Metropolis Monte Carlo

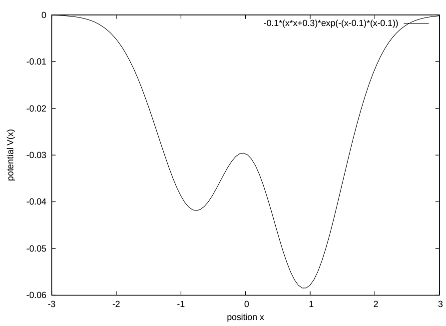
A Tutorial in a Simple System

Wolfhard Janke, Sandro Wenzel

21st September 2006

This introductory exercise is meant to give you a first demonstration and experience with the Metropolis algorithm. It is meant for all those of you who have never implemented such an algorithm before.

We will consider a very simple model in which a particle is trapped in a potential.



Your task is to implement a simple program and determine thermodynamic observables like the mean energy, heat capacity, mean location of the particle for different temperatures T.

Prerequisite is basic knowledge of a programming language (C or Fortran). The theory of the Metropolis algorithm is covered in the lecture notes (available from us).

Physical setup: We consider a classical particle confined in a 1-dimensional potential given by the following equation

$$V(x) = -0.1 * (x^2 + 0.3) * \exp(-(x - 0.1)^2) \tag{1}$$

The particle is connected to a heat bath at temperature T and therefore obeys Boltzmann statistics. The probability to find the particle at position x is of course given by

$$p(x) = \frac{1}{Z} \exp(-\beta * V(x)) \tag{2}$$

Verbal description of algorithm: The following steps should guide and help you in implementing the program:

1. setup the program with all important variables like current position. Implement a function that calculates the potential energy $V(x)$ for each x.

2. initialise starting position of your particle (e.g. $x_{\text{cur}} = x_0 = 0$)

3. calculate its current energy $E_{\text{cur}} = V(x_{\text{cur}})$

4. propose a new position

$$x_{\text{new}} = x_{\text{cur}} + (r - 0.5)s\,.$$

Here $r \in [0, 1]$ is a random number and s is some small stepsize you should use in your program. Try $s = 0.01$ first.

5. calculate energy of the proposed position $E_{\text{new}} = V(x_{\text{new}})$.

6. accept the new position with the Metropolis acceptance probability

$$\exp\left(-\beta(E_{\text{new}} - E_{\text{cur}})\right)\,. \tag{3}$$

Acceptance means that the particle position changes to $x_{\text{cur}} = x_{\text{new}}$. This can be done by using another random number $r \in [0, 1]$ and compare it with (3).

This step corresponds to the mentioned Markov process covered in the lecture. You go from one state to another state with the Metropolis probability

$$W(x_{cur} \longrightarrow x_{new}) = \min\left(1, \exp(-\beta(E_{new} - E_{cur}))\right)$$

7. record some quantities (position, energy) for measurement in program variables.

8. go back to step 3. Repeat a couple of times until you have enough statistics.

9. In the end calculate the mean values of all observable and print them.

Tasks:

1. Plot the Monte Carlo evolution of the positions x with gnuplot at $T = 0.05$. Do you get the same behaviour when you initially set your particle to $x_0 = -3$?

2. At each Monte Carlo step record the the current energy, the current position. Calculate the expectation value of those observables using these values.

3. Repeat the simulation for different T and draw a curve of the mean energy, position and optionally the heat capacity in dependence of T. Are your results physical?

4. Try and compare you results with numerical integration.

5. Add a second particle to the system. The particles interact with each other with the potential $V(x_1, x_2) = -\frac{1}{10}(x_1 - x_2)^2$ What is the mean distance between the particles at $T = 0.1$?