Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЕТ

по лабораторной работе № 2

“Формирование случайных чисел с заданным распределением”

Выполнил: И.А. Григорик

Проверил:  И.Г. Алексеев

МИНСК 2023

**1 ЦЕЛЬ**

Изучить основные способы создания последовательностей случайных чисел с плавающей запятой с заданными законами распределения вероятности.

**2 ЗАДАНИЕ**

Написать приложение для моделирования генераторов псевдослучайных чисел. Требуется реализовать выбрать два наиболее успешных алгоритмов генерации из предыдущей лабораторной работы. В данной работе используются следующие генераторы:

1. Генератор на основе алгоритма Лемера;
2. Генератор на основе метода серединных произведений.

Подобрать параметры для генераторов с целью получения наилучших результатов для каждого генератора (для выборок N = {10000, 500000, 100000000}).

Вывести гистограмму, матожидание, корреляцию и результаты распределения для лучшего случая.

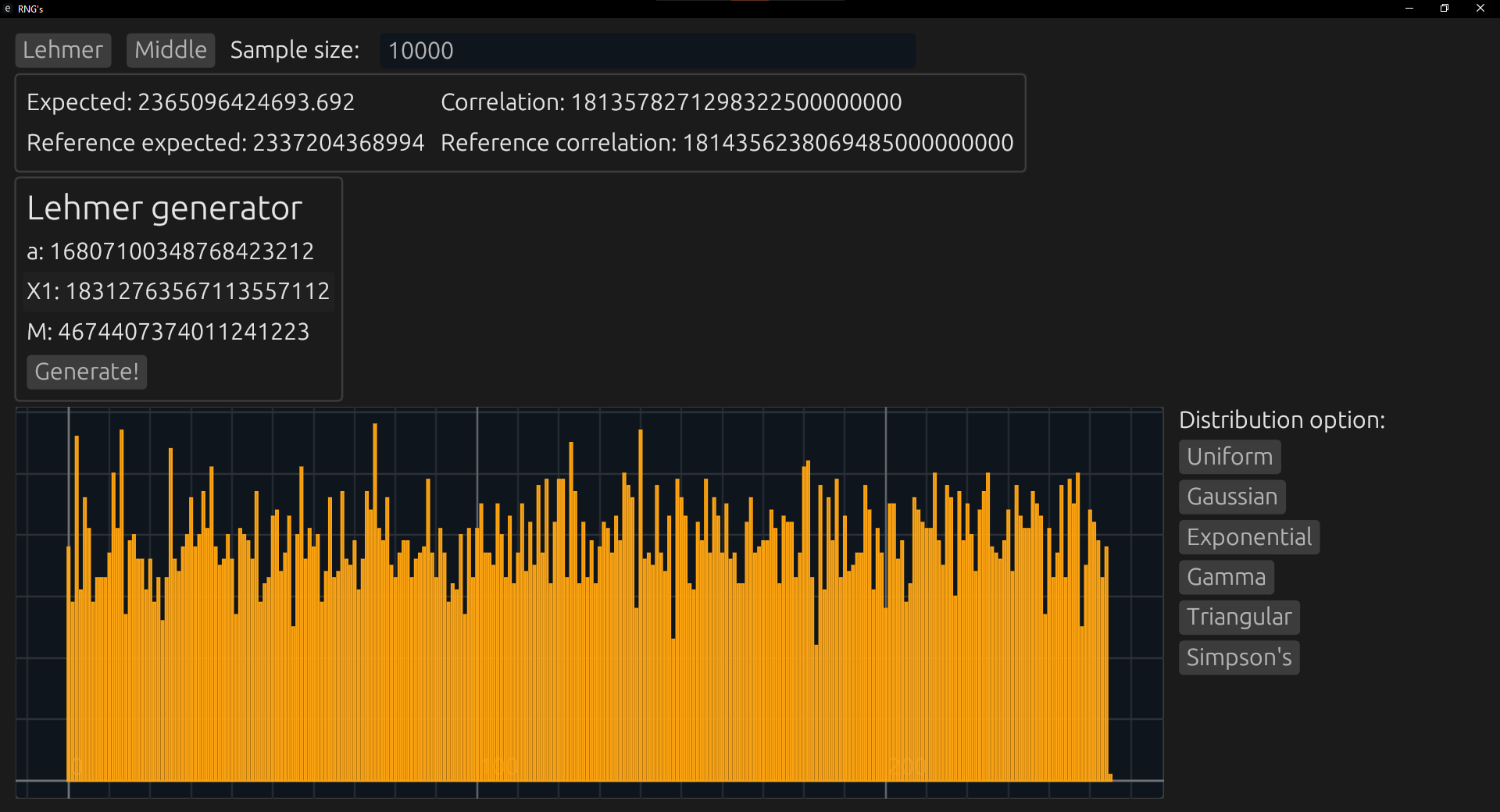
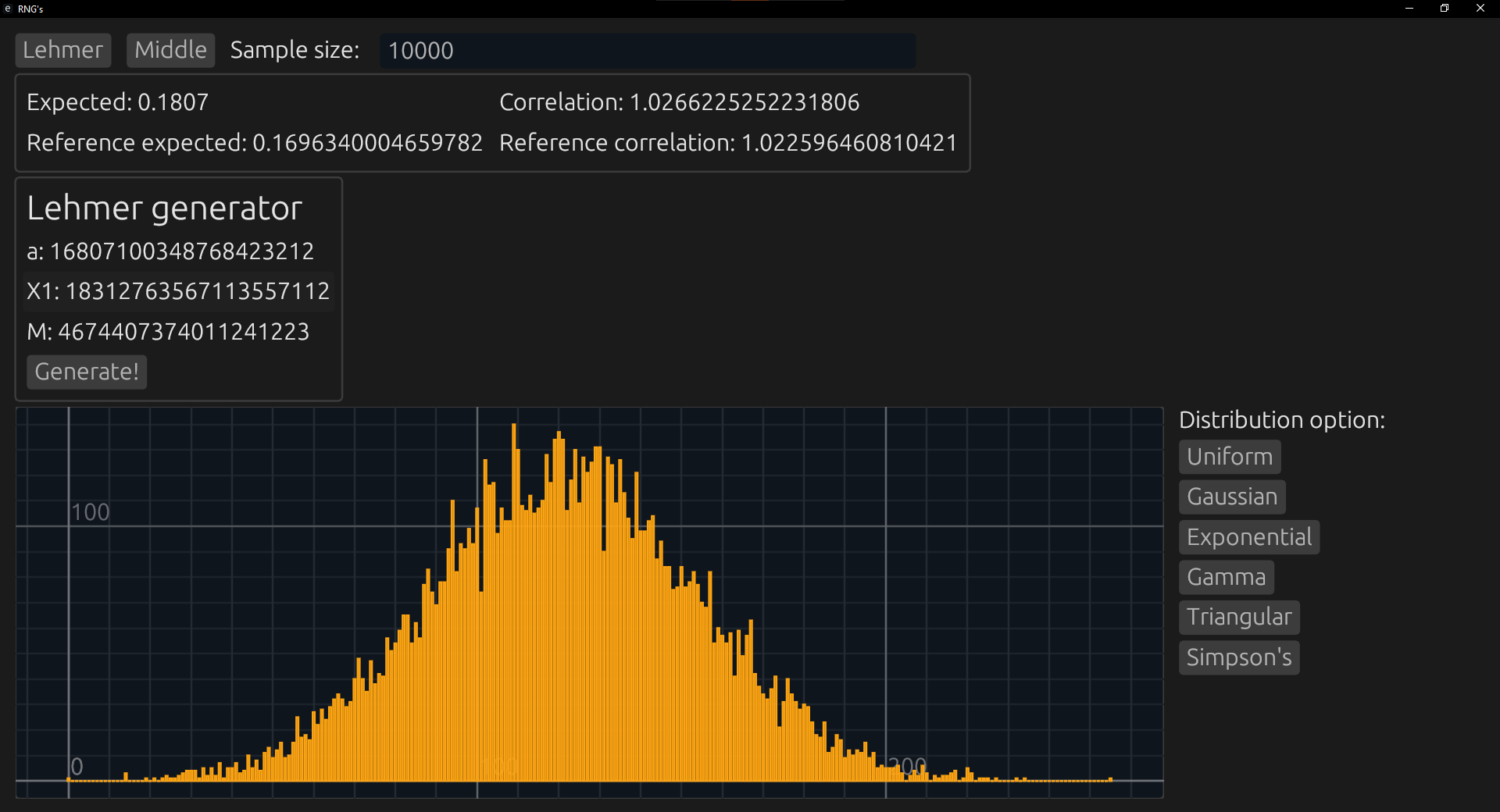
**3 ХОД РАБОТЫ**

**3.1 Генератор Лемера**

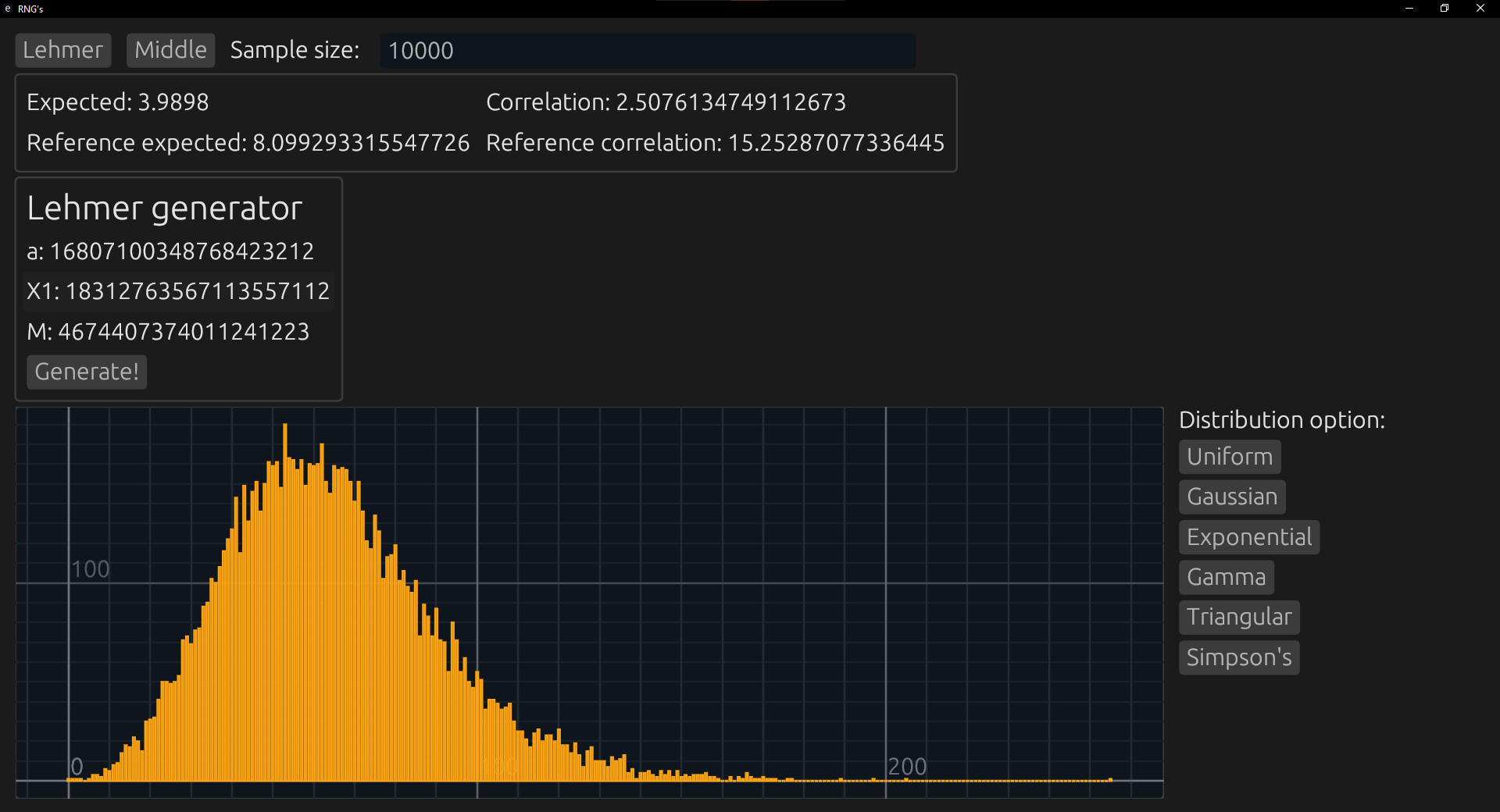
В качестве входных данных на генератор Лемера подаются значения:

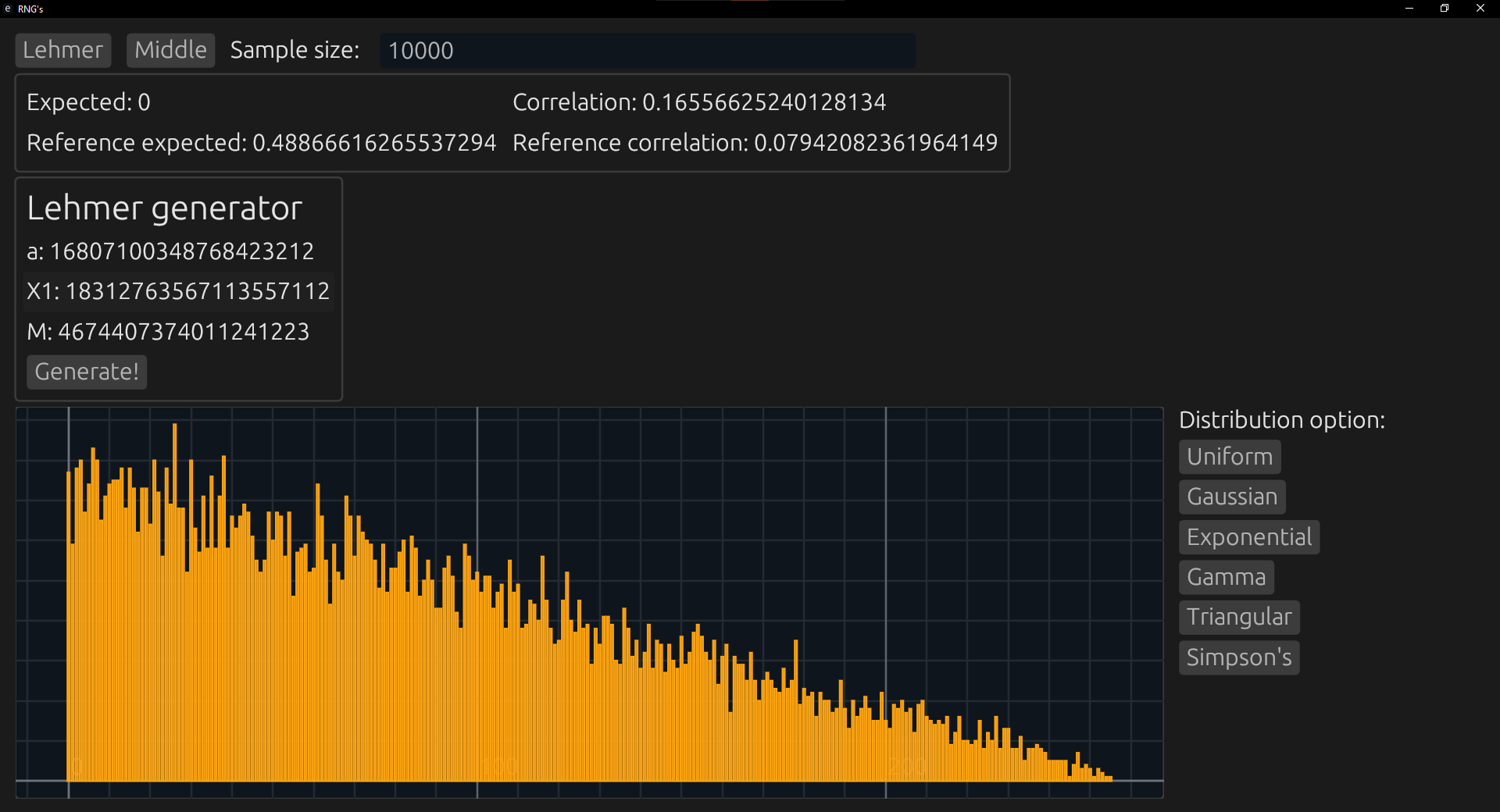
A: 11.680.710.034.876.434.312  
 X1:10.312.763.567.113.667.112  
 m: 4.674.407.374.011.241.223

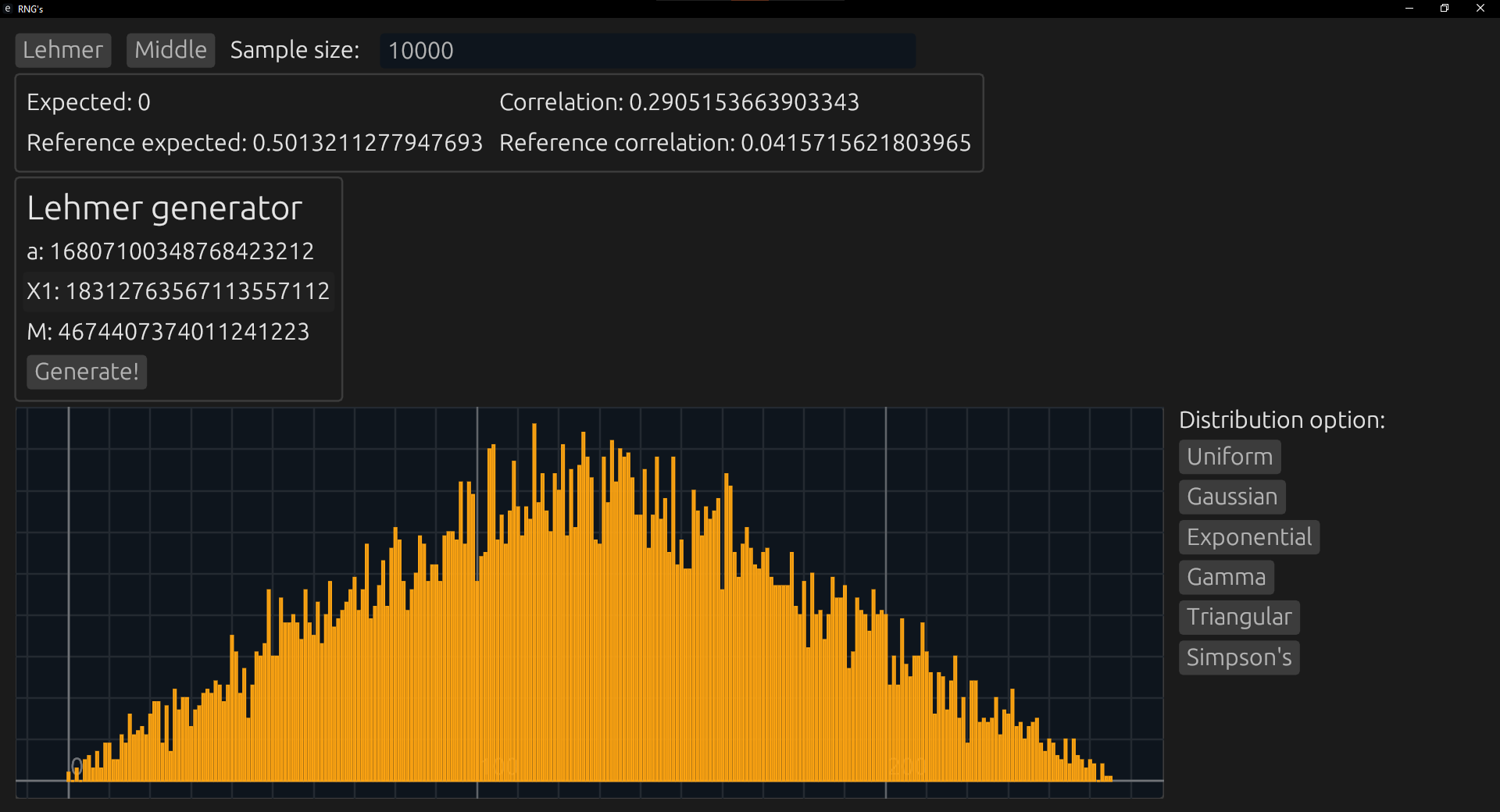
Генератор с выборкой равной n = 10000

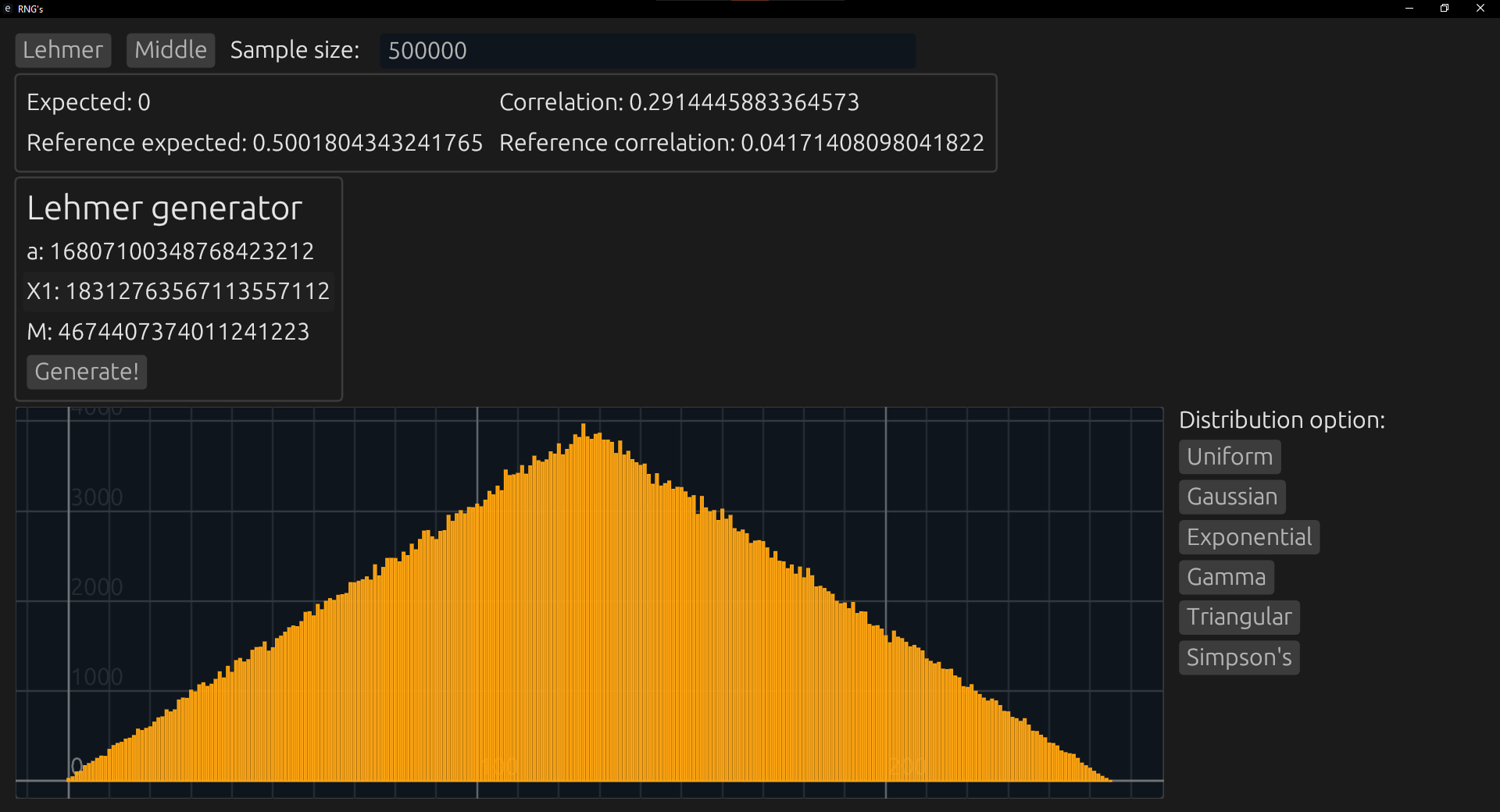
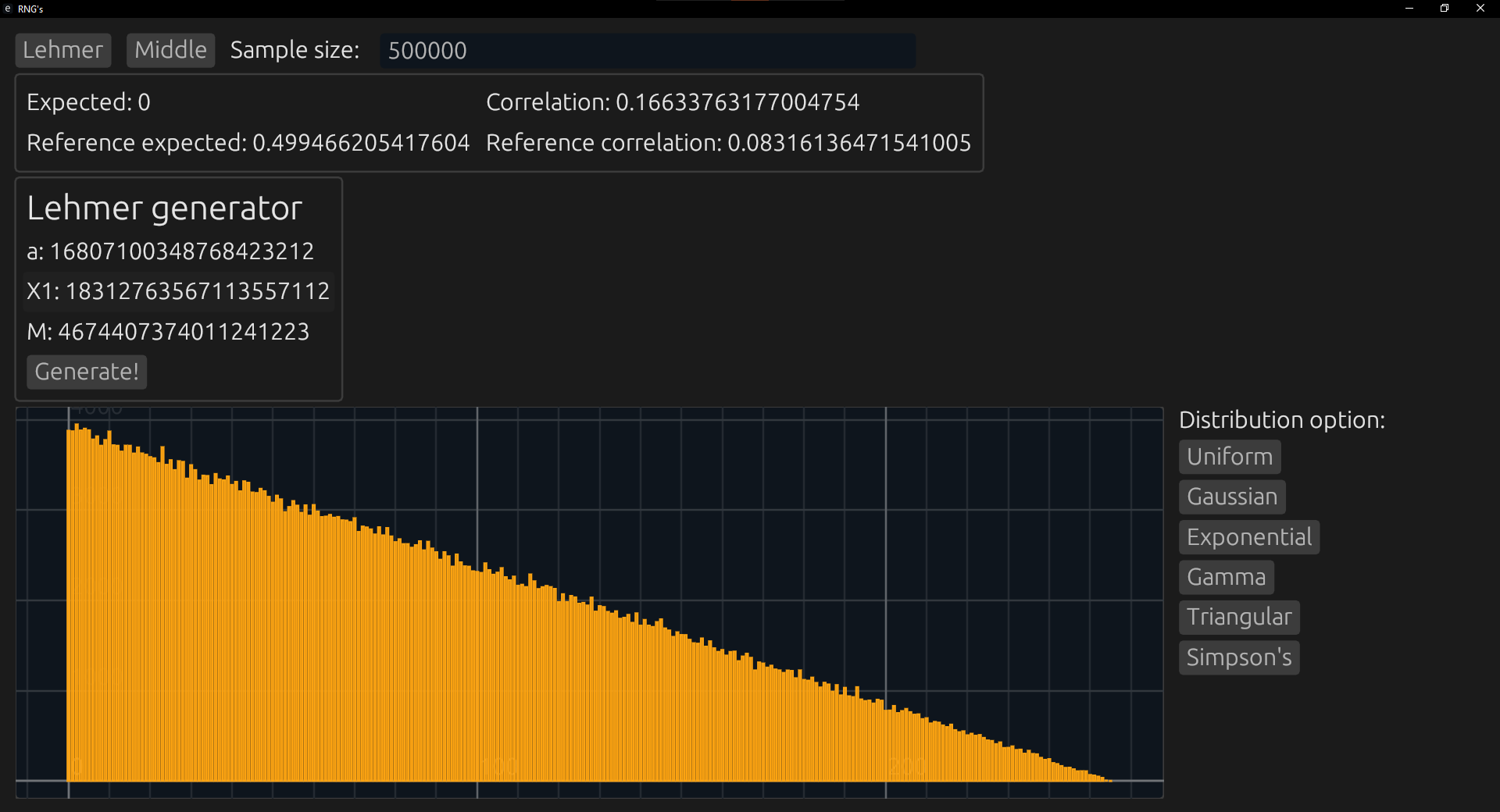
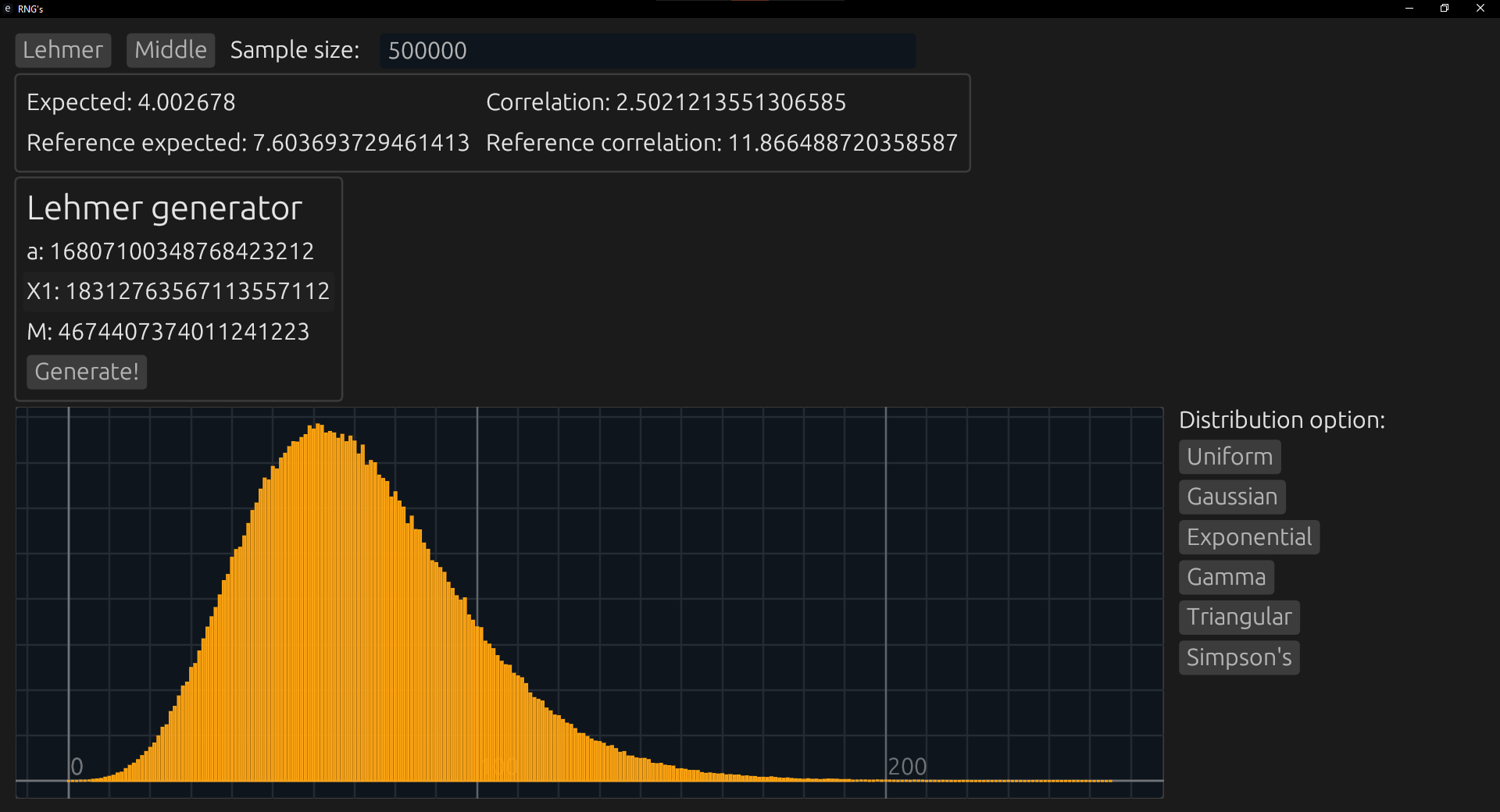
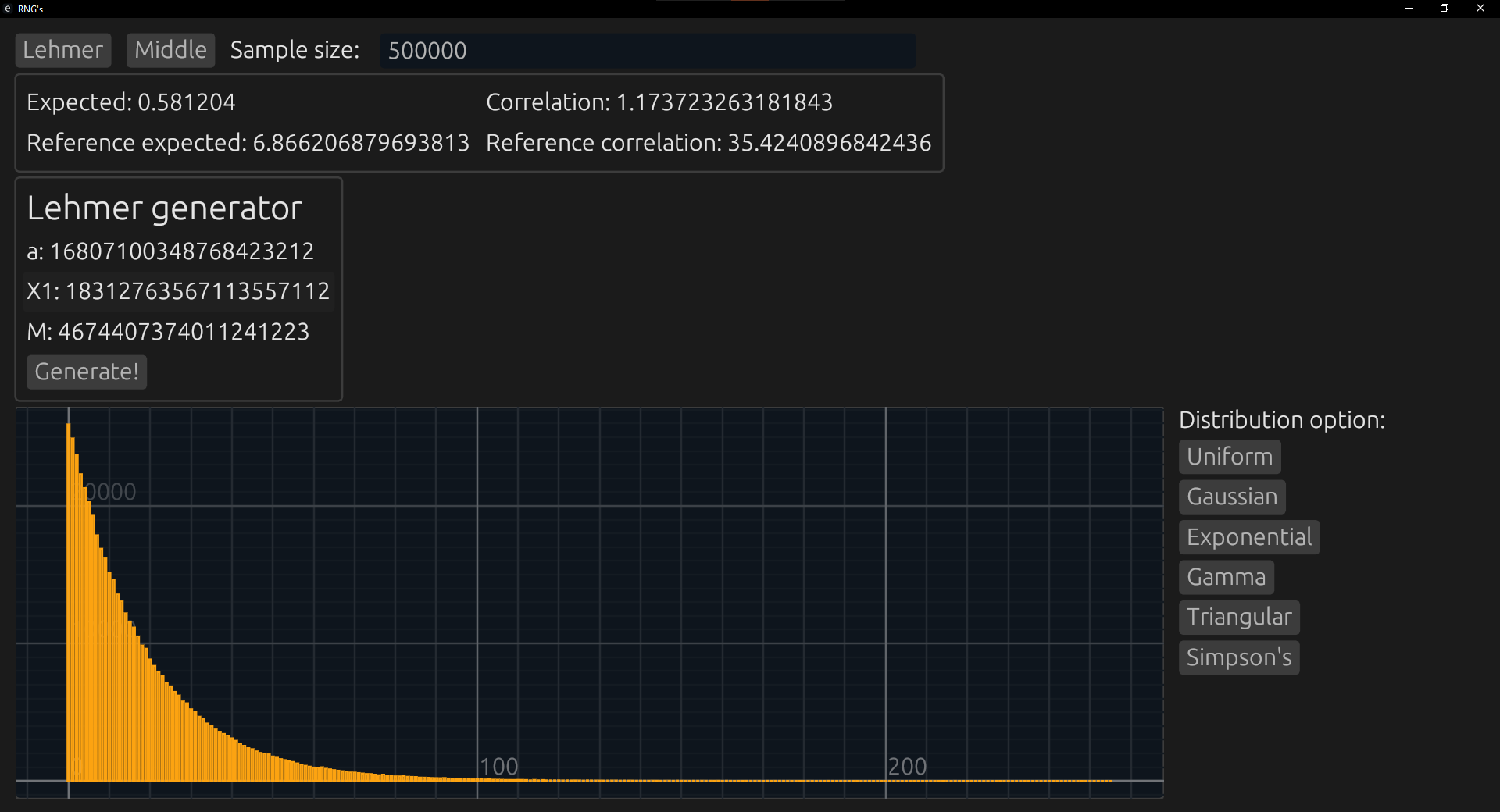
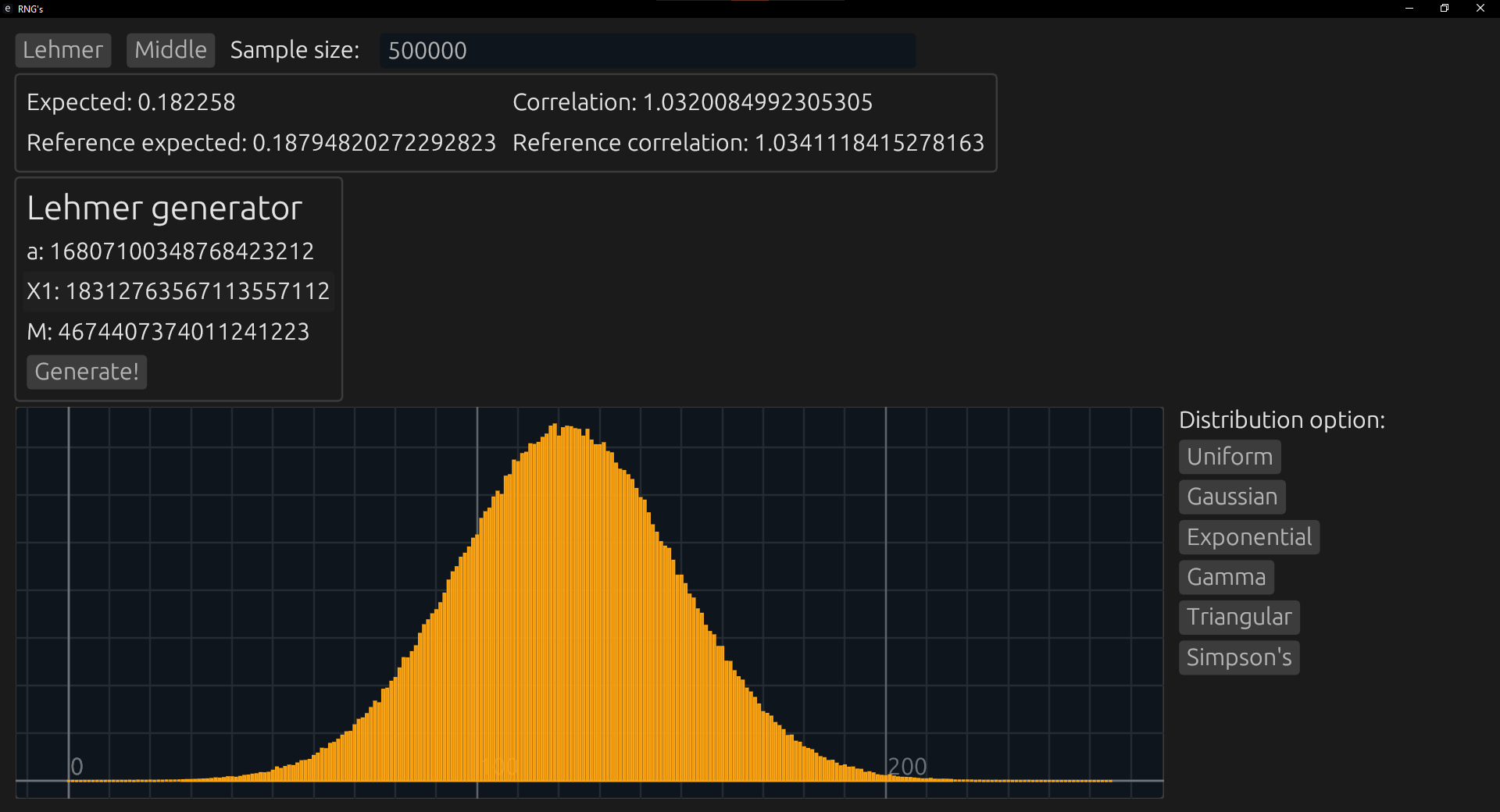




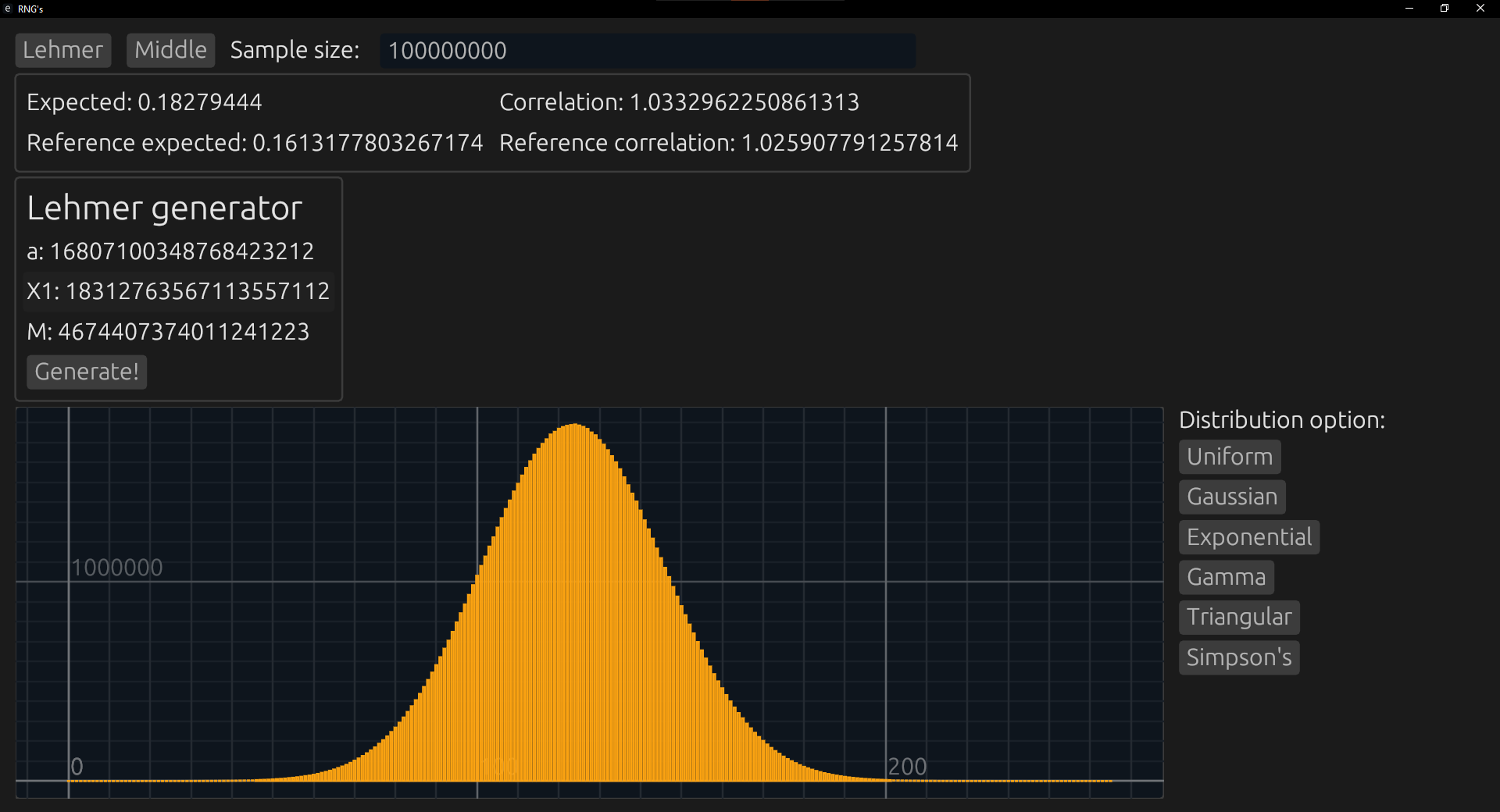
a = 452, b = 4248925791982749

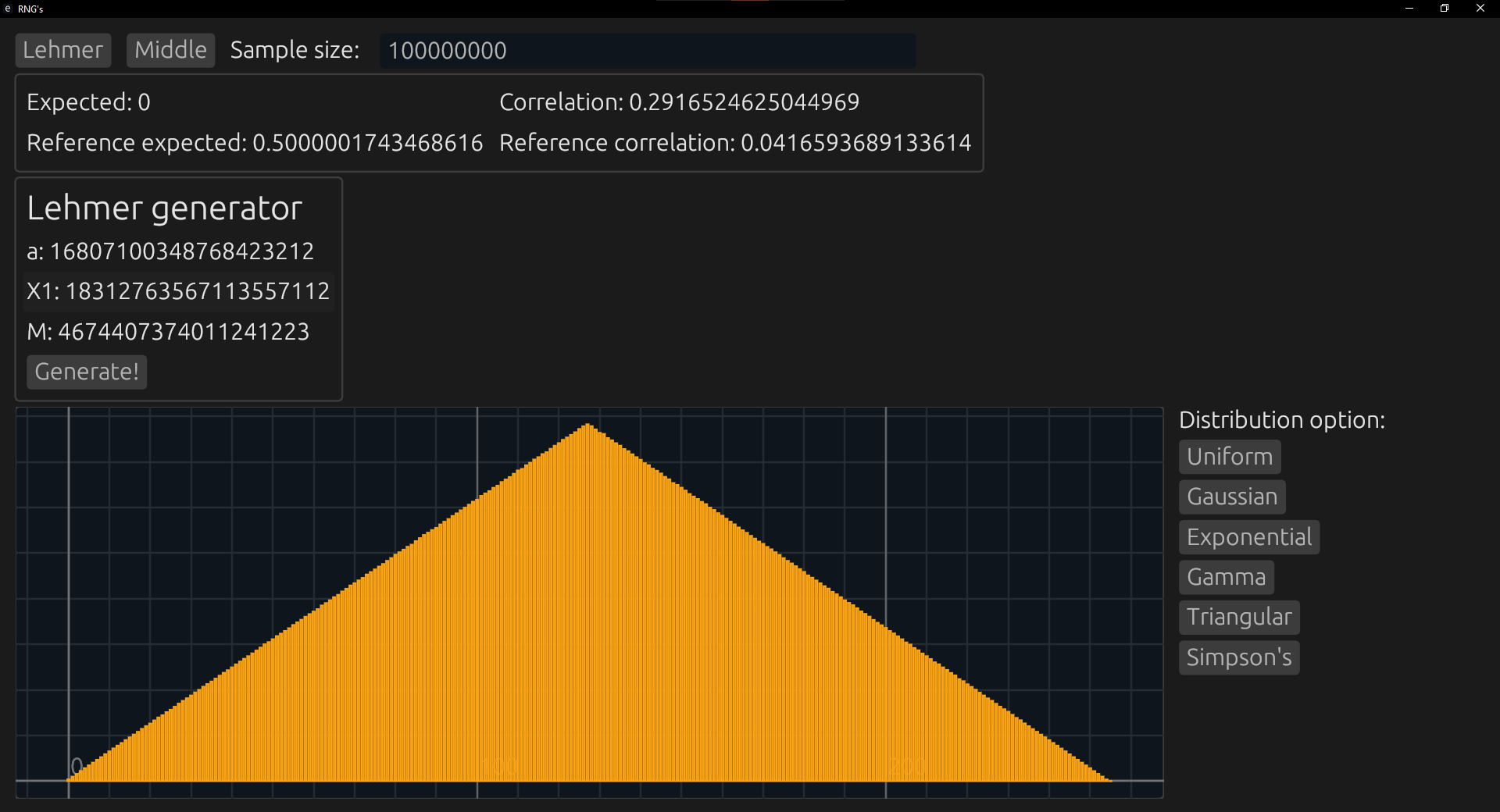
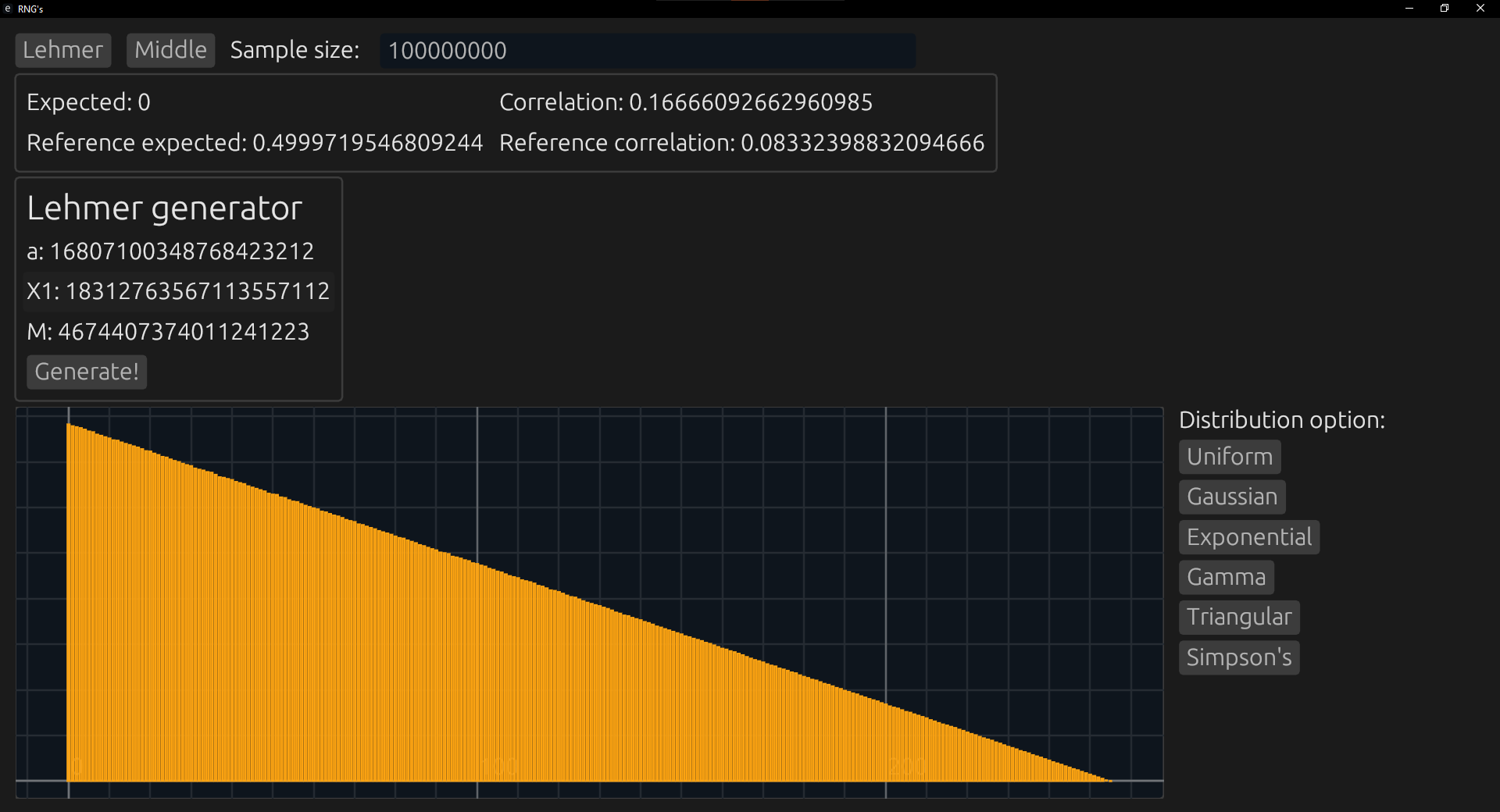
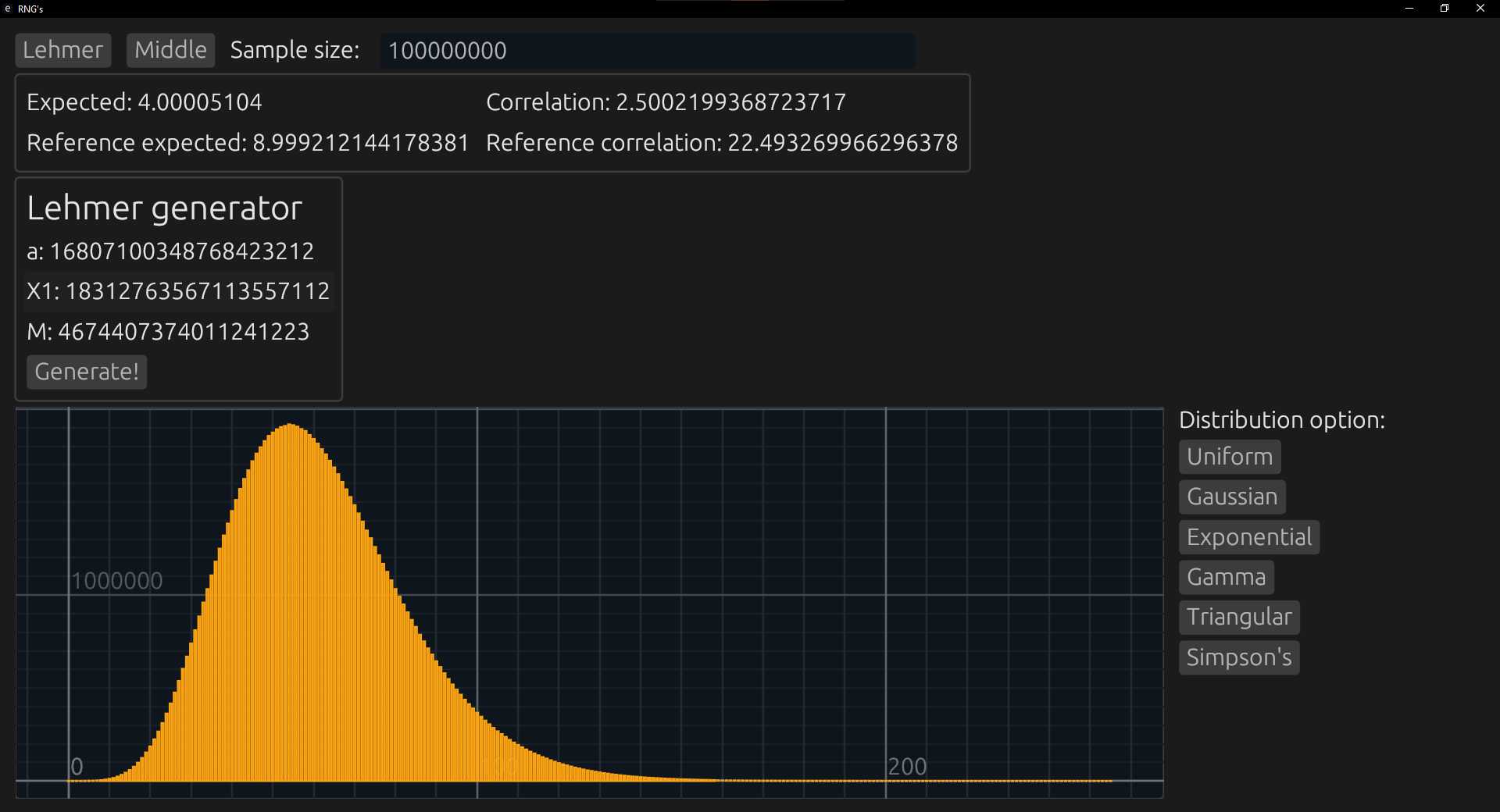
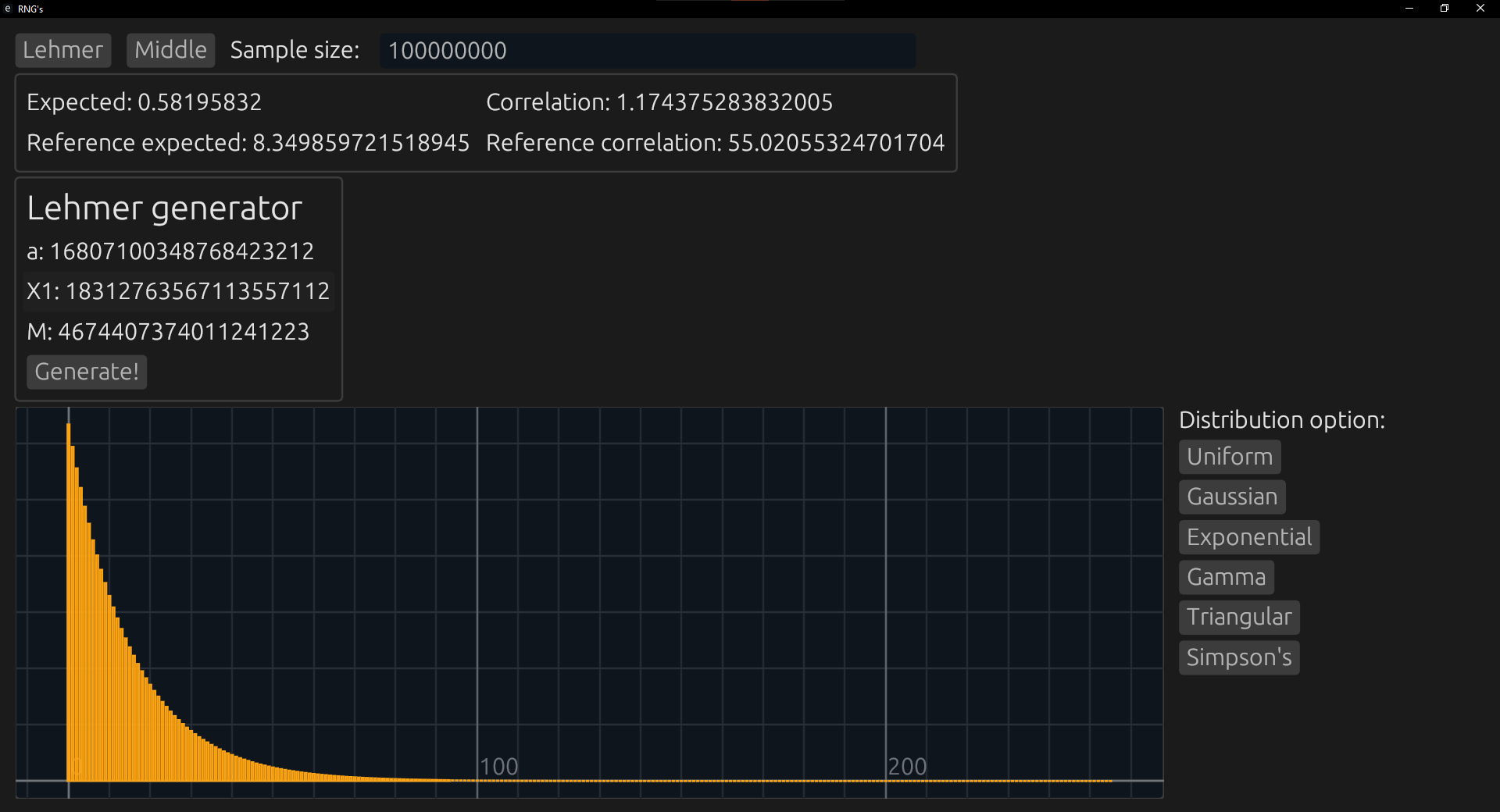
*X = y+z* 

Генератор с выборкой равной n = 500000



Генератор с выборкой равной n = 100000000

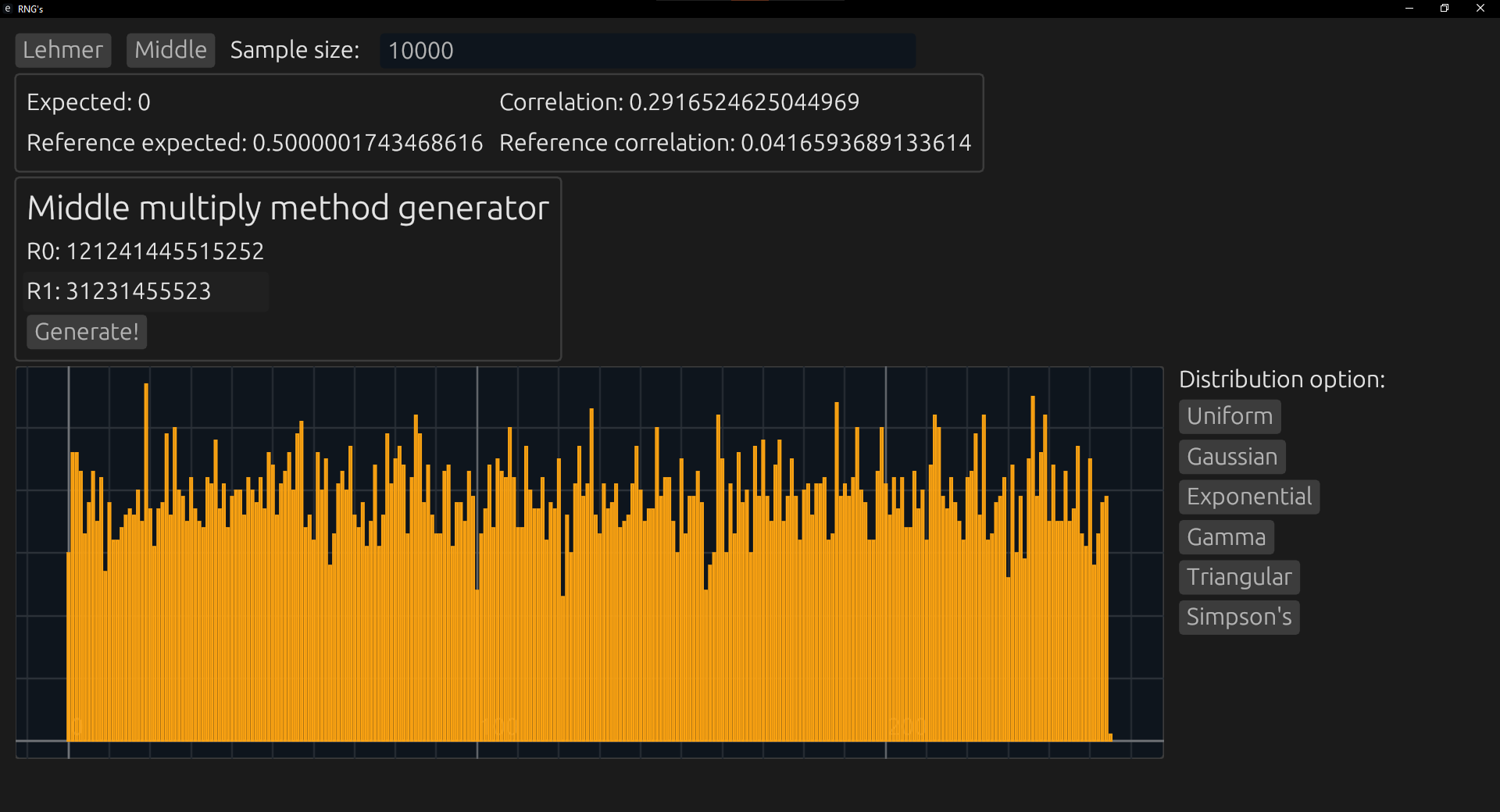


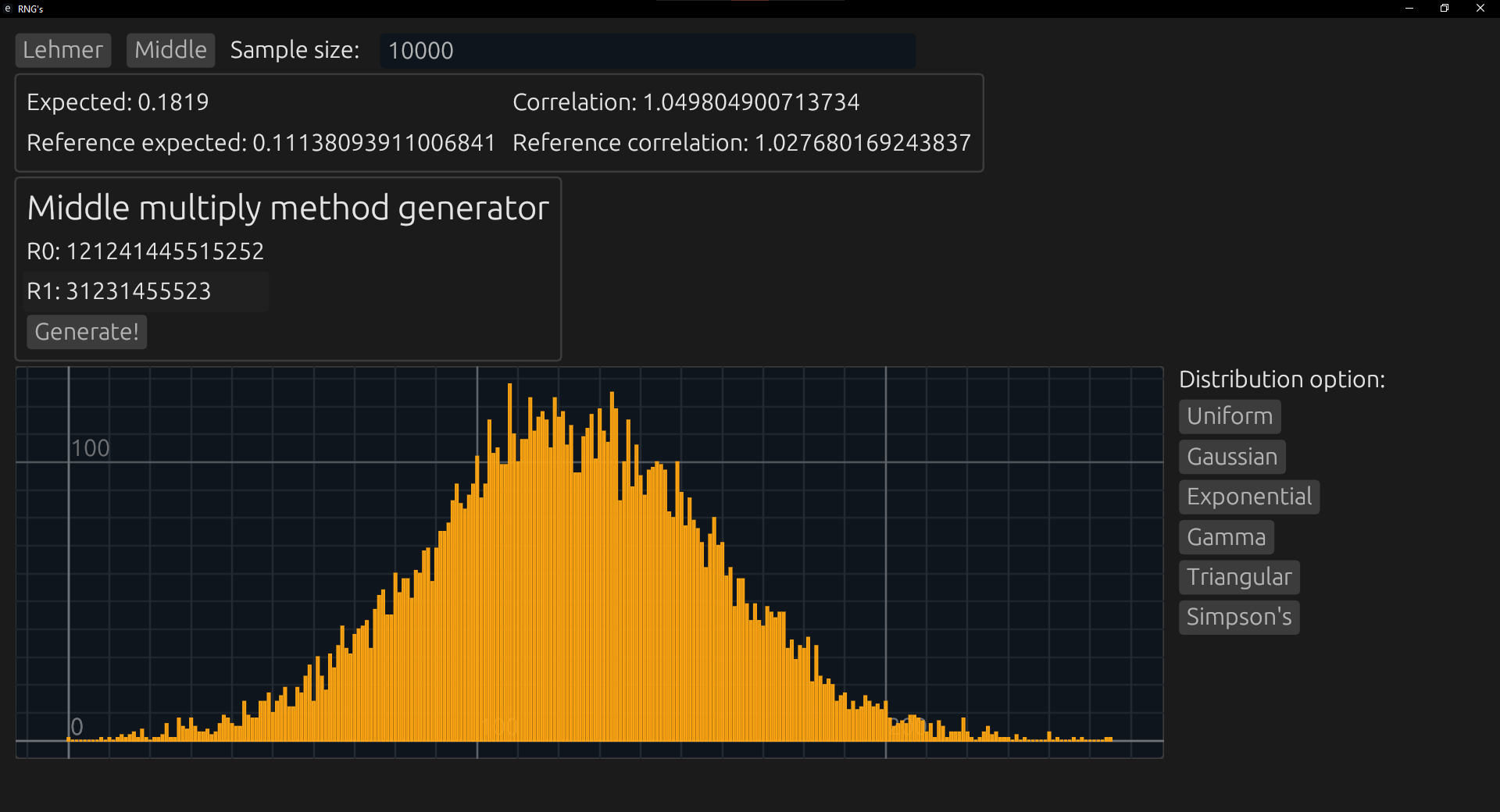


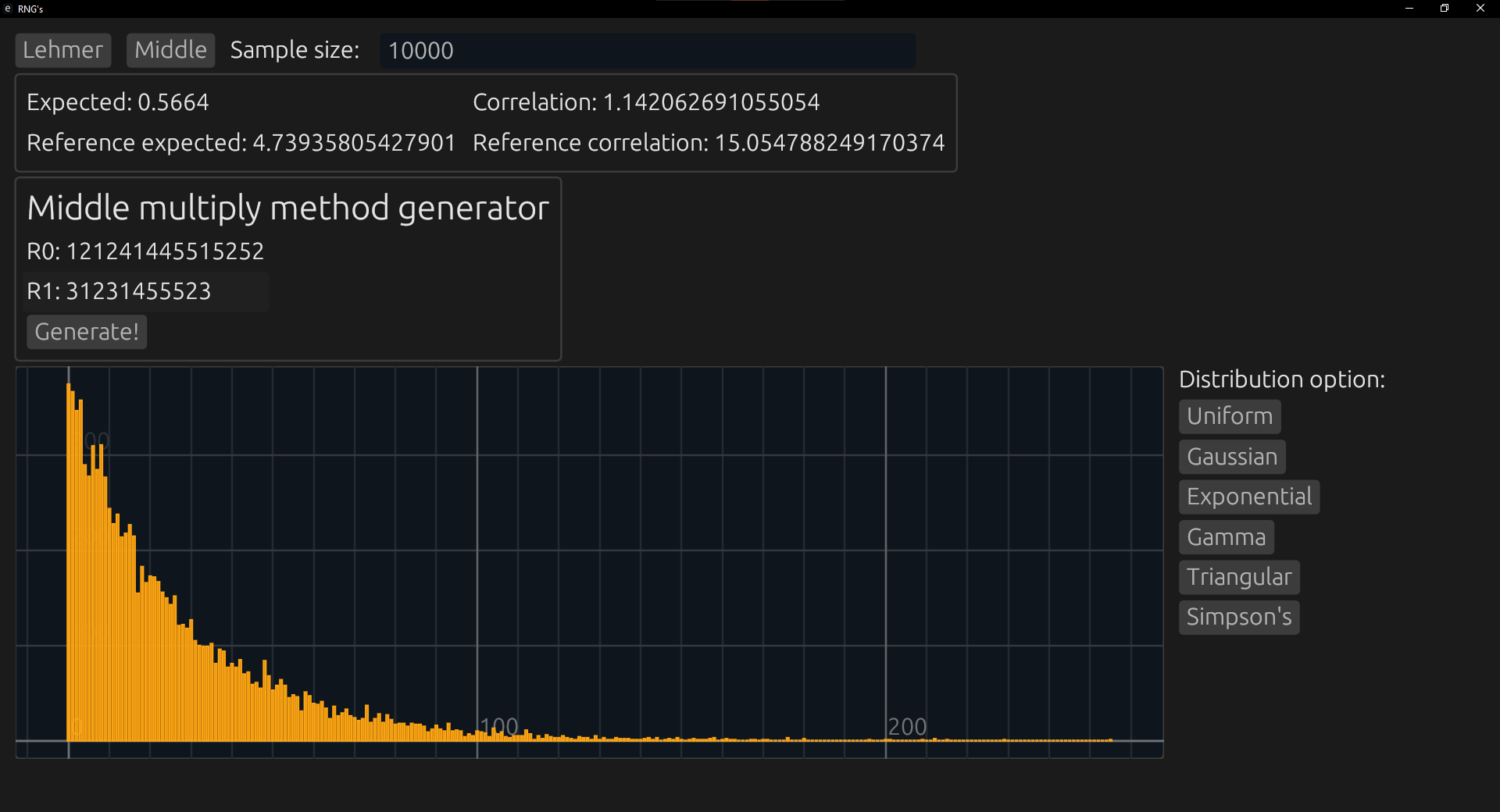
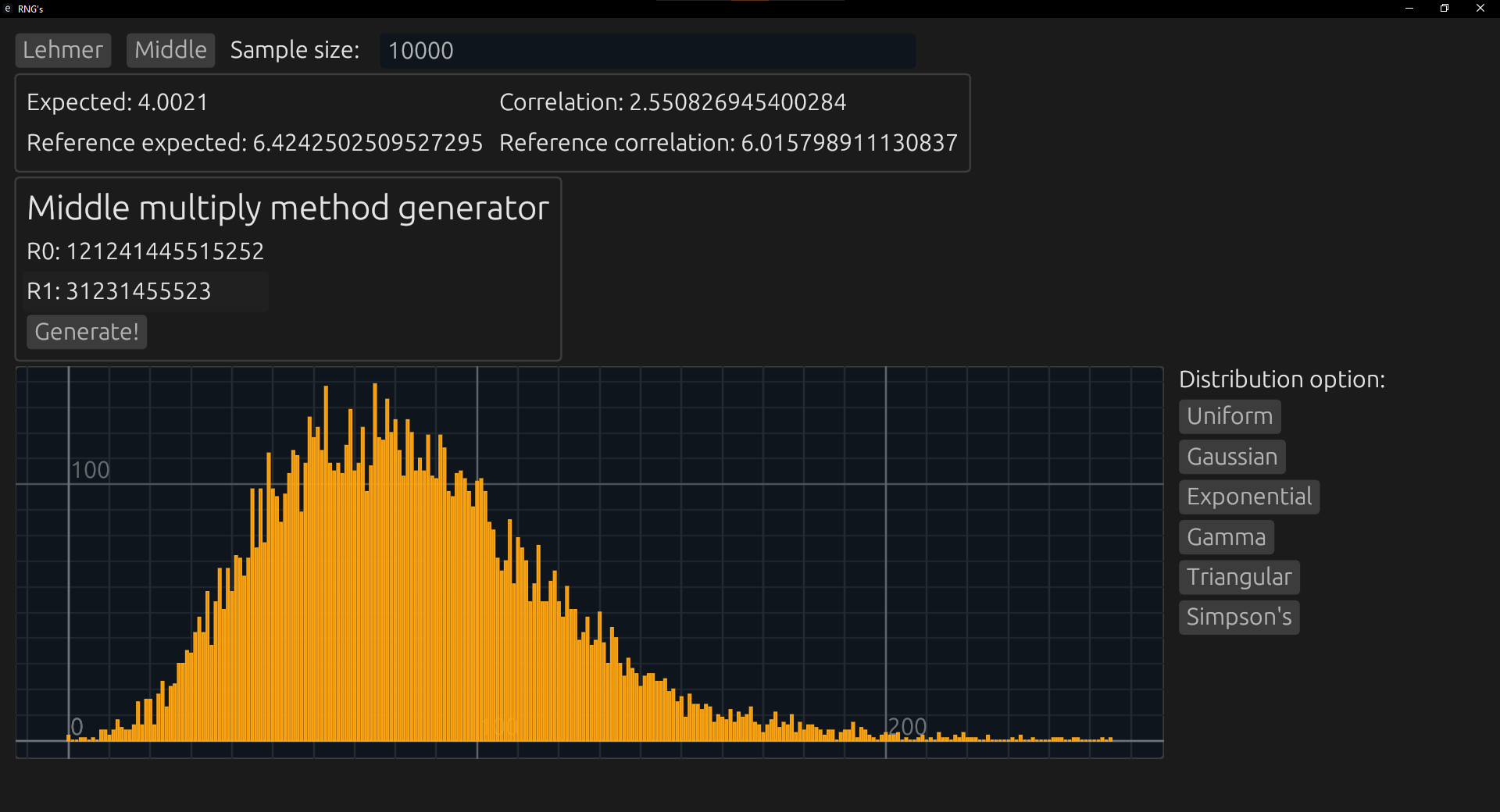
**3.2 Генератор серединных произведений**

В качестве начальных значений использовались следующие числа:

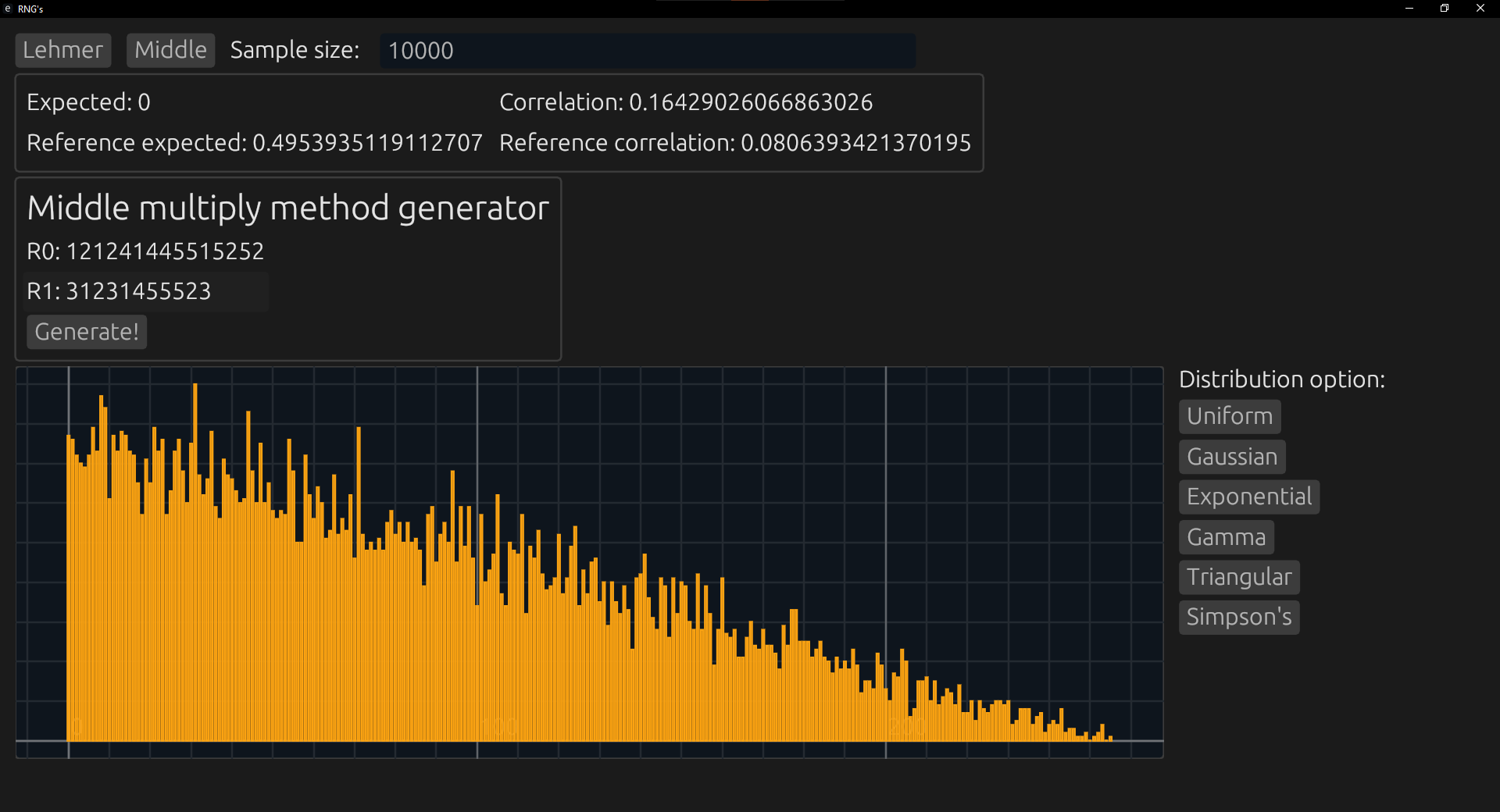
R0: 1.212.414.455.152.521.232  
R1: 3.123.145.552.312.332.141

Генератор с выборкой равной n = 10000

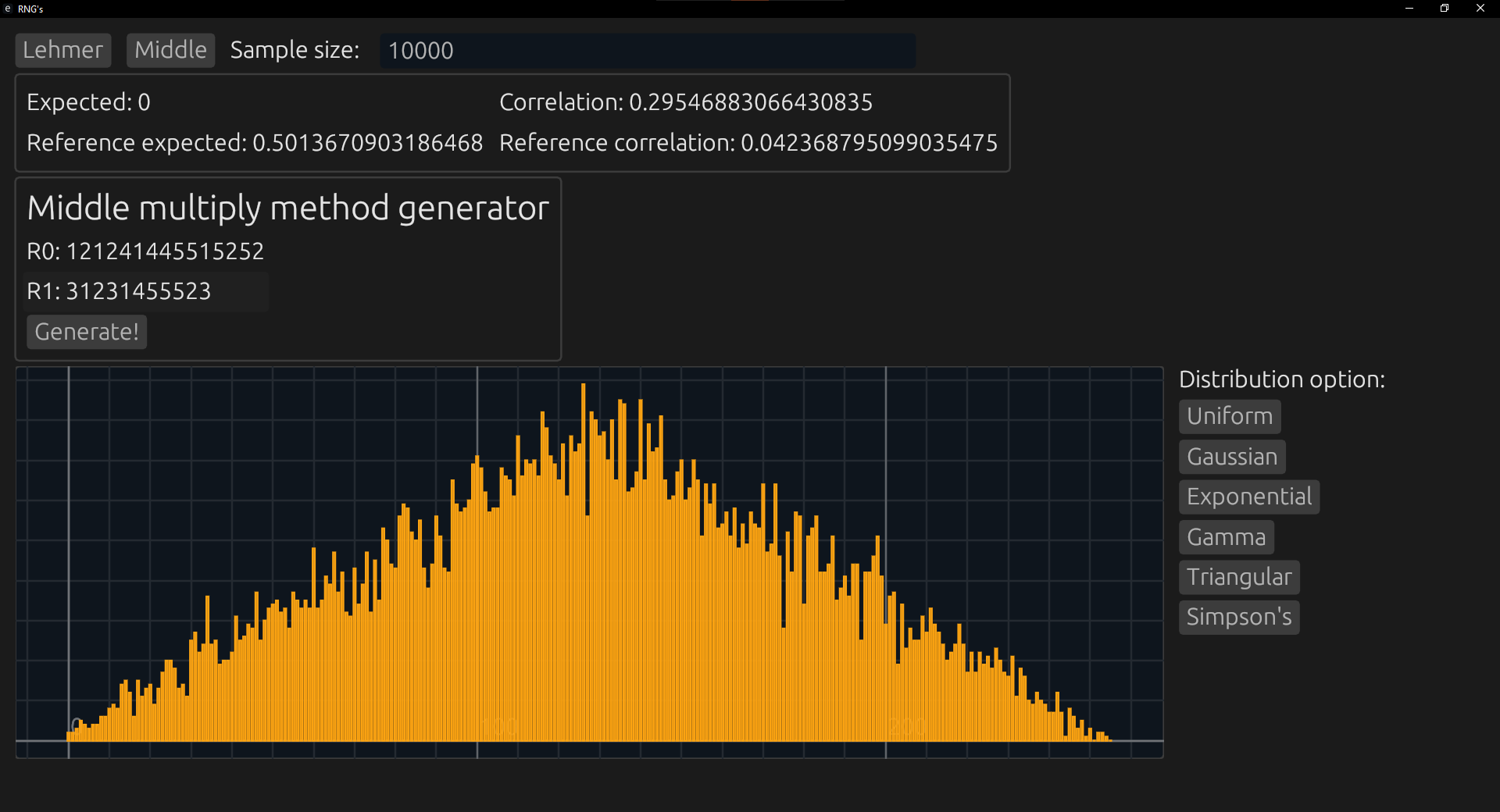


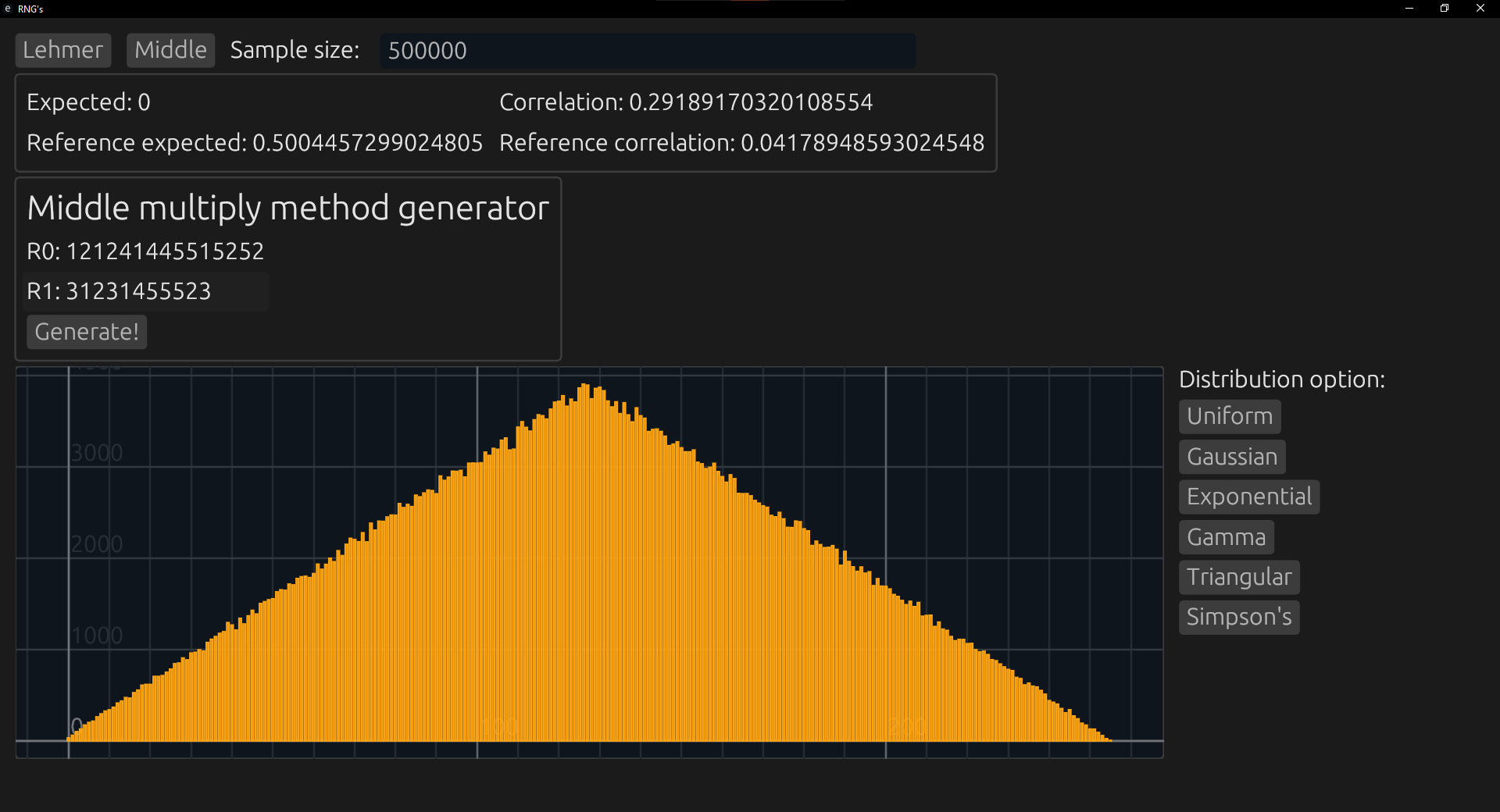
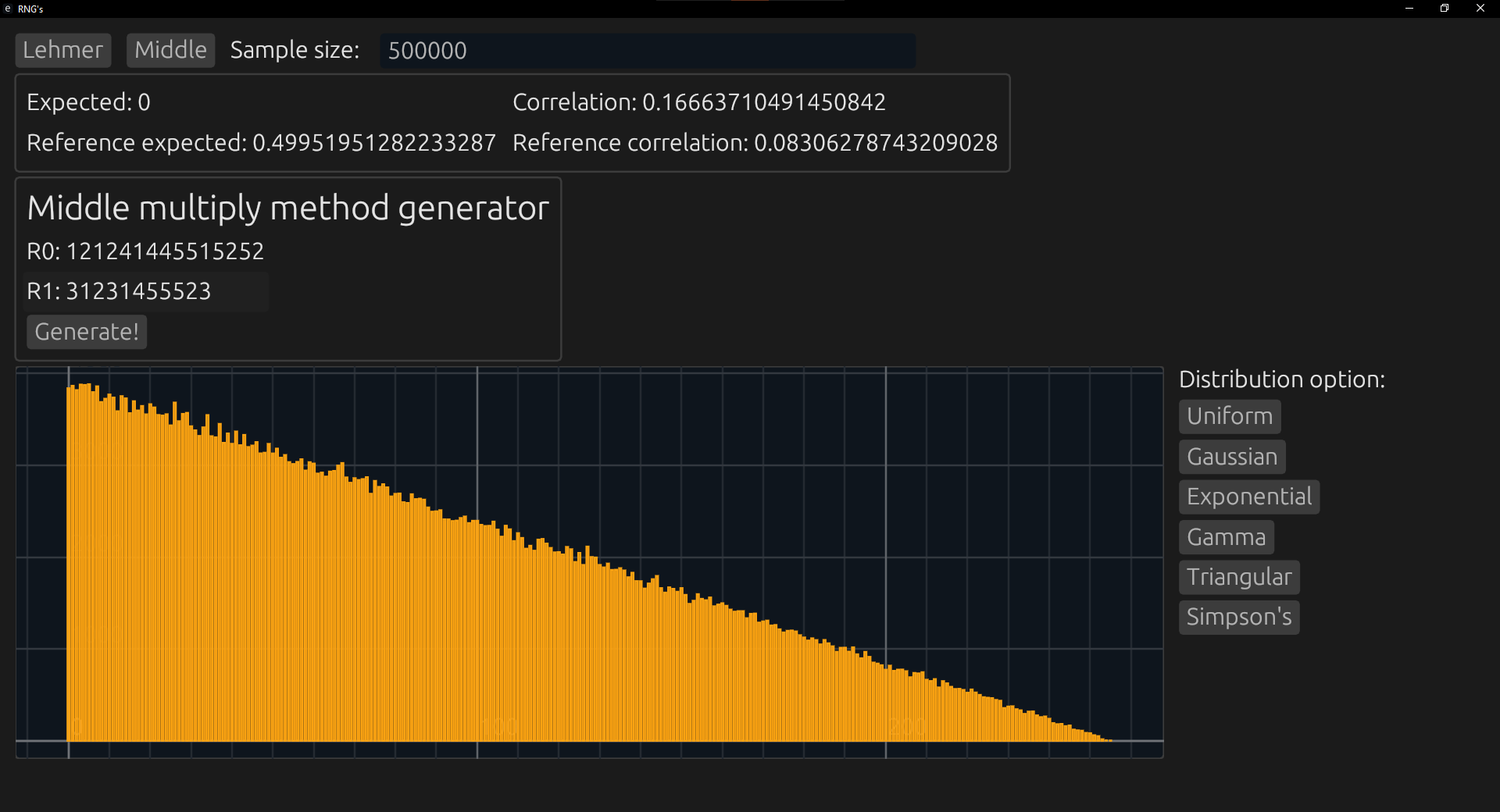
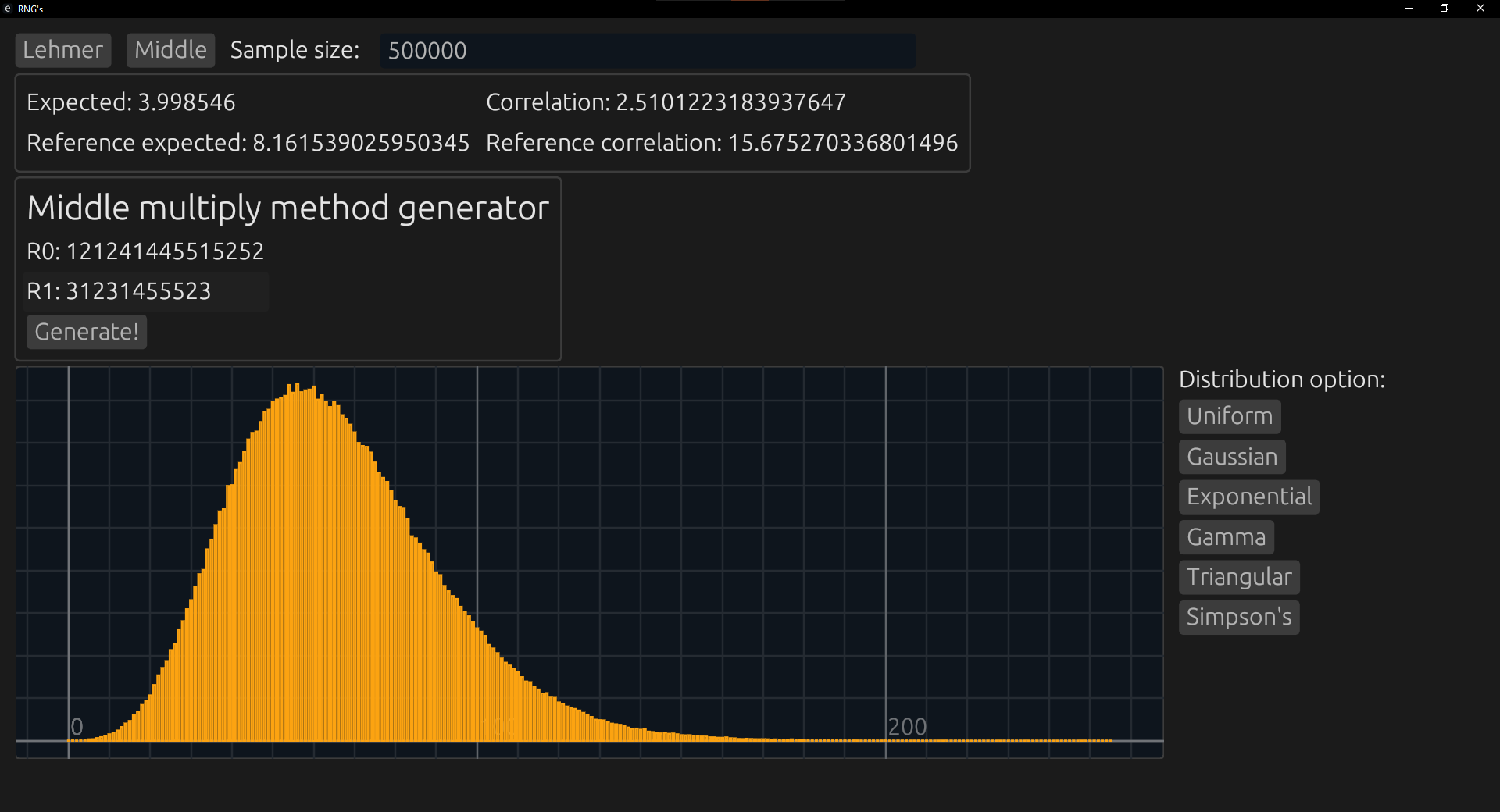
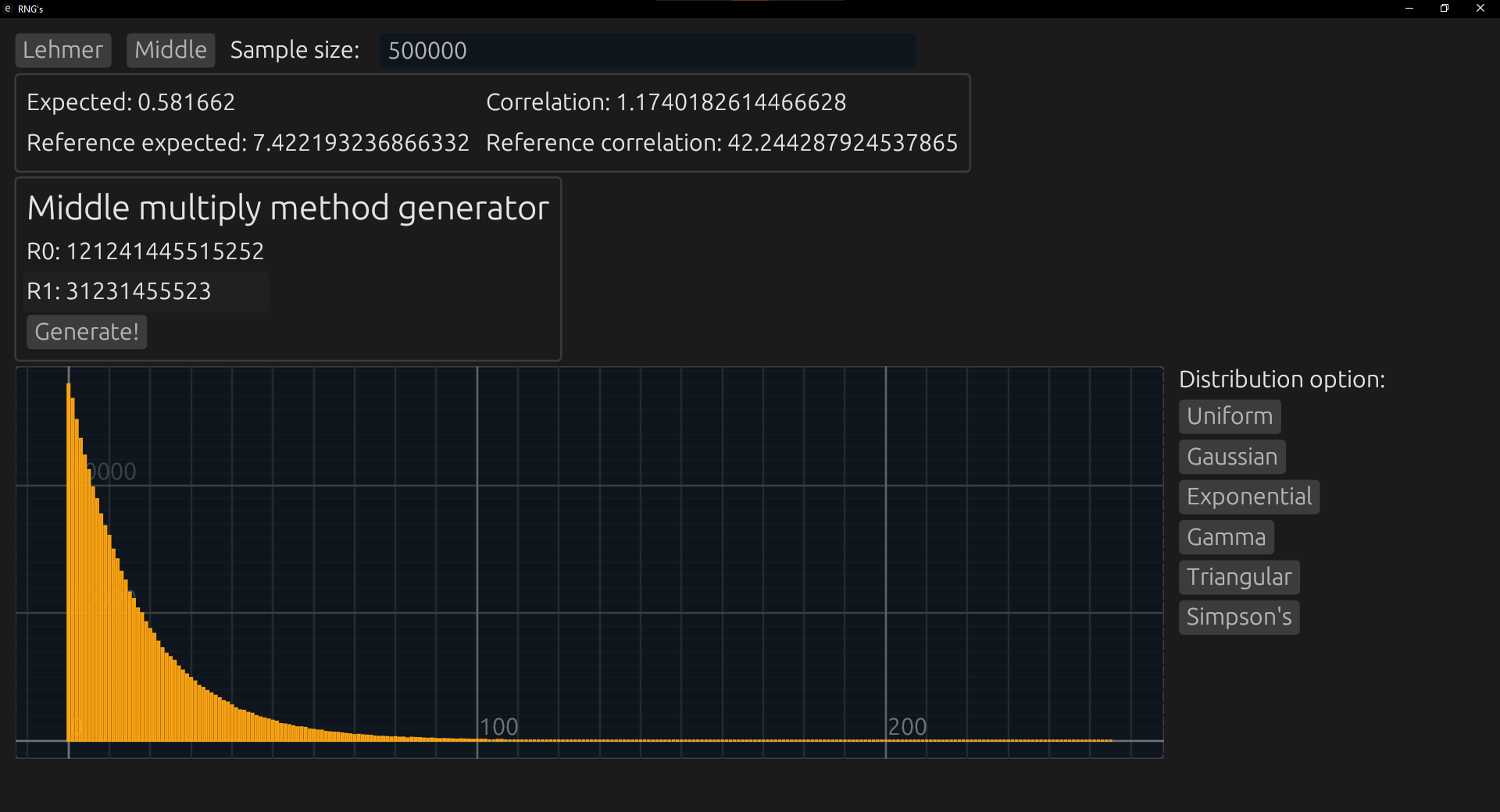
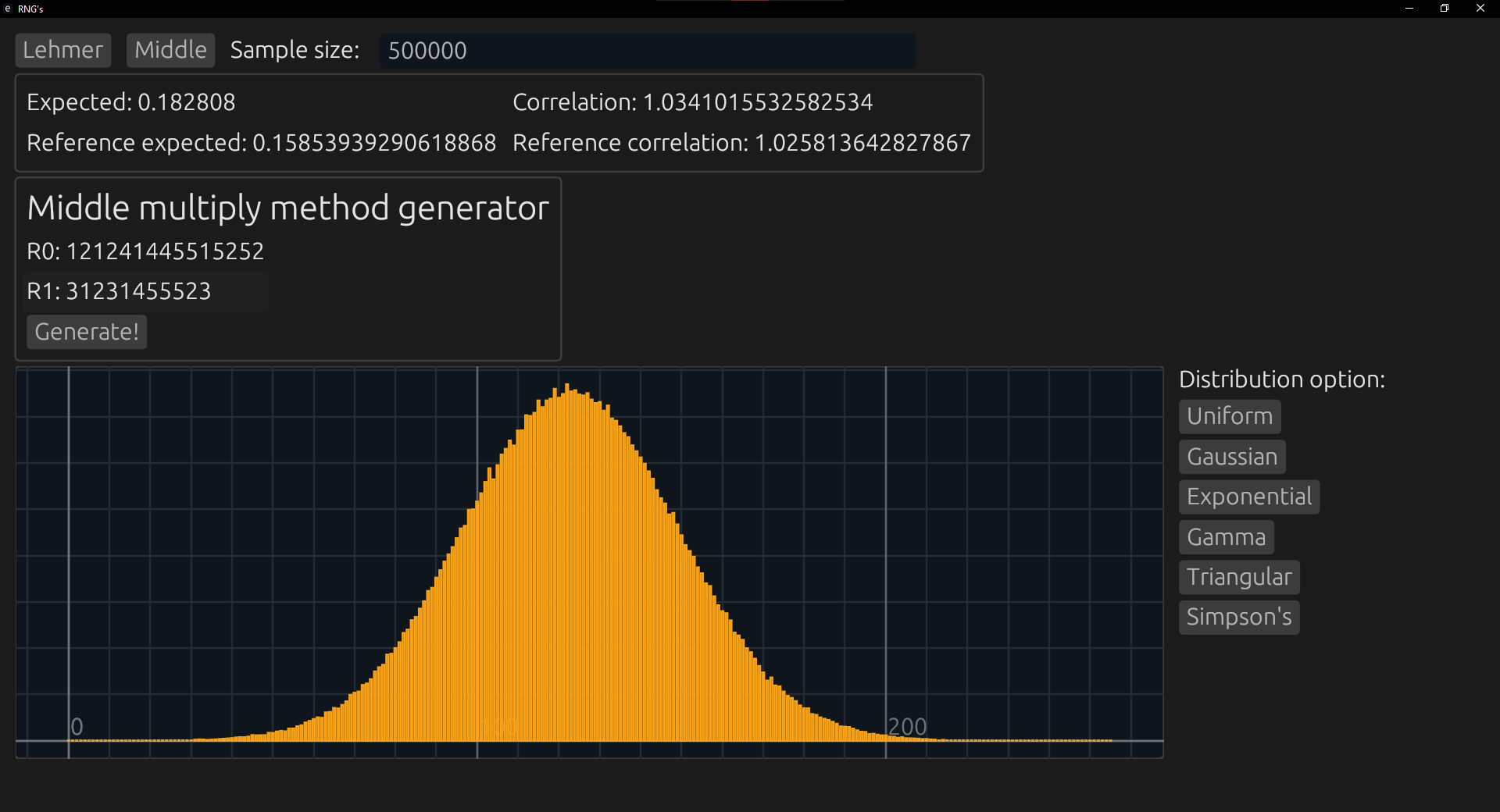
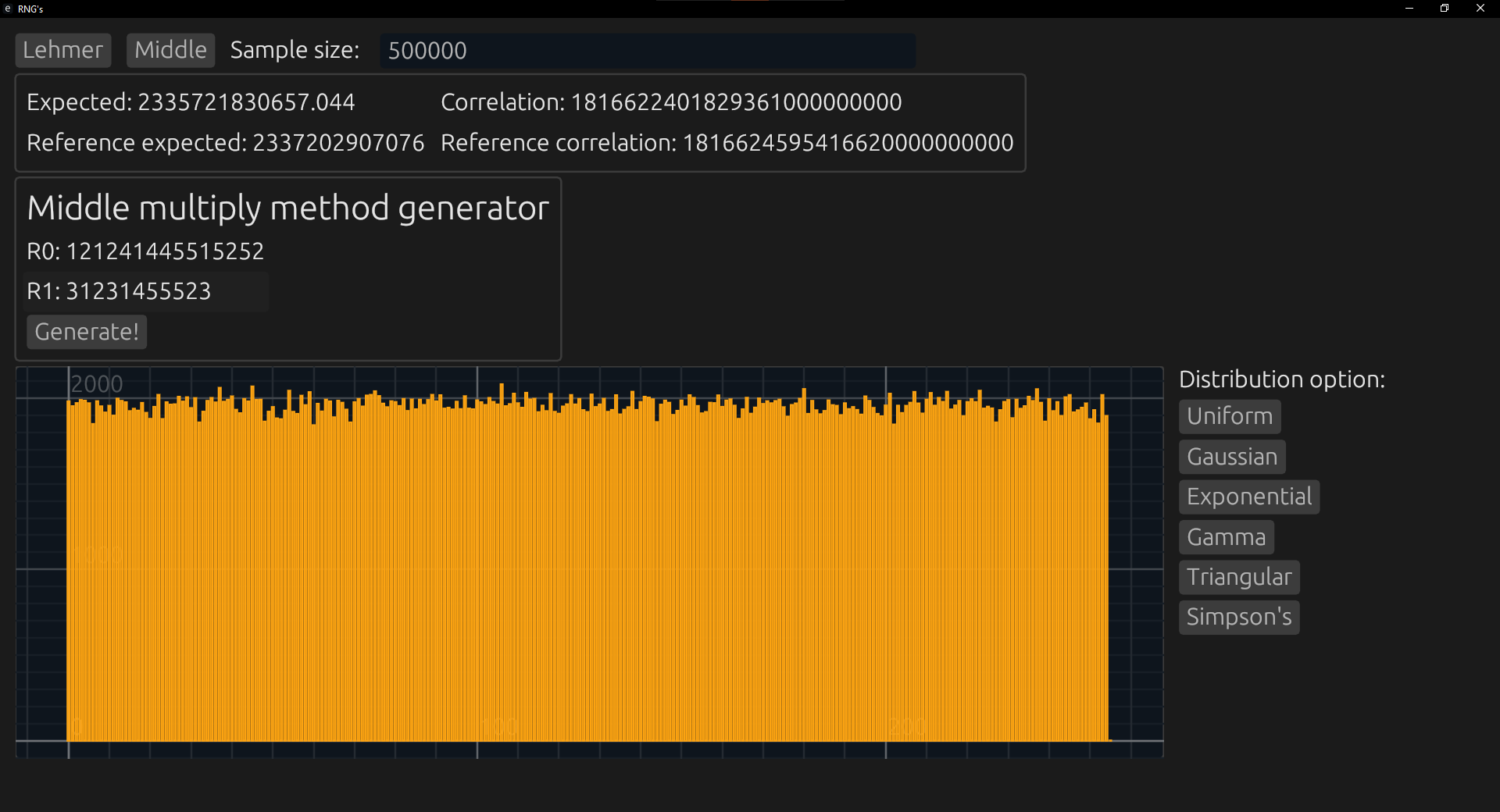
a =19, b = 449325284319556245



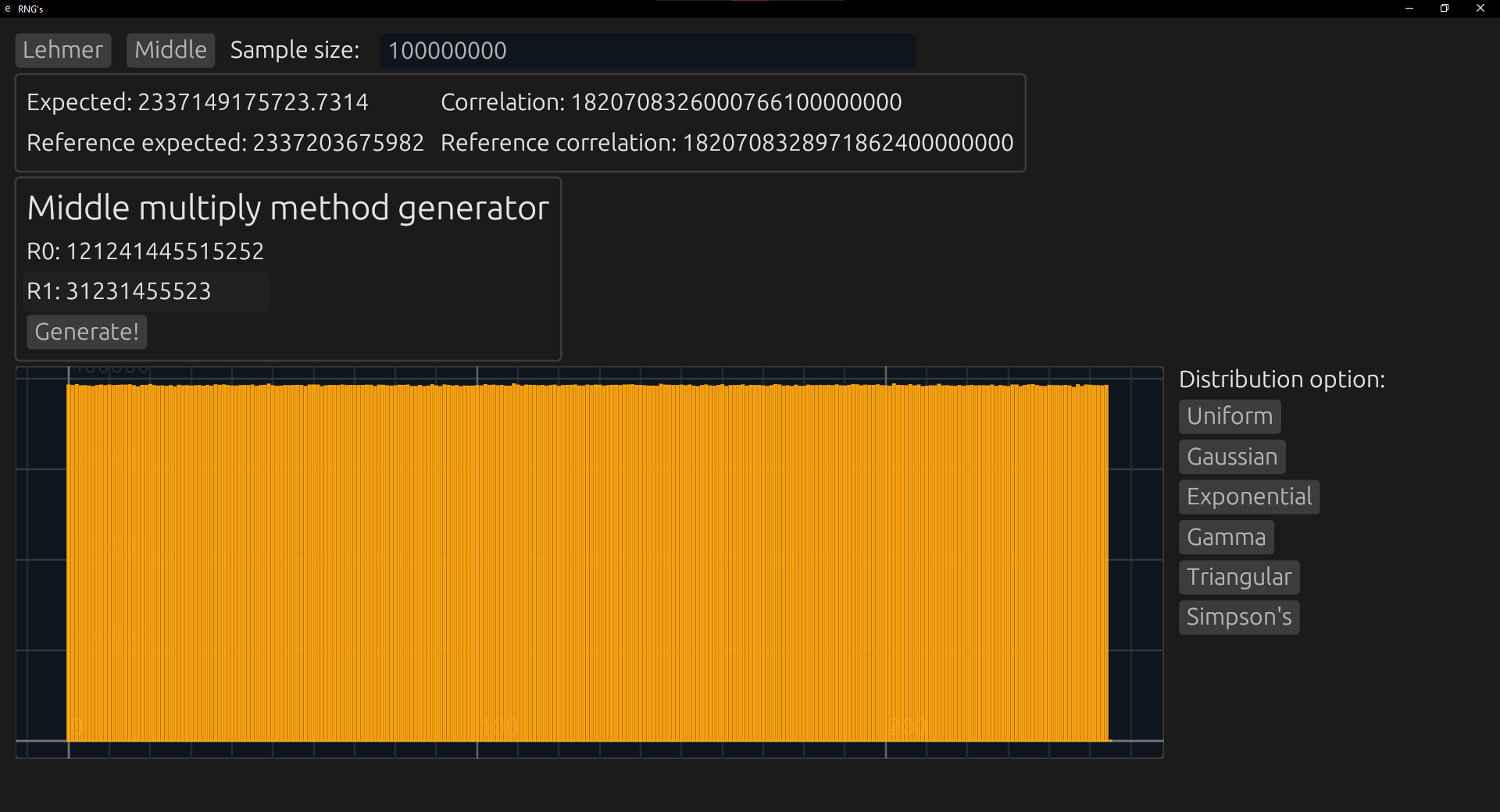
*X = y+z*

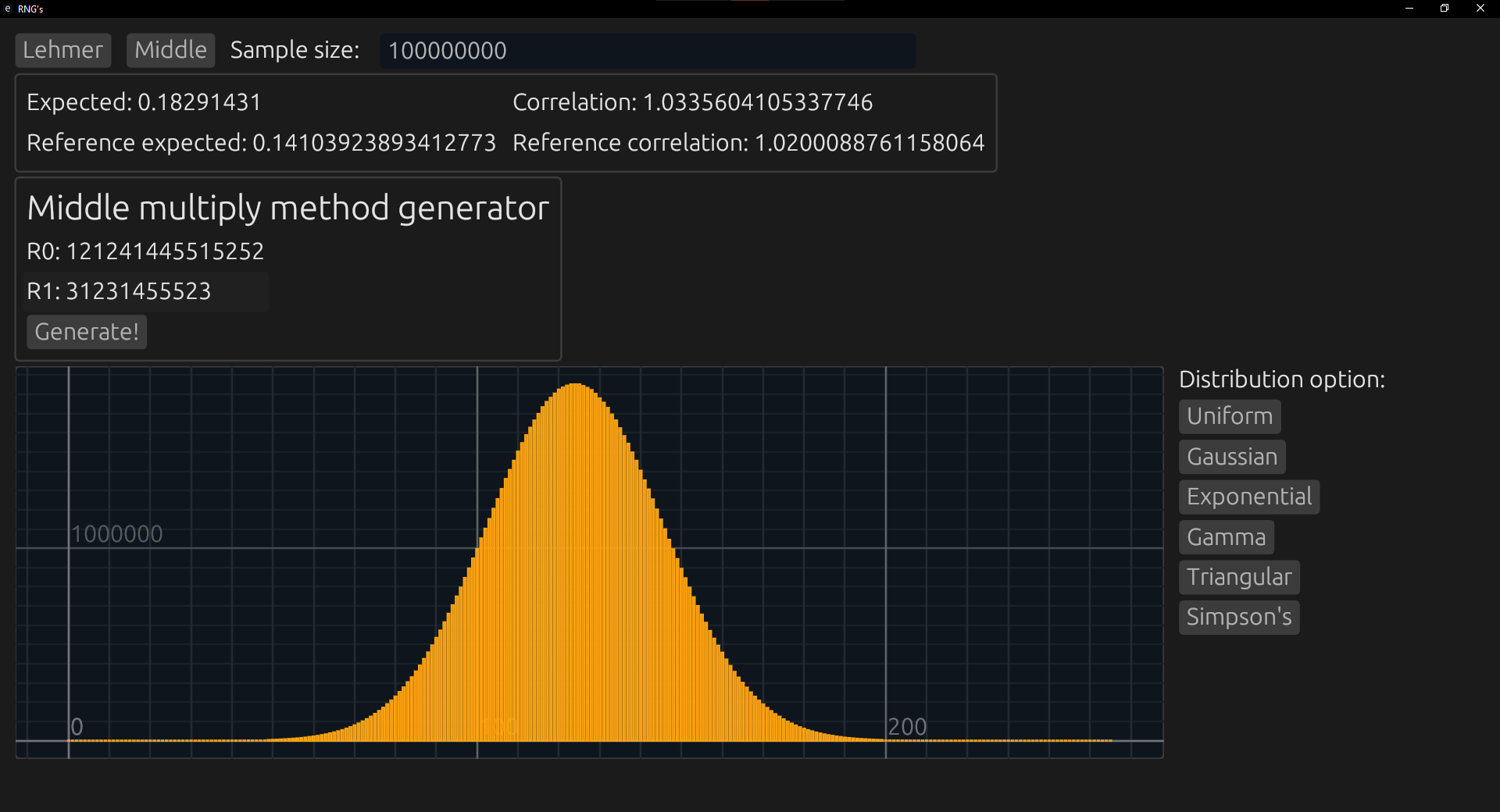


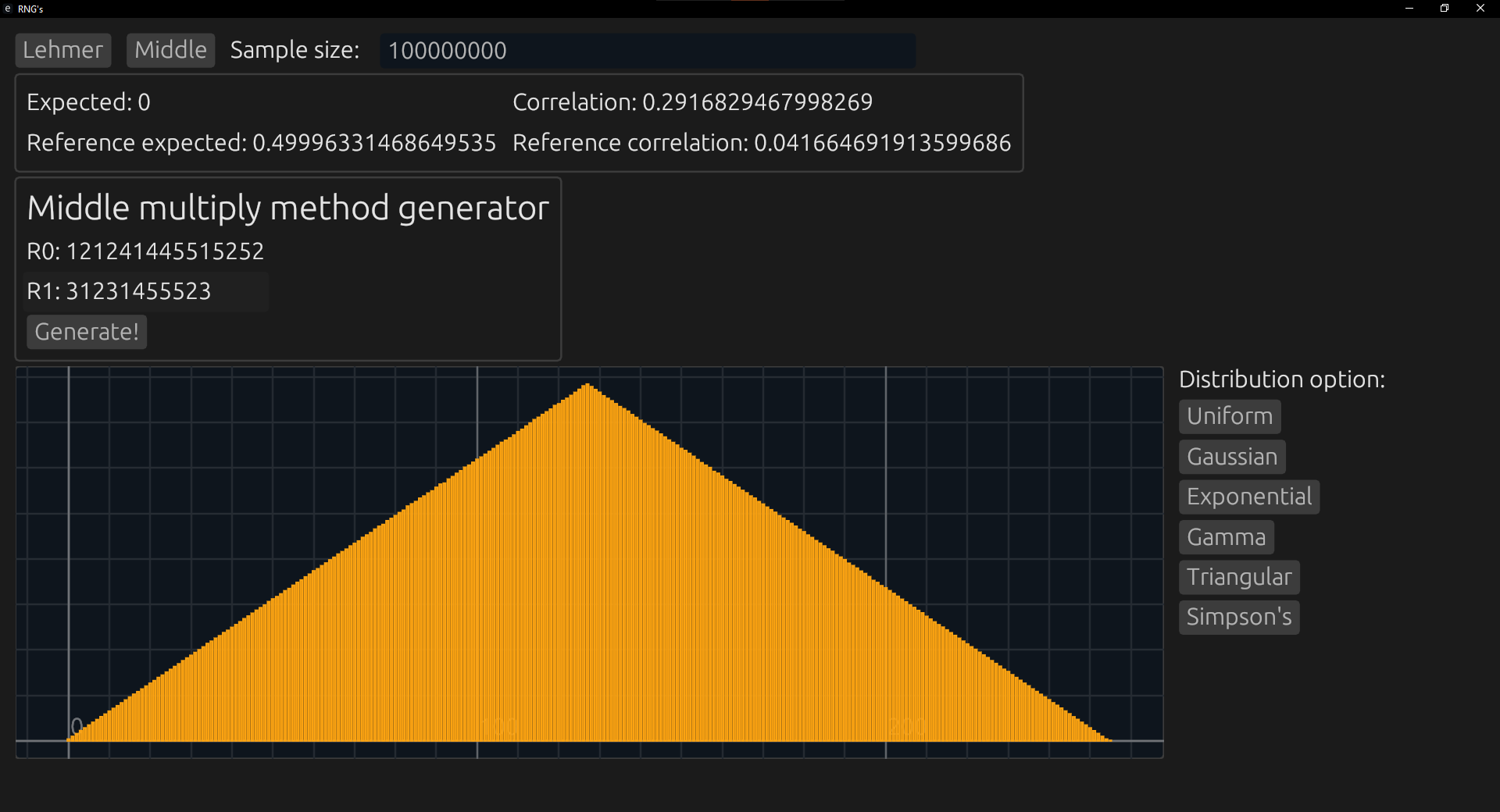
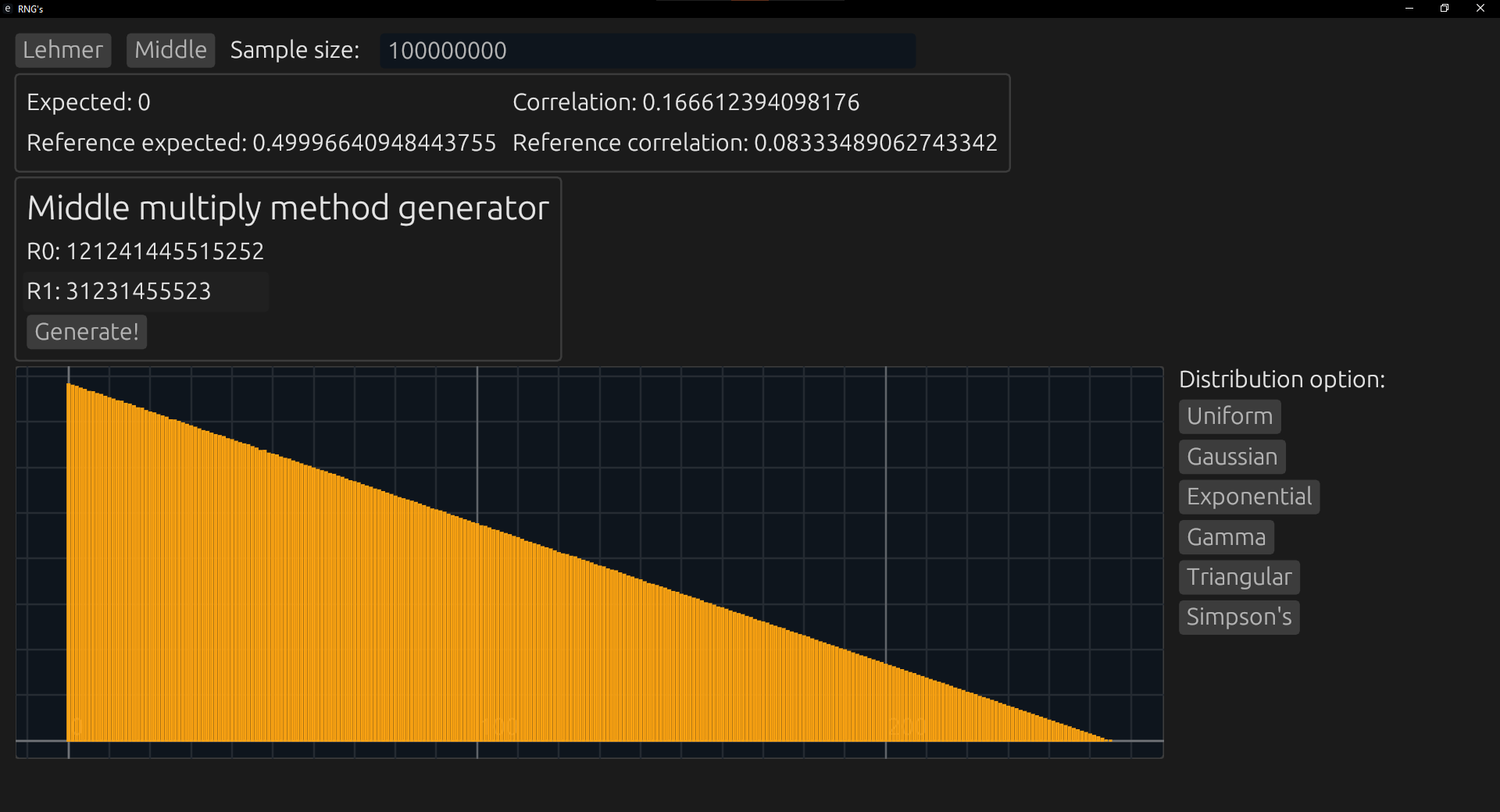
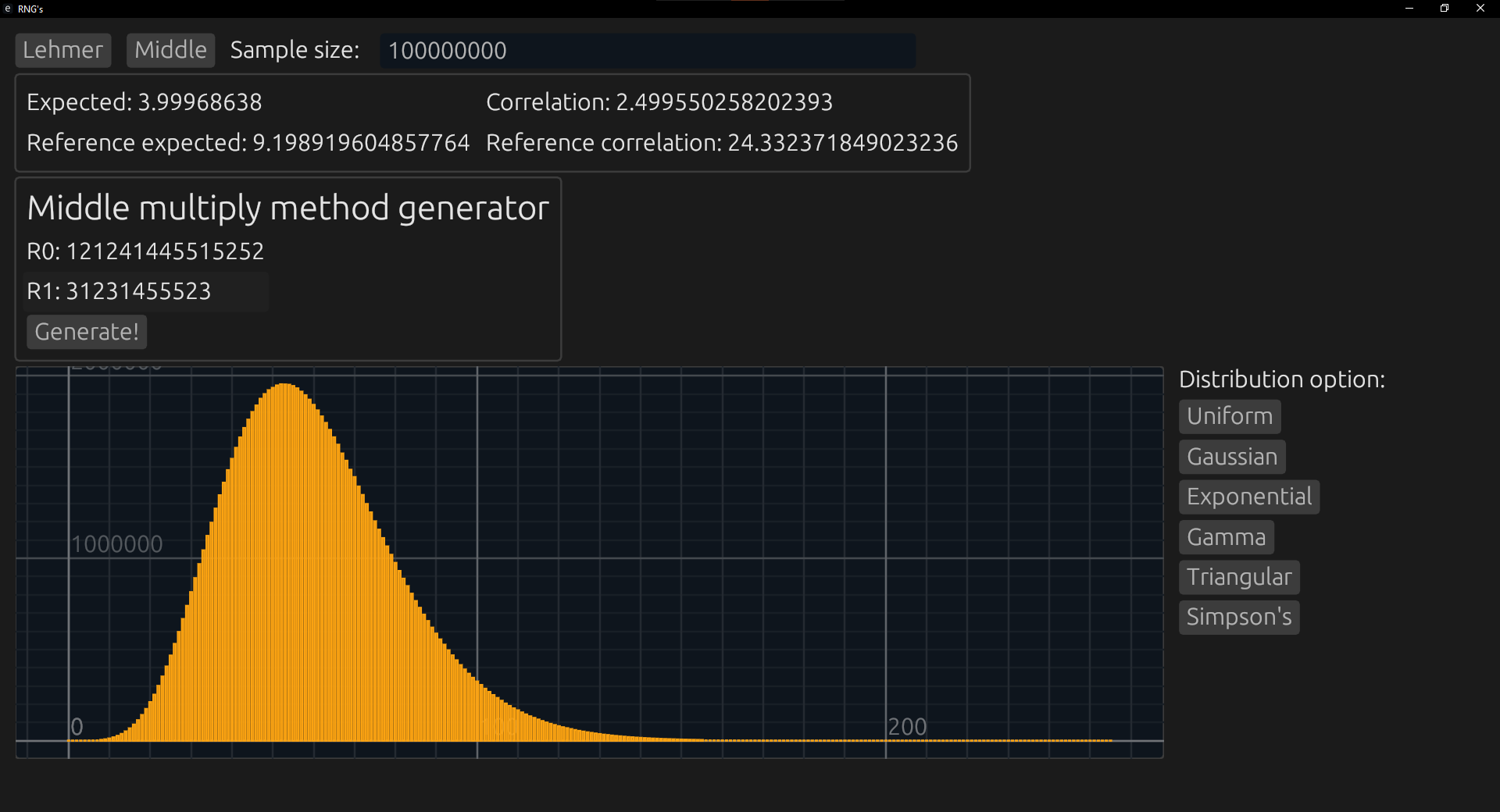
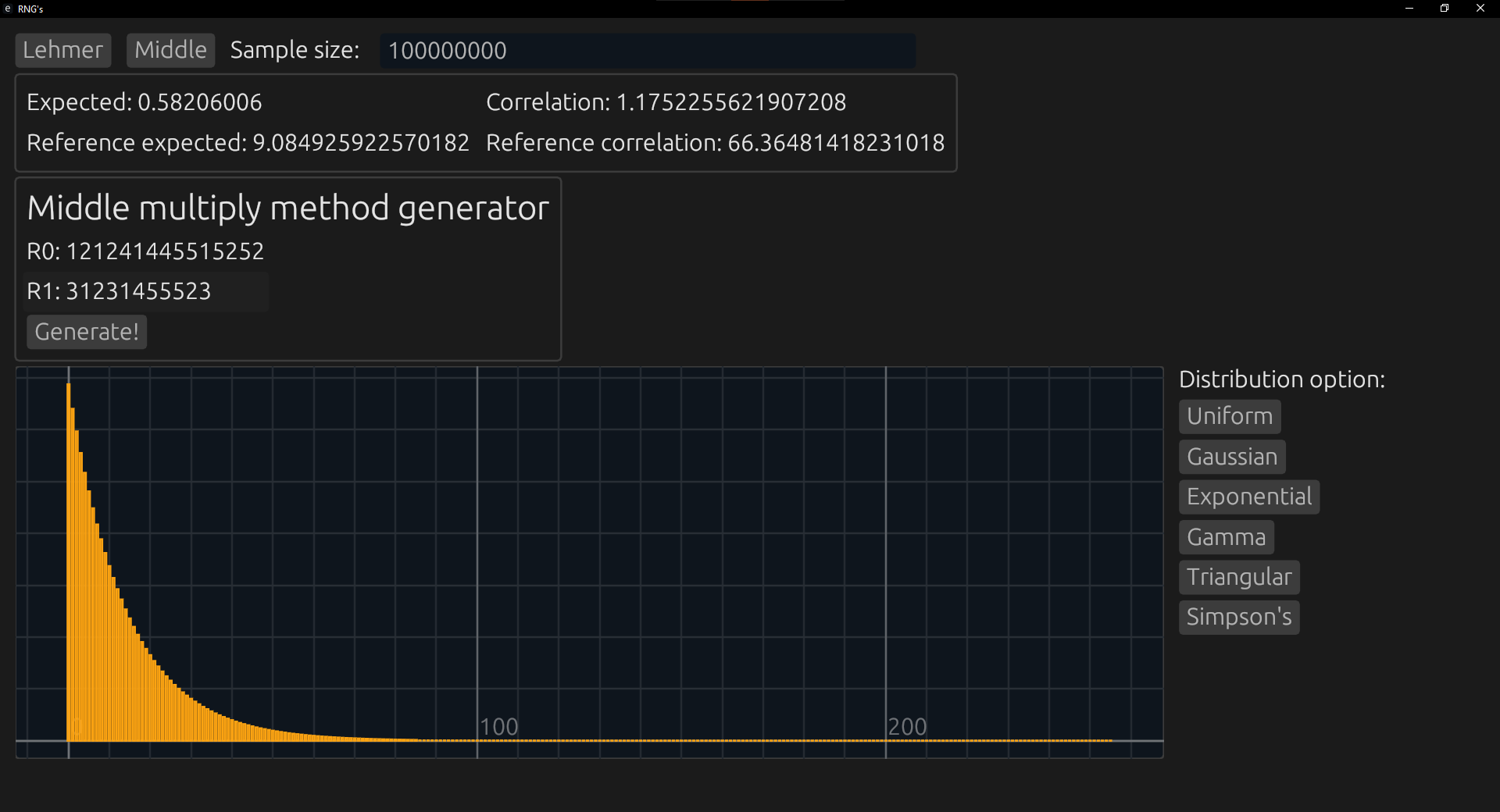
Генератор с выборкой равной n = 500000



Генератор с выборкой равной n = 100000000

****

****

****

**4 Листинг кода**

Данная лабораторная работа была выполнена на языке Rust

Генератор Лемера

fn generate\_lehmer(&mut self) {

let a = self.a\_buffer as u128;

let x1 = self.x1\_buffer as u128;

let m = self.m\_buffer as u128;

self.random\_numbers.push(x1 as u64);

for i in 1..self.count {

let t = a \* self.random\_numbers[i - 1] as u128;

let \_tmp = t % m;

self.random\_numbers.push(\_tmp as u64);

}

}

Генератор серединных произведений

fn generate\_middle(&mut self) {

let mut r0 = self.r0\_buffer as u128;

let mut r1 = self.r1\_buffer as u128;

let mask: u128 = 0xFFFFFFFFFFFFFFFF00000000;

for \_ in 0..self.count {

let mut new\_r = r0 \* r1;

new\_r &= mask;

new\_r >>= 32;

self.random\_numbers.push(new\_r as u64);

r0 = r1;

r1 = new\_r;

}

}

Генератор, основанный на методе регистров обратной связи

fn generate\_shift(&mut self) {

let mut num = self.shift\_buffer;

for \_ in 0..self.count {

let b\_0 = num & 1;

let b\_1 = (num & 2) >> 1;

let n\_bit = (b\_0 ^ b\_1) << 63;

num >>= 1;

num |= n\_bit;

self.random\_numbers.push(num);

}

}