Computational Physics Homework Report

Fateme RamezanZade-99100693 Problem Set 7

Random Number Generator with particular distribution: Metropolis Algorithm

1. Functions Description:

P(x):

This function accounts for the distribution function that we want our random numbers have; which is a gaussian function with $\sigma = 1$ in our case. It returns the value of $p(x)$ for its argument a x.

C(j,arr_x):

This function calculates the autocorrelation function of a list of numbers (array arr_x) for given length as j using this formula:

$$c(j) = \frac{\left|\langle x_i x_{i+j}\rangle_i - \langle x_i\rangle_i \langle x_{i+j}\rangle_i\right|}{\sigma^2}$$

2. Main Code

   Part 1

   Firstly, I have defined x with initial value 0, because my gaussian function has its maximum value there. Then I defined delta as the step length, N as the number of steps and an array (x_arr) to store the value of x after each accepted step. Now a loop starts, which repeats this process in every turn: first decides weather the step is taken towards right or left, with equal probabilities, using a random number. Then calculates the place after taking the step, which is y. now picks a random number (r) in range(0,1) and compares it with the ratio of the value of distribution in both places. If the step is accepted, the value of x would be updated with y and the array would be appended with it. Now the ratio of this array's length minus 1 to N is printed as the acceptance rate, and a histogram is plotted using its elements.

   Part 2

   The previous code is repeated in a loop with different step lengths (from 0.1 to 20, adding 0.1 each time) and the acceptance rate corresponding to each step is stored in an array.

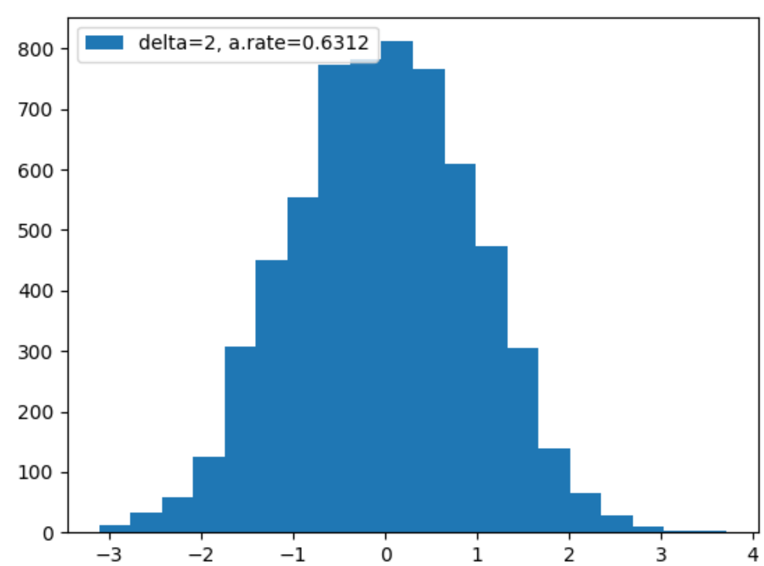
   Part 3

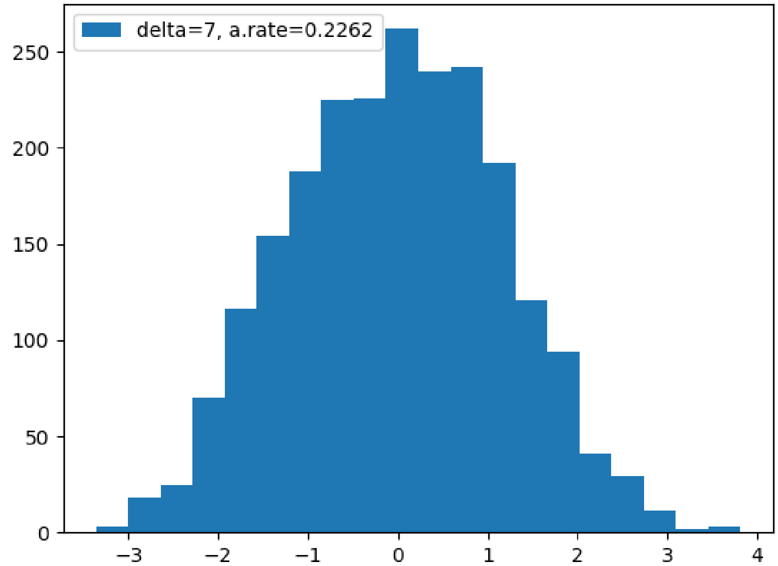
   The metropolis algorithm is repeated and in each round, the autocorrelation function is calculated for different j in range (1,10) using the function c(j,arr_x). After the loop ends, the average value of c is plotted for each j. in order to calculate correlation time, a line is fitted to ln(c(j))-j function.

3. Results

   Part 1

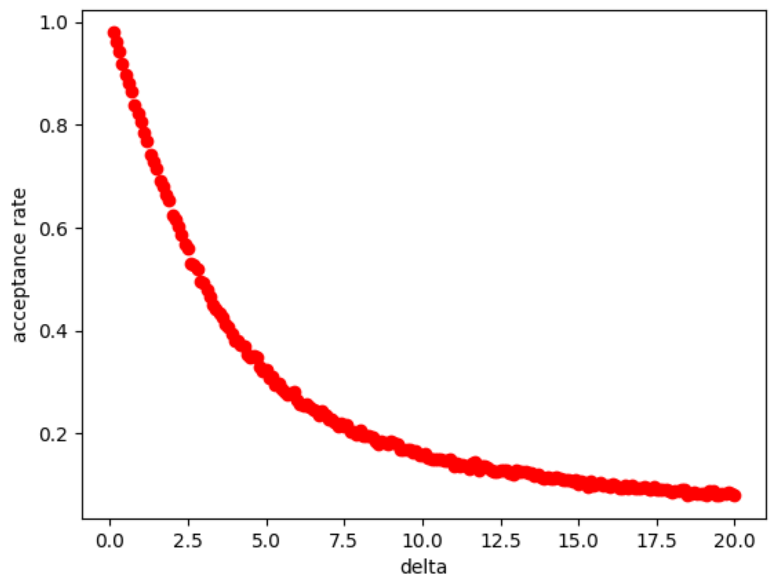
   The number of metropolis steps(N)=10000





Part 2

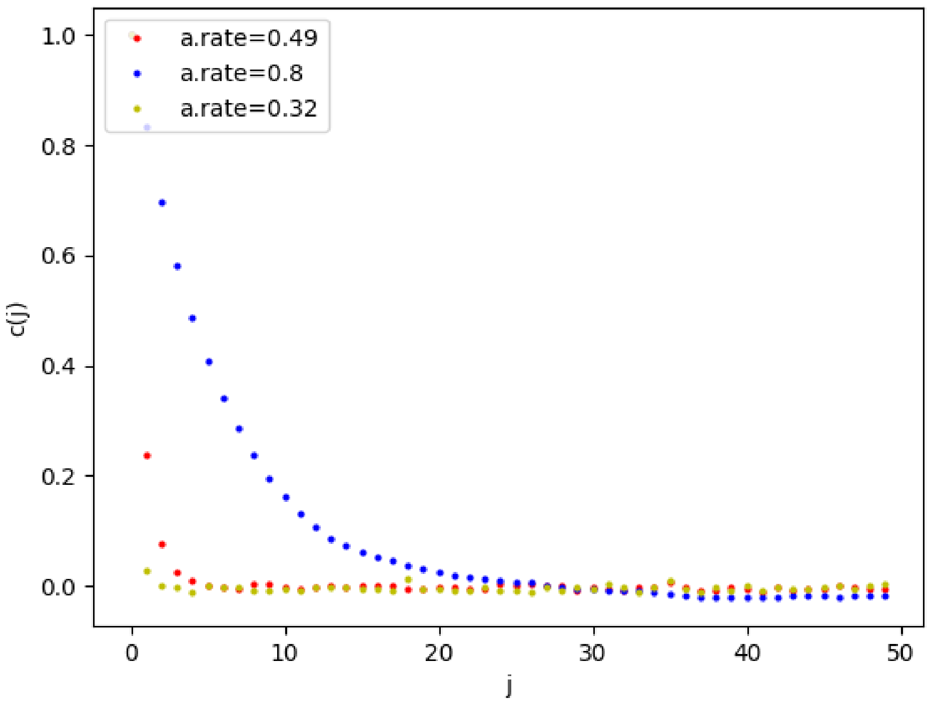
delta	0.5	1	1.6	2.2	2.9	3.9	5.3	8.1	15.8
a. rate	0.8974	0.8052	0.6889	0.604	0.4954	0.41	0.2951	0.1954	0.0998



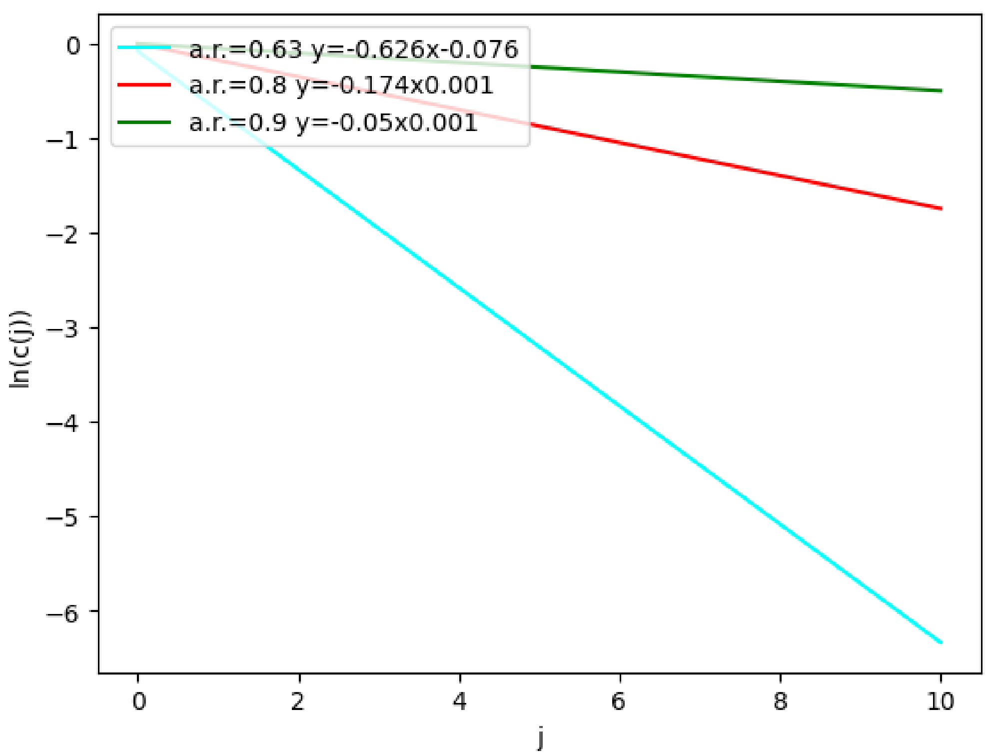
Part 3

(I wasn't able to calculate the logarithm function for very small quantities, therefore I didn't calculate the correlation time for low acceptance rates.

Cor. time	20.48	4.92	2.07	1.584	0.84	0.59			
a. rate	0.8974	0.8052	0.6889	0.604	0.4954	0.41	0.2951	0.1954	0.0998



a.rate=0.49
a.rate=0.8
a.rate=0.32
c(j)
j



a.r.=0.63 y=-0.626x-0.076
a.r.=0.8 y=-0.174x0.001
a.r.=0.9 y=-0.05x0.001
ln(c(j))
j