distribution functions.

Consider the quantity $y(x_1, x_2, \ldots, x_n)$ which is a function of a set of n independent variable $\{x_1, x_2, \ldots, x_n\}$. Each variable x_i will be considered as a random variable with its own PDF $g_i(x_i)$. As the x_i take on their distribution of values, y may also be regarded as a random variable. It will also have a PDF, say $g_y(y)$. Clearly $g_y(y)$ is completely determined by the set of functions $\{g_i(x_i)\}$ and the functional relationship $y(x_1, x_2, \ldots, x_n)$. We are now looking for the function $g_y(y)$.

We may be sure that the function $g_y(y)$ represents the distribution of the random variable y if it reproduces all of the moments of the function $y(x_1, x_2, \ldots, x_n)$ as the random variables $\{x_i\}$ take on their distributions of values.

Thus we have the set of conditions

$$\int dy \, y^m g_y(y) = \int \left\{ \prod_{i=1}^{n} dx_i \, g_i(x_i) \right\} y^m(x_1, x_2, \ldots, x_n) \tag{8.9}$$

for $m = 0, 1, \ldots, \infty$.

Multiplying both sides of Eq. (8.9) by $(it)^m/m!$ and summing over all m gives two expressions for the characteristic function for the variable y

$$\phi_y(t) = \int dy \, e^{ity} g_y(y)$$

$$= \int \left\{ \prod_{i=1}^{n} dx_i \, g_i(x_i) \right\} e^{iy(x_1, x_2, \ldots, x_n)t} \tag{8.10}$$

or, inverting the Fourier transform, we find an expression for the PDF of y:

$$g_y(y') = \int \left\{ \prod_{i=1}^{n} dx_i \, g_i(x_i) \right\} \delta[y' - y(x_1, x_2, \ldots, x_n)] \tag{8.11}$$

For the case of only a single variable x Eq. (8.10) becomes

$$\phi_y(t) = \int dy \, e^{ity} g_y(y) = \int dx \, e^{iy(x)t} g(x) \tag{8.12}$$

If $y(x)$ is a monotonic function of x we can invert the equation for $y(x)$ to find $x(y)$ and, by a change of variables, write the rhs of Eq. (8.12) as

$$\phi(t) = \int dy \, e^{iyt} \frac{g[x(y)]}{\frac{dy}{dx}(y)} \tag{8.13}$$

The fundamental theorem of sampling

Consider $F(x)$, the cumulative distribution function itself, as a random variable resulting from the insertion for its argument of a random variable x obtained from the sampling of $g(x)$, the same $g(x)$ from which $F(x)$ was calculated.

Then, for Eq. (8.12) we have, with $F(x) = y(x)$

$$\int_0^1 dF\, e^{iFt} g_F(F) = \int_a^b dx\, e^{iF(x)t} g(x) \tag{8.14}$$

Thus we have an equation involving the distribution function $g_F(F)$ for the random variable F. From the definition of $F(x)$ we have $dF = g(x)dx$, so we can rewrite Eq. (8.14) by transforming the integration variable on the rhs to F:

$$\int_0^1 dF\, e^{iFt} g_F(F) = \int_0^1 dF\, e^{iFt} \tag{8.15}$$

and we can identify $g_F(F) \equiv 1$. Thus, under these conditions, F is uniformly distributed from 0 to 1. Thus if x is distributed according to $g(x)$, $F(x)$ is distributed uniformly. Of course the inverse holds as well. Since the function $F(x)$ is non-decreasing it can be inverted to obtain $x(F)$.

Thus if we choose a sequence of random variables F_i uniformly distributed from 0 to 1, the x_i which result from the expression $x(F)$ will be distributed according to $g(x)$. This method of sampling can be applied to most one-dimensional sampling problems. It is called **transformation method**.

Generalization to more than one dimension: If x_i are random variables with a joint PDF $g(x_1, x_2, \ldots)dx_1 dx_2 \ldots$ and if $y_1, y_2, \ldots$ are each functions of all x_i, then the joint PDF of the y_i is

$$g(x_1, x_2, \ldots)dx_1 dx_2 \ldots = g(y_1, y_2, \ldots)\left|\frac{\partial(x_1, x_2, \ldots)}{\partial(y_1, y_2, \ldots)}\right| \tag{8.16}$$

where $|\ldots|$ represents the Jacobian determinant.

Sampling monomials for $0 \leqslant x \leqslant 1$

We can use the previous result to sample a power of x over the range 0 to 1.

Consider a PDF $g(x)$ with the form

$$g(x) = 2x \quad \text{for } 0 \leqslant x \leqslant 1 \tag{8.17}$$

It is correctly normalized since $\int_0^1 dx\, g(x) = 1$. Now we have

$$F(x) = \int_0^x dx\, g(x) = x^2 \curvearrowright x(F) = \sqrt{F} \qquad (8.18)$$

Thus choosing a number of uniformly distributed random values F and taking the square root of each will give values of x with a linear probability distribution. In general, for a power of x:

$$g(x) = (n+1)x^n \quad \text{for } 0 \leqslant x \leqslant 1 \qquad (8.19)$$

we have

$$F(x) = (n+1)\int_0^x dx\, x^n = x^{n+1} \curvearrowright x(F) = F^{\frac{1}{n+1}} \qquad (8.20)$$

It is very important to verify that the desired distribution has indeed been obtained for each sampled function.

This can be done by calculating a histogram of the random values x and comparing to the target PDF $g(x)$. An example will be given below.

Now we can consider an example Monte Carlo integration. Take the integral

$$\int_0^1 dx\, x^4 = \frac{1}{5} \qquad (8.21)$$

We have several possibilities to write the integrand as a product $g(x)f(x)$ (the x_i are always distributed according to $g(x)$ from column 1); see Table 8.1.

$g(x)$	$f(x)$	Integral	Monte Carlo result
$2x$	x^3	$\frac{1}{2}\int_0^1(2x)(x^3)dx$	$\frac{1}{2}\frac{1}{N}\sum_{i=1}^{N} x_i^3$
$3x^2$	x^2	$\frac{1}{3}\int_0^1(3x^2)(x^2)dx$	$\frac{1}{3}\frac{1}{N}\sum_{i=1}^{N} x_i^2$
$4x^3$	x	$\frac{1}{4}\int_0^1(4x^3)(x)dx$	$\frac{1}{4}\frac{1}{N}\sum_{i=1}^{N} x_i$
$5x^4$	1	$\frac{1}{5}\int_0^1(5x^4)(1)dx$	$\frac{1}{5}\frac{1}{N}\sum_{i=1}^{N} 1$

Table 8.1: Examples for writing an integrand x^4 as product of two functions.

The last case gives the exact answer without the need for random numbers. This illustrates the point that the ability to sample a function is exactly
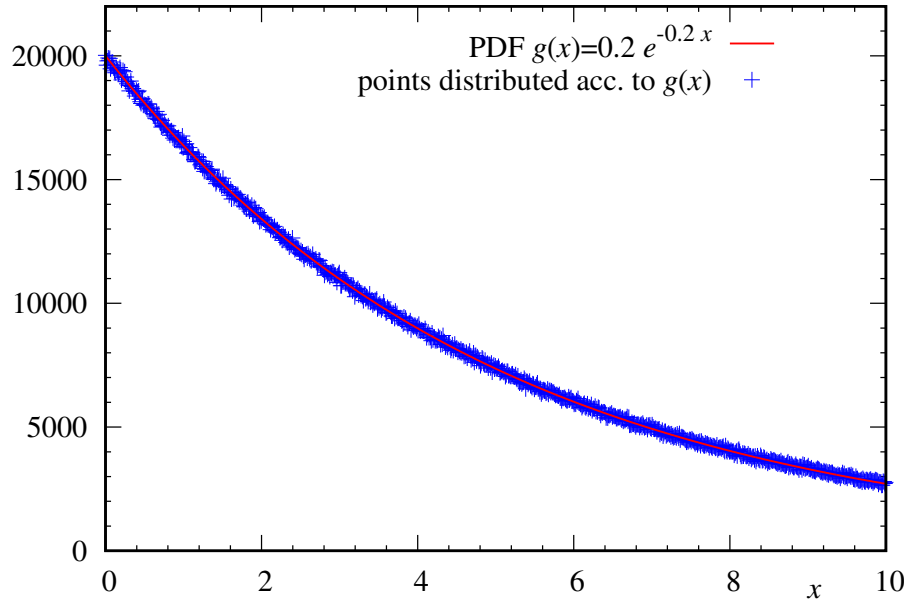
Figure 8.1: Comparison of probability distribution function $g(x)$ and a histogram obtained from 10^7 random numbers.

equivalent to being able to integrate it.

Comparing the variance of the integrand to the value of the integral we find

$$\frac{\sigma}{\langle f \rangle} = \frac{m}{\sqrt{2m+1}} \tag{8.22}$$

where m is the power in the sums in the table. The more of the variation of the integrand that can be included in the sampling function, rather than in the evaluated function, the more accurate the integral will be, since the evaluated function will approach a constant.

Sampling the exponential function $0 \leqslant x \leqslant \infty$

An exponential is easy to sample and very common. The PDF is

$$g(x) = pe^{-px} \quad \text{for } 0 \leqslant x \leqslant \infty \tag{8.23}$$

and the CDF

$$F(x) = 1 - e^{-px} \curvearrowright x = -\frac{1}{p}\ln(1-F) \tag{8.24}$$

If F is uniformly distributed, so is $1 - F$ so that we can also use $x = -\ln(F)/p$ for sampling.

Other algebraically invertible functions

In order to be able to directly sample a function it must have certain

113

properties:

1) It must be normalizable and positive definite so it can represent a PDF.
2) I must be integrable analytically.
3) The integral (antiderivative) must be invertible.

Sampling a Gaussian distribution

The Gaussian is not easily sampled in this direct manner. While the PDF is given by

$$g(x) = \frac{2}{\sqrt{\pi}} e^{-x^2} \quad \text{for } 0 \leqslant x \leqslant \infty \tag{8.25}$$

and the CDF by

$$F(x) = \frac{2}{\sqrt{\pi}} e^{-x^2} \int_0^x dx \, e^{-x^2} \equiv \text{erf}(x) \tag{8.26}$$

the error function is not analytically invertible. However one can use a trick often employed with Gaussians of transforming from rectangular to polar coordinates for two functions. We first notice that for two functions $g(x)$

$$g(x)g(y)dx\,dy = \frac{4}{\pi} e^{-r^2} r\,dr\,d\theta = \frac{2}{\pi} e^{-r^2} (2r\,dr)d\theta \tag{8.27}$$

Thus, to sample the same area (same probability) we must take a slice of

$$e^{-r^2} 2r\,dr = e^{-z} dz \qquad z = r^2 \tag{8.28}$$

and

$$\frac{2}{\pi} d\theta \tag{8.29}$$

Transforming back to Cartesian coordinates

$$x = r \cos \theta \quad y = r \sin \theta \tag{8.30}$$

the sampling functions are

$$\begin{aligned} x &= \sqrt{-\ln F_1} \cos 2\pi F_2 \\ y &= \sqrt{-\ln F_1} \sin 2\pi F_2 \end{aligned} \tag{8.31}$$

where the range of θ is extended to go from 0 to 2π so that Eq. (8.31) gives x and y in the range $-\infty$ to ∞.
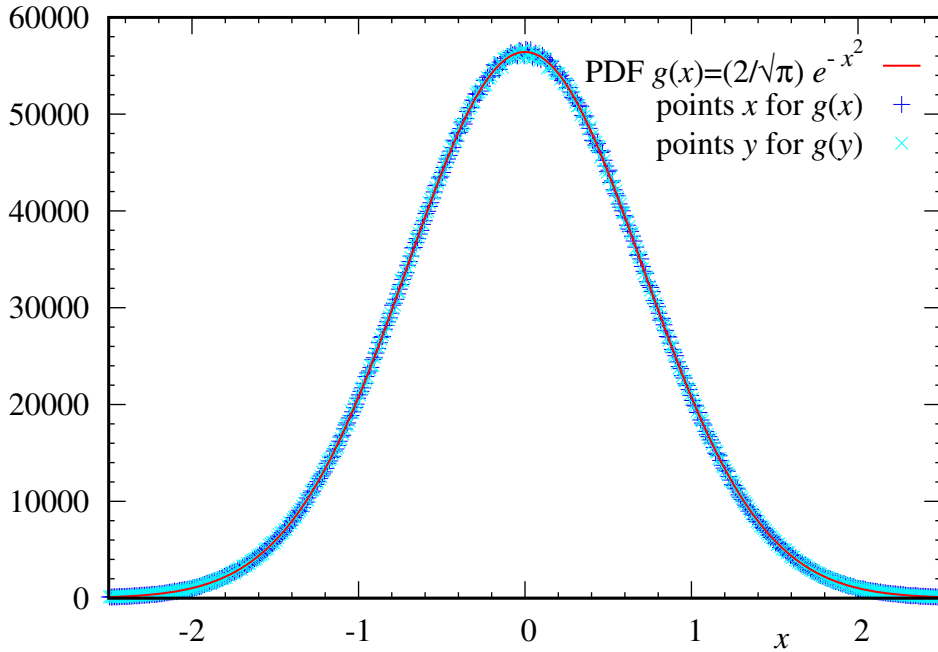
Figure 8.2: Example of sampling a Gaussian using random numbers.

This procedure is named after Box-Muller and yields two normally distributed random variables from two uniformly distributed ones.

Brute Force inversion of $F(x)$

One can always numerically integrate $g(x)$ and invert $F(x)$ but this is often not very satisfactory especially if the PDF is changing rapidly. It is necessary to compute a table of x in even steps of F

The rejection technique

For sampling a variable over a finite range, the rejection technique is often very useful. To set up this method it is first necessary to find the maximum g_{max} of the PDF $g(x)$ and renormalize to form the quantity

$$\tilde{g}(x) = \frac{1}{g_{max}} g(x) \tag{8.32}$$

Procedure:
1) Choose a random value of x uniformly in the range $a \leqslant x \leqslant b$.
2) Choose a second uniformly distributed random number y.
3) If $\tilde{g}(x) \geqslant y$ use the value of x picked. Otherwise repeat from step 1.

In this method, we use evenly distributed points (x, y) and reject them if they are are not below the curve $p(x)$.
As a high fraction of rejected points means low computational efficiency,

instead of using uniformly distributed random numbers for y we can find a PDF $\bar{g}(x)$ for which the CDF $\bar{F}(x)$ is invertible and with $g(x) < \bar{g}(x) \forall x$. Then the fraction of rejected points is only the ratio of the areas below $g(x)$ and $\bar{g}(x)$, respectively.

The Metropolis algorithm

The techniques discussed so far have the drawback that they are difficult to apply to the sampling of multidimensional functions. The Metropolis algorithm is designed to do just that, and without it, Monte Carlo methods wouldn't be as powerful as they are.

The algorithm is based on the concept of a random walk. In the two-dimensional classical random walk a point is started at the origin of an xy coordinate system, and during successive time intervals is allowed to move one unit in any (rectangular) direction with equal probability. The probabilistic description of this process is that of an expanding Gaussian probability distribution which, in the limit of some very long time, would cover the entire two-dimensional space.

If a similar random walker is started off at some point in a multi-dimensional space (a walker is specified by giving the values of all of the coordinates in space) and allowed to take a vectorial step (*i.e.* all components are changed at once) he will eventually find his way into all parts of the space. However, for the present application, we don't want the walker to simply cover all parts of the space equally, we would like him to spend more time in those regions where the function to be sampled is large. In this way the ensemble of the positions of the walker provide a **realization** of the PDF since the probability of finding a walker in a given region should be the same as $g(x_1, x_2, \ldots, x_N)$. In other words, if one could bin up all of the positions of all of the steps of the walker the probabilities would follow the PDF. Now we have to find a way to modify the rules of the walk so that this occurs. We will first give the algorithm and then discuss why it works.

Suppose we wish to represent some multi-dimensional PDF $g(\vec{X})$. For a walker, say at some point $\vec{X}$, we propose a random step (of a uniform limit δ in each coordinate) to some point $\vec{X}'$. For example, if $\vec{X}$ is a three-

dimensional variable for M particles then the movements would be

$$(x_1', y_1', z_1') = (x_1 + \delta\{-1, 1\}, y_1 + \delta\{-1, 1\}, z_1 + \delta\{-1, 1\})$$
$$(x_2', y_2', z_2') = (x_2 + \delta\{-1, 1\}, y_2 + \delta\{-1, 1\}, z_2 + \delta\{-1, 1\})$$
$$\vdots \qquad \vdots$$
$$(x_M', y_M', z_M') = (x_M + \delta\{-1, 1\}, y_M + \delta\{-1, 1\}, z_M + \delta\{-1, 1\}) \quad (8.33)$$

where each $\delta\{-1, 1\}$ corresponds to a different random value from $[-1, 1]$. After proposing a step in this manner, the value of the desired PDF is computed at the new and old positions giving $g(\vec{X})$ and $g(\vec{X'})$ and the ratio

$$q \equiv \frac{g(\vec{X'})}{g(\vec{X})} \qquad (8.34)$$

is formed. The step is then accepted with probability q. This means that if q is greater than one, it is accepted. If q is less than one, a random number uniformly distributed between 0 and 1 is generated and only if q is greater than this number is the step accepted. This algorithm was proposed in N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, *Equation of State Calculations by Fast Computing Machines*, J. Chem. Phys. **21**, 1087 (1953).

The step may be proposed in a number of ways, not just the described one. In practical cases several walkers are carried along simultaneously in order to cover the space so that an estimate of the value of the quantity being calculated can be made at each step. One would try to start them off in a distribution as closely resembling the expected final distribution as possible.

To understand why the Metropolis algorithm works think of the flow of probability like the flow of something material. There are some areas more dense than others with a constant flux of material between the areas. We have to create a flow such that the equilibrium distribution is the one that represents $g(\vec{X})$. We would like to find a method to control the flow of probability in this dynamic equilibrium such that we reach a steady state as quickly as possible.

Consider the conditions needed once the equilibrium state is reached. if $p(\vec{X'}, \vec{X})$ is the probability of a flow from $\vec{X}$ to $\vec{X'}$ and $g(\vec{X})$ is the PDF at $\vec{X}$ then, since the large probability of flow from low density to high density must match the small probability of flow from high density to low density

(so that the densities remain constant), we must have

$$p(\vec{X}', \vec{X})g(\vec{X}) = p(\vec{X}, \vec{X}')g(\vec{X}') \tag{8.35}$$

This equation simply states that the rate of flow toward high density over the rate of flow away from high density is the inverse of the ratio of the densities. It is called detailed balance.
We can now break the probability $p(\vec{X}', \vec{X})$ into the product of two factors corresponding to a conditional probability.
We shall express this probability by first proposing the change $\vec{X} \rightarrow \vec{X}'$ and then making a test to see if the proposed change is accepted. Thus we write

$$p(\vec{X}', \vec{X}) = a(\vec{X}', \vec{X})t(\vec{X}', \vec{X}) \tag{8.36}$$

where $t(\vec{X}', \vec{X})$ is the probability that we propose a point near $\vec{X}'$ starting at $\vec{X}$ and $a(\vec{X}', \vec{X})$ is the probability that such a "move" will be accepted. Any combination $a\,t$ which satisfies Eq. (8.35) will work, some may work faster than others, some may be easier to implement than others. For simplification we choose

$$t(\vec{X}', \vec{X}) = t(\vec{X}, \vec{X}') \tag{8.37}$$

The balance equation (8.35) is now reduced to

$$a(\vec{X}', \vec{X})g(\vec{X}) = a(\vec{X}, \vec{X}')g(\vec{X}') \tag{8.38}$$

It only remains for us to pick a function a which satisfies this equation. Let's define the quantity

$$q(\vec{X}', \vec{X}) = \frac{g(\vec{X}')}{g(\vec{X})} \tag{8.39}$$

Now consider

$$a(\vec{X}', \vec{X}) \equiv \min[1, q(\vec{X}', \vec{X})] \tag{8.40}$$

By examining Eq. (8.38) for the case that $q < 1$ and $q > 1$ it can be seen that it is satisfied in both situations. The condition for acceptance is indeed the one of the Metropolis algorithm.

There is an alternative which is called the heat bath method:

$$a(\vec{X}', \vec{X}) \equiv \frac{q(\vec{X}', \vec{X})}{1 + q(\vec{X}', \vec{X})} \tag{8.41}$$

The vector containing the walkers represents the PDF in the sense that more walkers are concentrated in the region where the function is large. This can be a very useful way to represent a function. For simple functions it may seem like a poor expression but for multi-dimensional functions it is very efficient. If we wish to average over a function, the values are only needed where it is large. The usual representation of a function at all points in space becomes incredibly wasteful. Consider a mesh of only 10 points in each dimension. For thirty particles in three dimensions we would need 10^{90} tabulated values, impossible on any computer. With the Monte Carlo representation, problems with the order of a thousand particles can be solved.