SHARIF UNIVERSITY OF TECHNOLOGY
PHYSICS DEPARTMENT

Problem Set 8

Student name: *Ali Fayyaz - 98100967*

Course: *Computational Physcis - (Spring 2023)*
Due date: *April 21, 2023*

Exercise 8.1

The Metropolis Algorithm:

(*a*) Use the Metropolis algorithm and create a random number generator that obeys a Gaussian distribution.

(*b*) Determine Metropolis step sizes Δ such that the acceptance rate is $a_r \in \{0.1, 0.2, 0.3, ..., 0.9\}$.

(*c*) Find the correlation length for each of the given acceptance rates.

Answer. The four functions below were declared to answer the questions:

- **guassian():**

  Basically the function for Gaussian distribution that takes a number x, μ as the mean and σ as the standard deviation of the distribution.

- **generate_gaussian():**

  The Metropolis Algorithm is implemented in this function, as it generates Gaussian random numbers using the uniform random number generator, based on the said algorithm. It takes as input a *step* for the Metropolis step size (equivalent to Δ in the textbook), a *sample_size* which determines the number of samples to be drawn from the Gaussian distribution, and *x0* as the initial position for the Metroplis algorithm. *mu* and *sigma* can also be given as parameters of the Gaussian distribution.

  The exact algorithm explained in the textbook is implemented, and in order to counter the effects of correlation, for every 100 numbers generated, one is reported as coming from a Gaussian distribution.

  The function returns randomly generated Gaussian numbers of the desired sample size, all the numbers generated (including the ones skipped because of correlation effects) and also the acceptance rate of the input Metropolis step size.

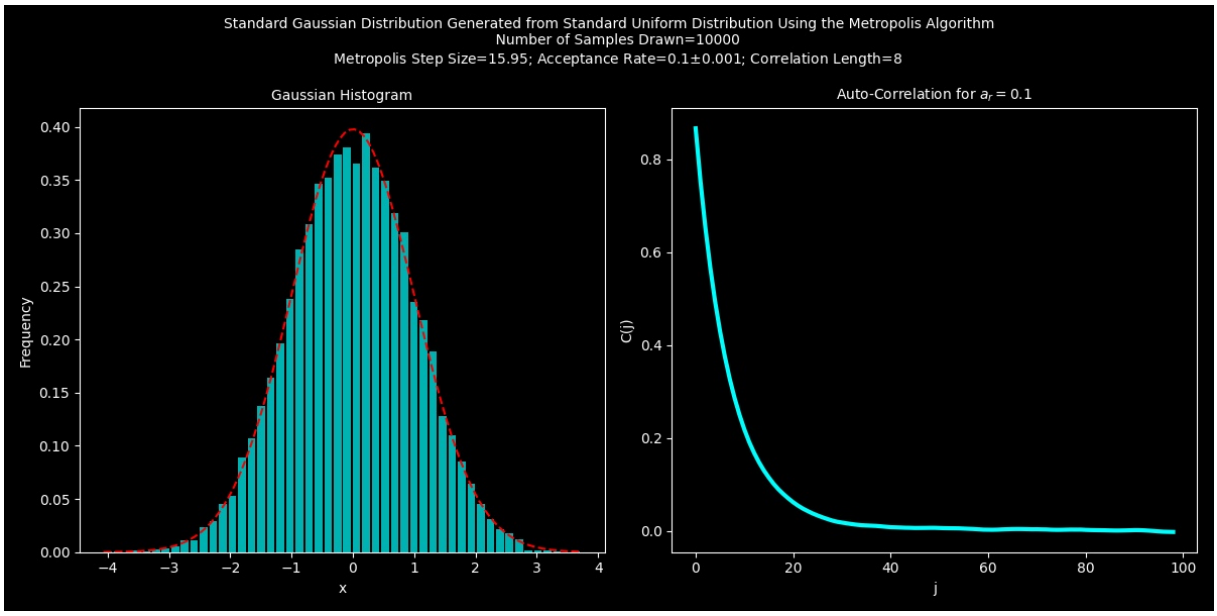
- **find_step_size():**

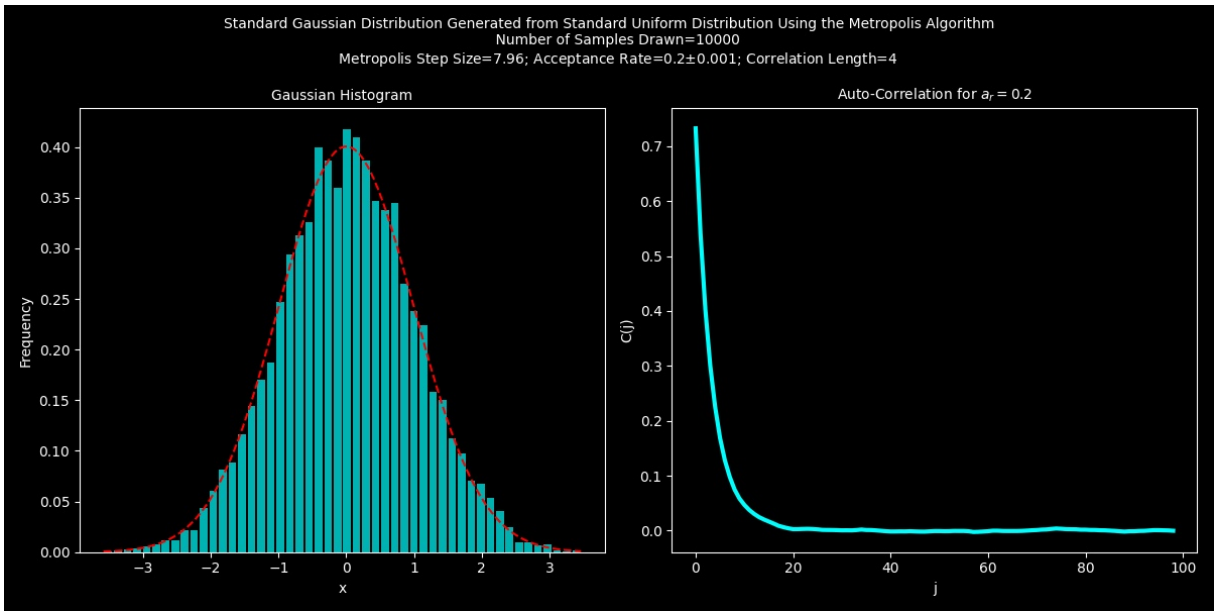
  This function finds an optimum Metropolis step size (Δ) for a given acceptence rate. It takes as input an interval, traverses it, examines different Metroplois step sizes in the said interval and utilizes *numpy.isclose()* with a given relative tolerance, and returns the first step size in the interval that pruduces numbers with an acceptance rate close enough to the desired acceptance rate.

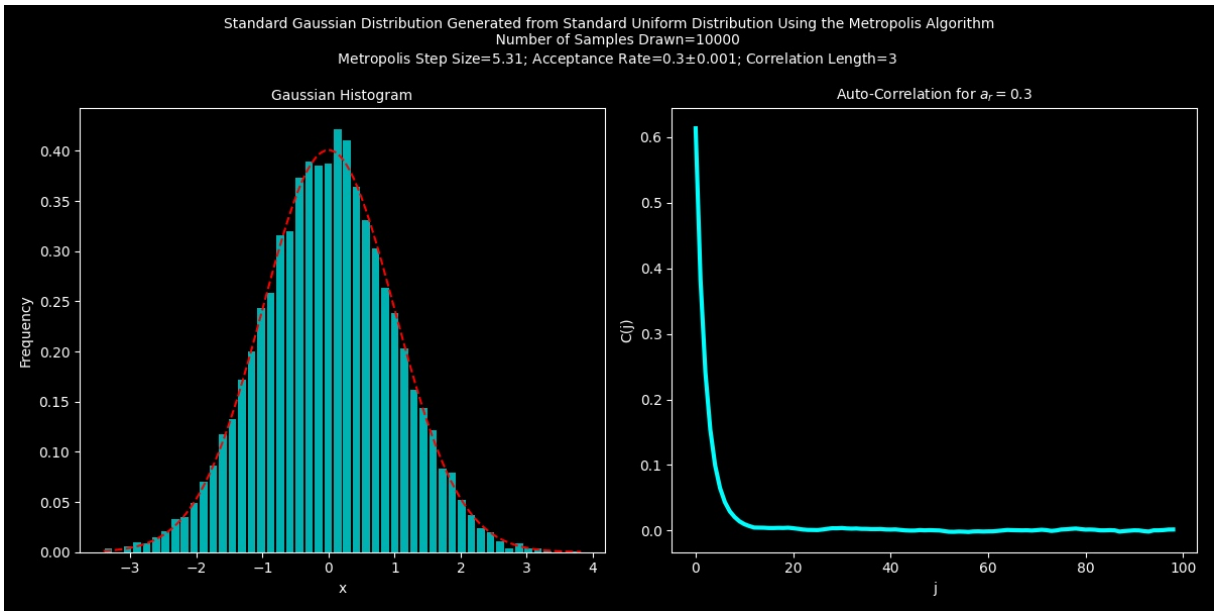
  To illustrate how this works, take for instance the desired acceptance rate is $a_r = 0.1$. With a little trial and error, one can find that the optimum Metroplis step size to generate the given a_r is in the interval (15.5, 16). One can then invoke **find_step_size()** to get a step size with the desired accuracy in the decimals. For a relative tolerance *rtol = 0.001*, for this specific example, an optimum step size is $\Delta \simeq 15.95$. All the data generated here are accurate to the second decimal place, using this function.

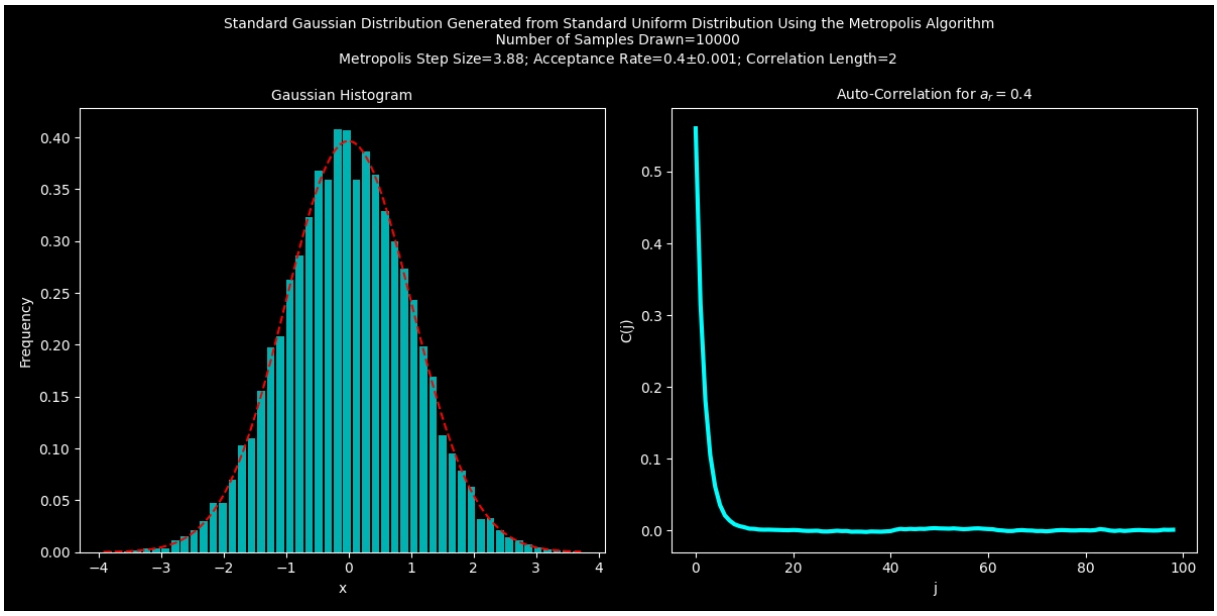
- **auto_correlation():**

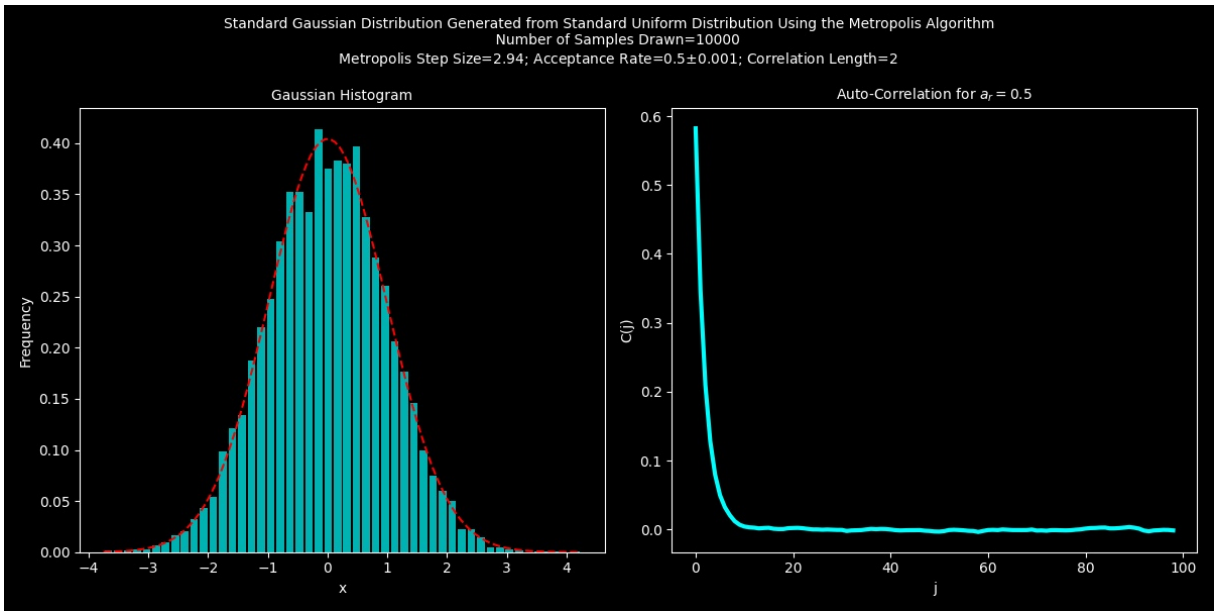
  This function calculates $C(j)$, as defined by equation 8.5 of the textbook, for every j in the set {1,2,3,...,99}. It returns two values: ξ, as the first j where $C(j) \leq e^{-1}$, and also an array containing all the values of $C(j)$ for all the j's in the set. These values will be plotted to demonstrate the exponential behavior of $C(j)$.

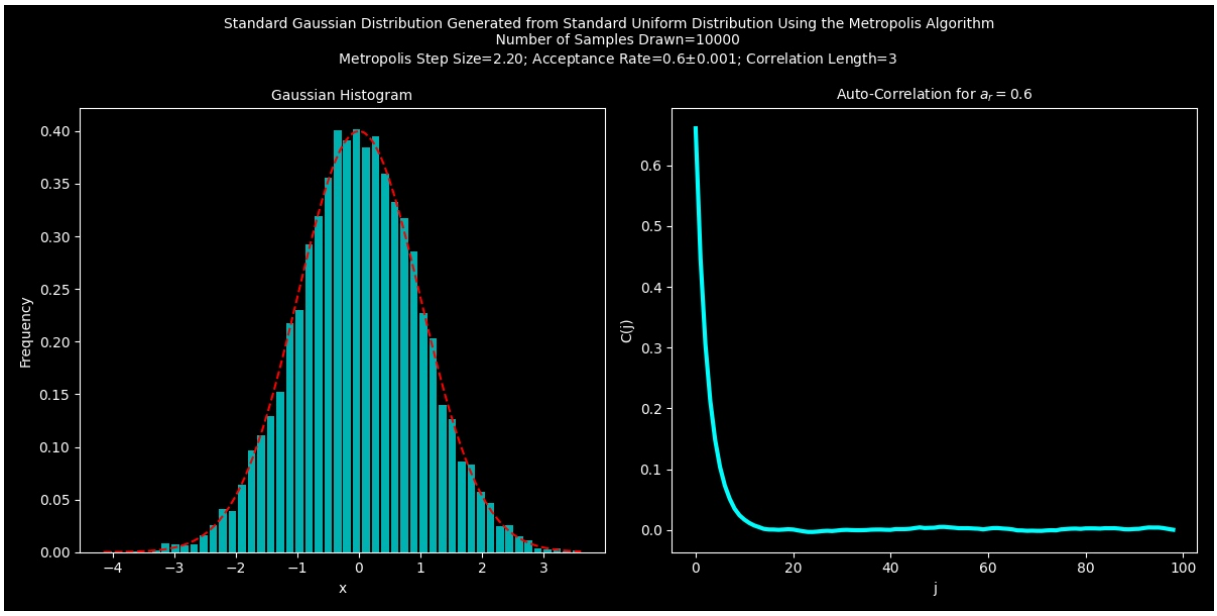

Standard Gaussian Distribution Generated from Standard Uniform Distribution Using the Metropolis Algorithm
Number of Samples Drawn=10000
Metropolis Step Size=15.95; Acceptance Rate=0.1±0.001; Correlation Length=8
Gaussian Histogram
Auto-Correlation for $a_r = 0.1$
Frequency
x
C(j)
j

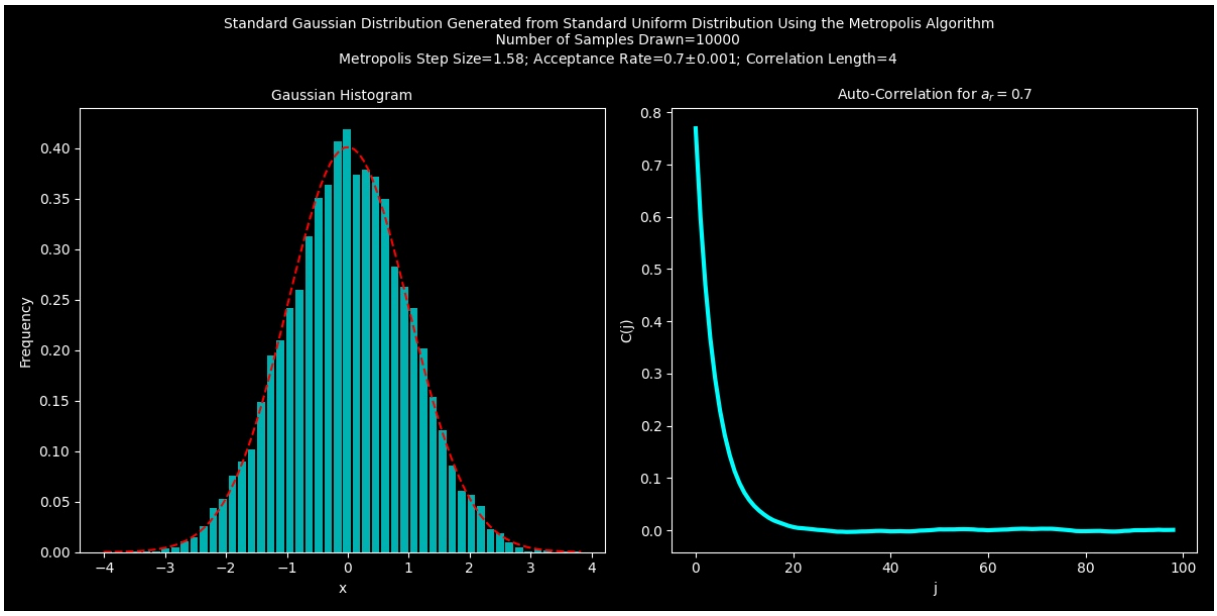

Standard Gaussian Distribution Generated from Standard Uniform Distribution Using the Metropolis Algorithm
Number of Samples Drawn=10000
Metropolis Step Size=7.96; Acceptance Rate=0.2±0.001; Correlation Length=4
Gaussian Histogram
Auto-Correlation for $a_r = 0.2$
Frequency
x
C(j)
j

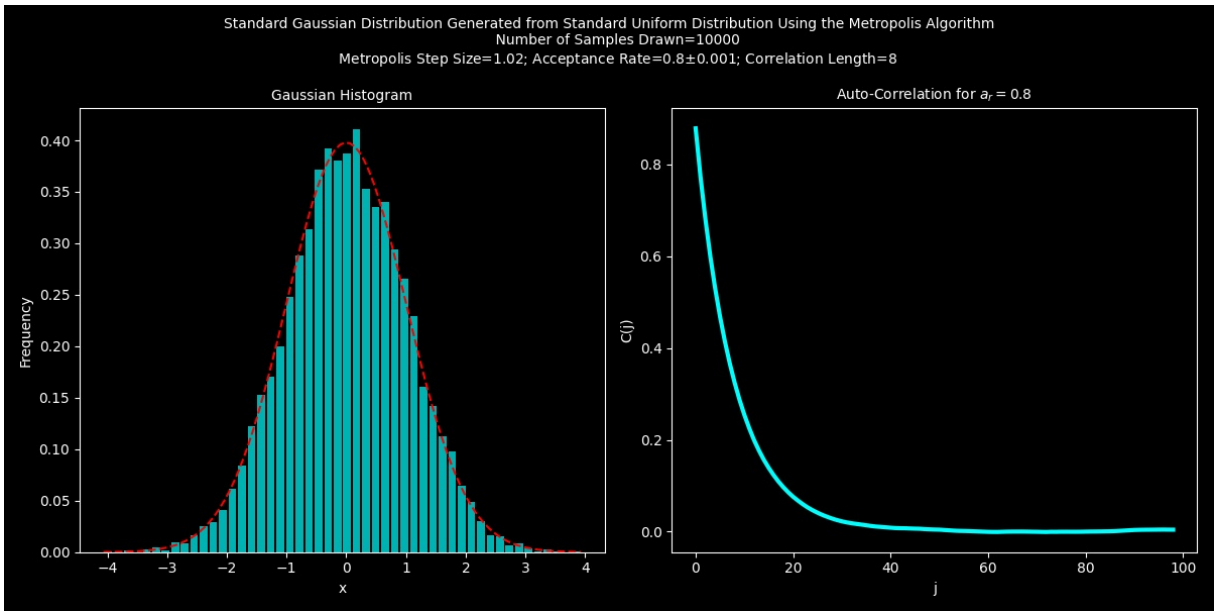

Standard Gaussian Distribution Generated from Standard Uniform Distribution Using the Metropolis Algorithm
Number of Samples Drawn=10000
Metropolis Step Size=5.31; Acceptance Rate=0.3±0.001; Correlation Length=3
Gaussian Histogram
Auto-Correlation for $a_r = 0.3$
Frequency
x
C(j)
j

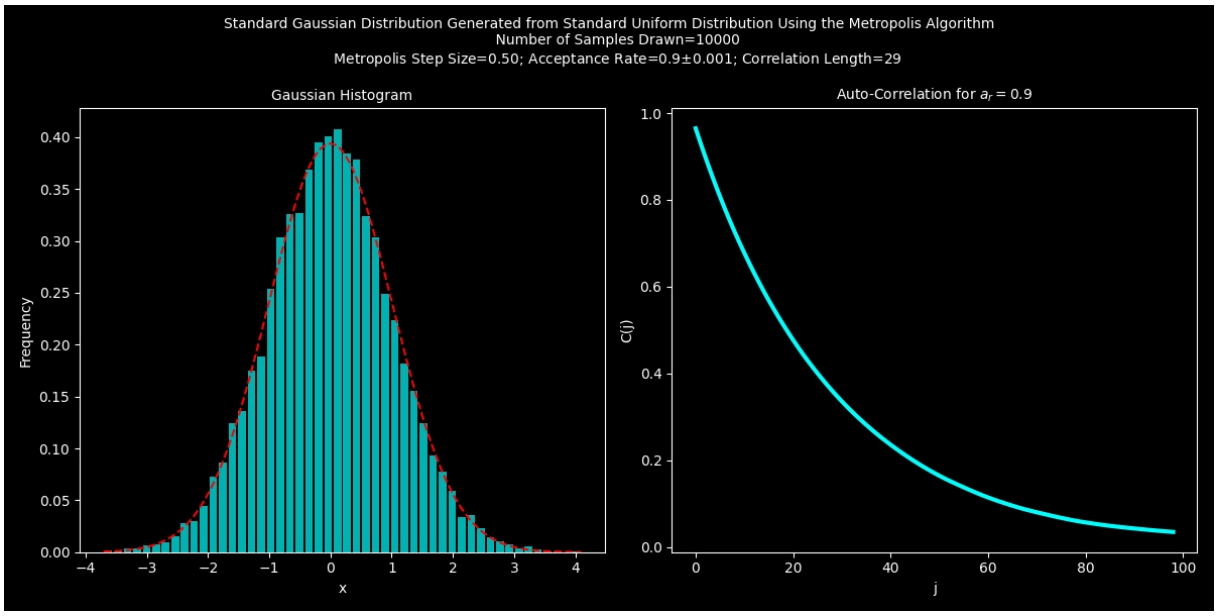

Standard Gaussian Distribution Generated from Standard Uniform Distribution Using the Metropolis Algorithm
Number of Samples Drawn=10000
Metropolis Step Size=3.88; Acceptance Rate=0.4±0.001; Correlation Length=2
Gaussian Histogram
Auto-Correlation for $a_r = 0.4$
Frequency
x
C(j)
j


Standard Gaussian Distribution Generated from Standard Uniform Distribution Using the Metropolis Algorithm
Number of Samples Drawn=10000
Metropolis Step Size=2.94; Acceptance Rate=0.5±0.001; Correlation Length=2
Gaussian Histogram
Auto-Correlation for $a_r = 0.5$
Frequency
x
C(j)
j


Standard Gaussian Distribution Generated from Standard Uniform Distribution Using the Metropolis Algorithm
Number of Samples Drawn=10000
Metropolis Step Size=2.20; Acceptance Rate=0.6±0.001; Correlation Length=3
Gaussian Histogram
Auto-Correlation for $a_r = 0.6$
Frequency
x
C(j)
j


Standard Gaussian Distribution Generated from Standard Uniform Distribution Using the Metropolis Algorithm
Number of Samples Drawn=10000
Metropolis Step Size=1.58; Acceptance Rate=0.7±0.001; Correlation Length=4
Gaussian Histogram
Auto-Correlation for $a_r = 0.7$
Frequency
x
C(j)
j


Standard Gaussian Distribution Generated from Standard Uniform Distribution Using the Metropolis Algorithm
Number of Samples Drawn=10000
Metropolis Step Size=1.02; Acceptance Rate=0.8±0.001; Correlation Length=8
Gaussian Histogram
Auto-Correlation for $a_r = 0.8$
Frequency
x
C(j)
j


Standard Gaussian Distribution Generated from Standard Uniform Distribution Using the Metropolis Algorithm
Number of Samples Drawn=10000
Metropolis Step Size=0.50; Acceptance Rate=0.9±0.001; Correlation Length=29
Gaussian Histogram
Auto-Correlation for $a_r = 0.9$
Frequency
x
C(j)
j