

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

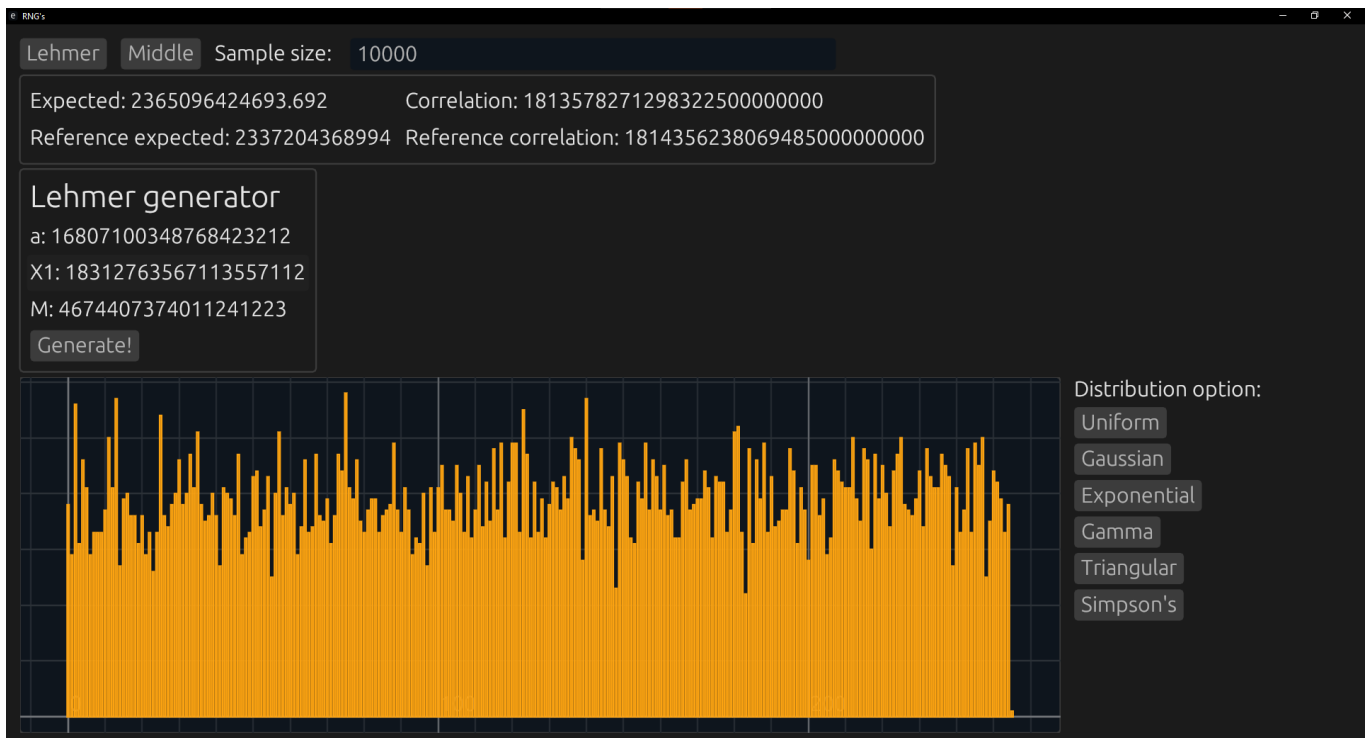
В качестве входных данных на генератор Лемера подаются значения:

A: 11.680.710.034.876.434.312

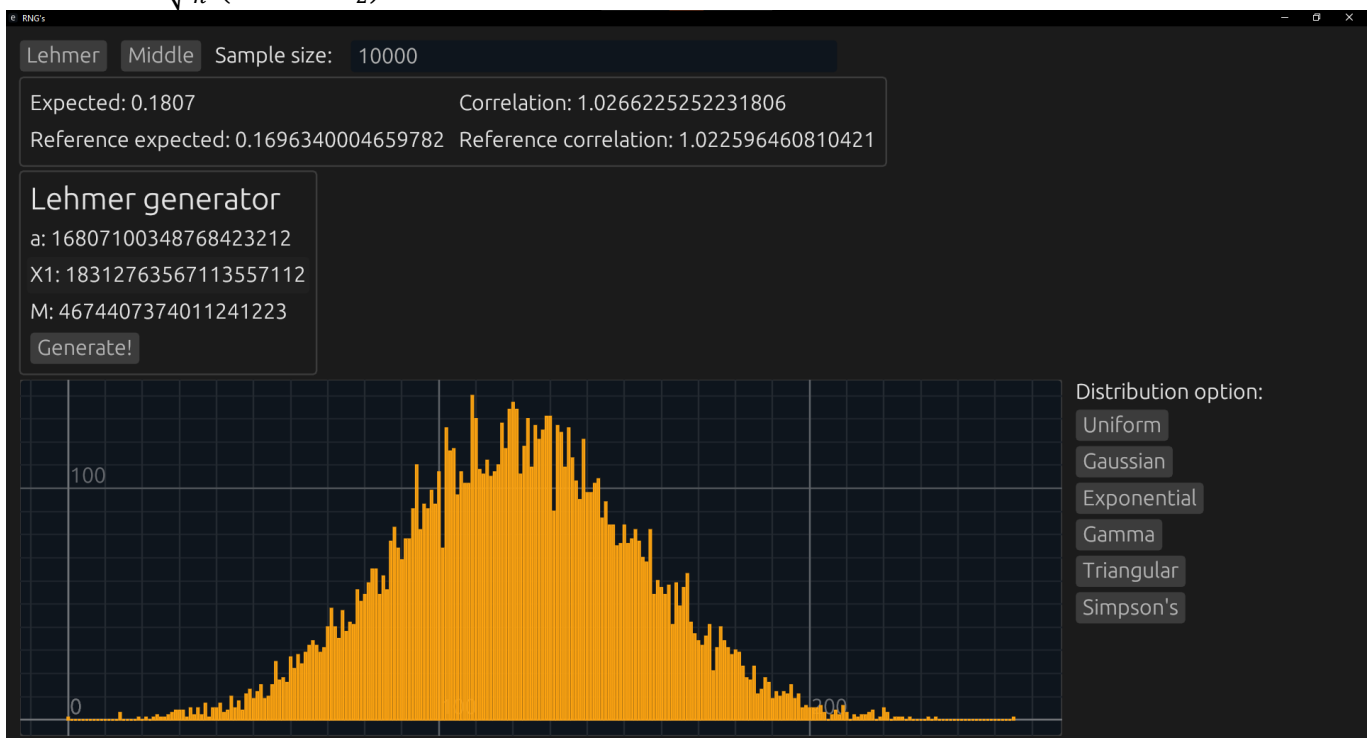
X1:10.312.763.567.113.667.112

m: 4.674.407.374.011.241.223

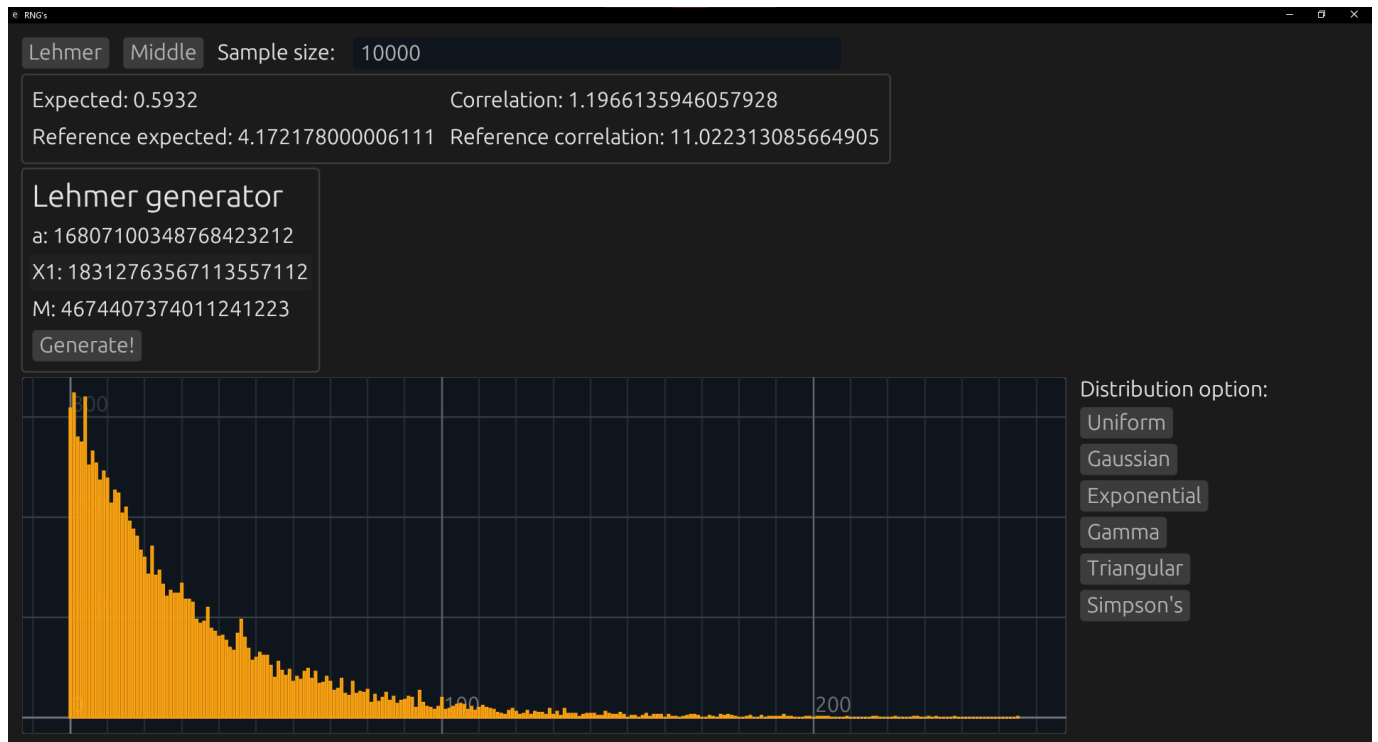
Генератор с выборкой равной $n = 10000$



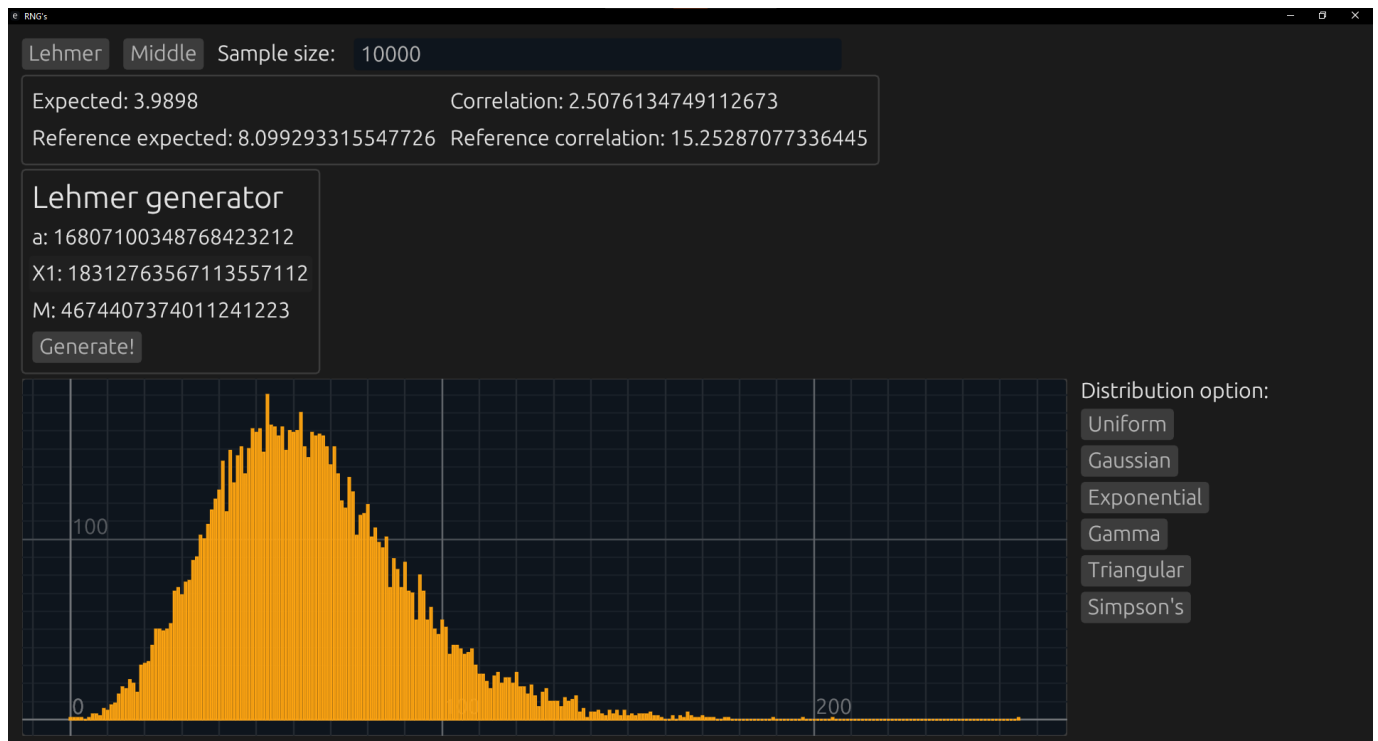
$$x = m_x + \sigma_x \sqrt{\frac{12}{n}} \left(\sum_{i=1}^n R_i - \frac{n}{2} \right), \sigma_x = 21, n = 6$$



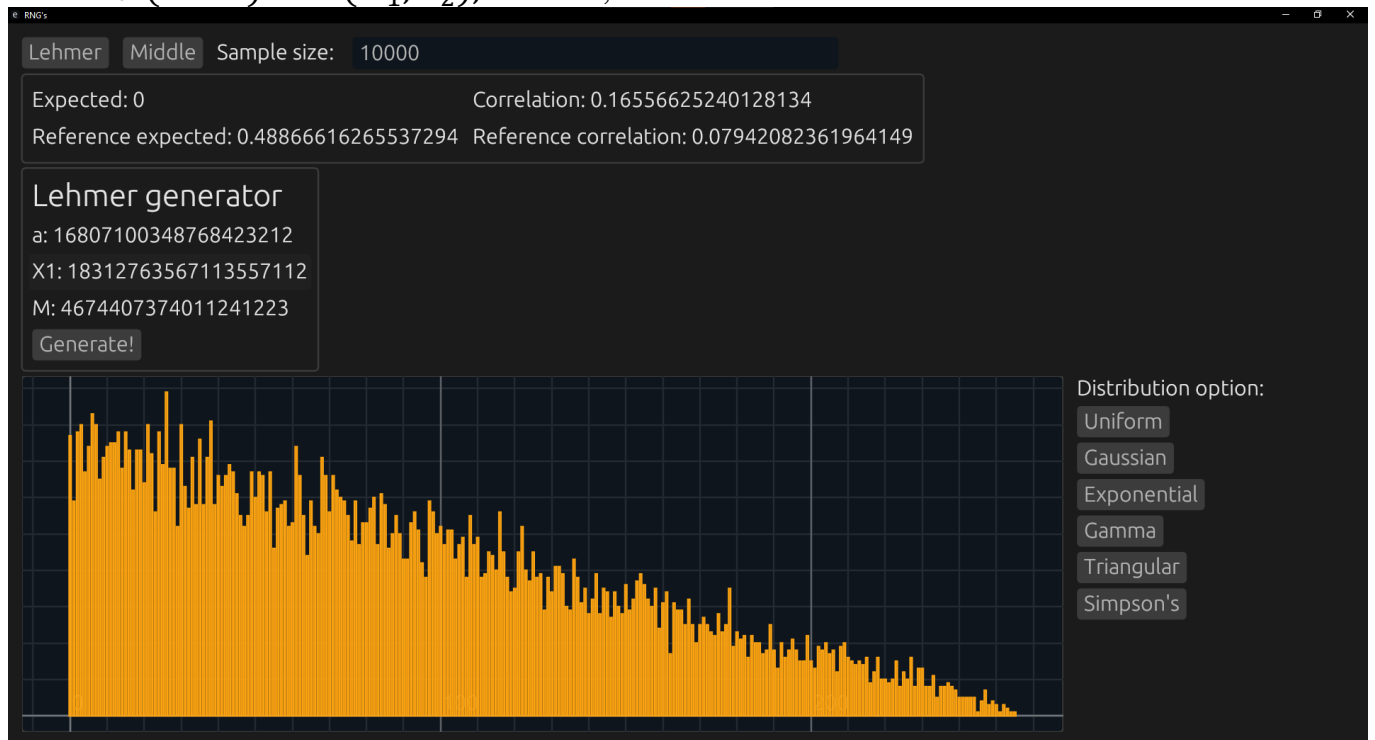
$$X = -\frac{1}{\lambda} \ln R. \lambda = 2,71828$$



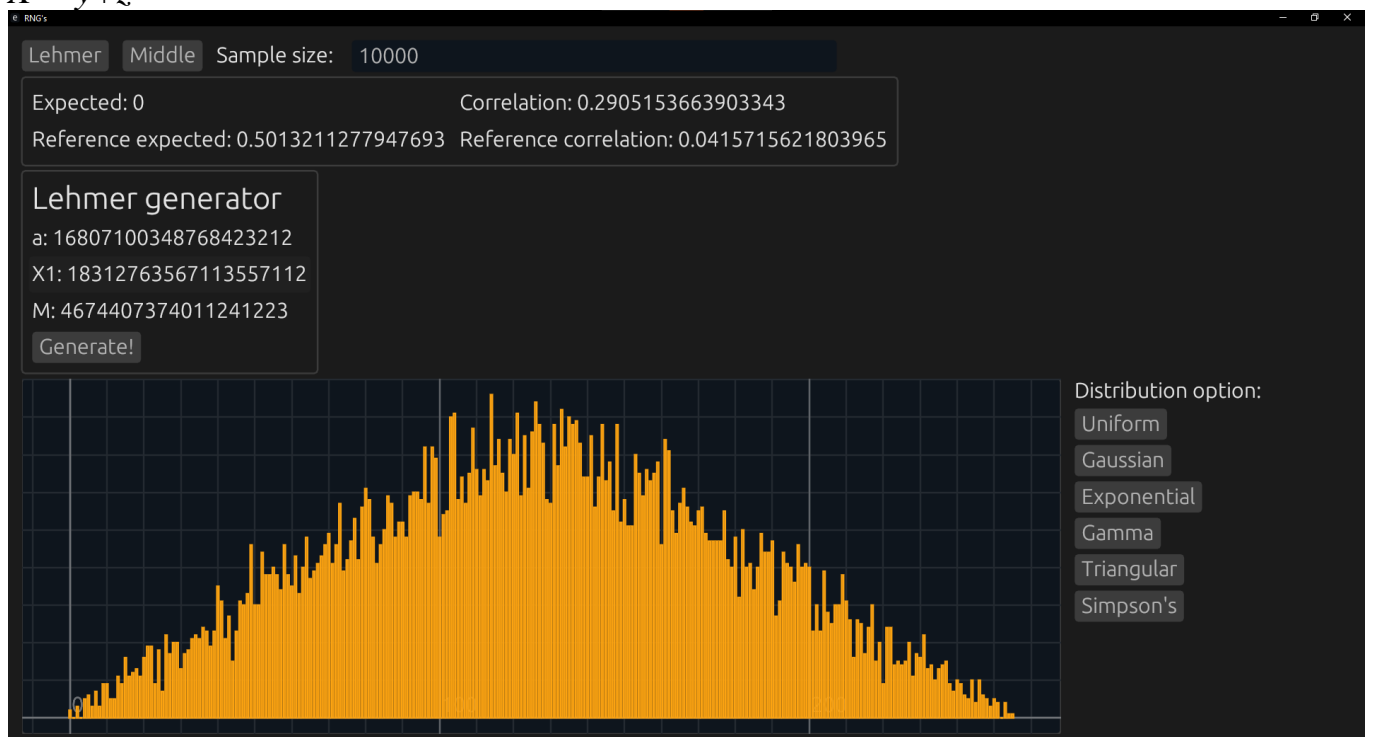
$$X = -\frac{1}{\lambda} \ln(\prod_{i=1}^{\eta} R_i), \lambda = 2.71828$$



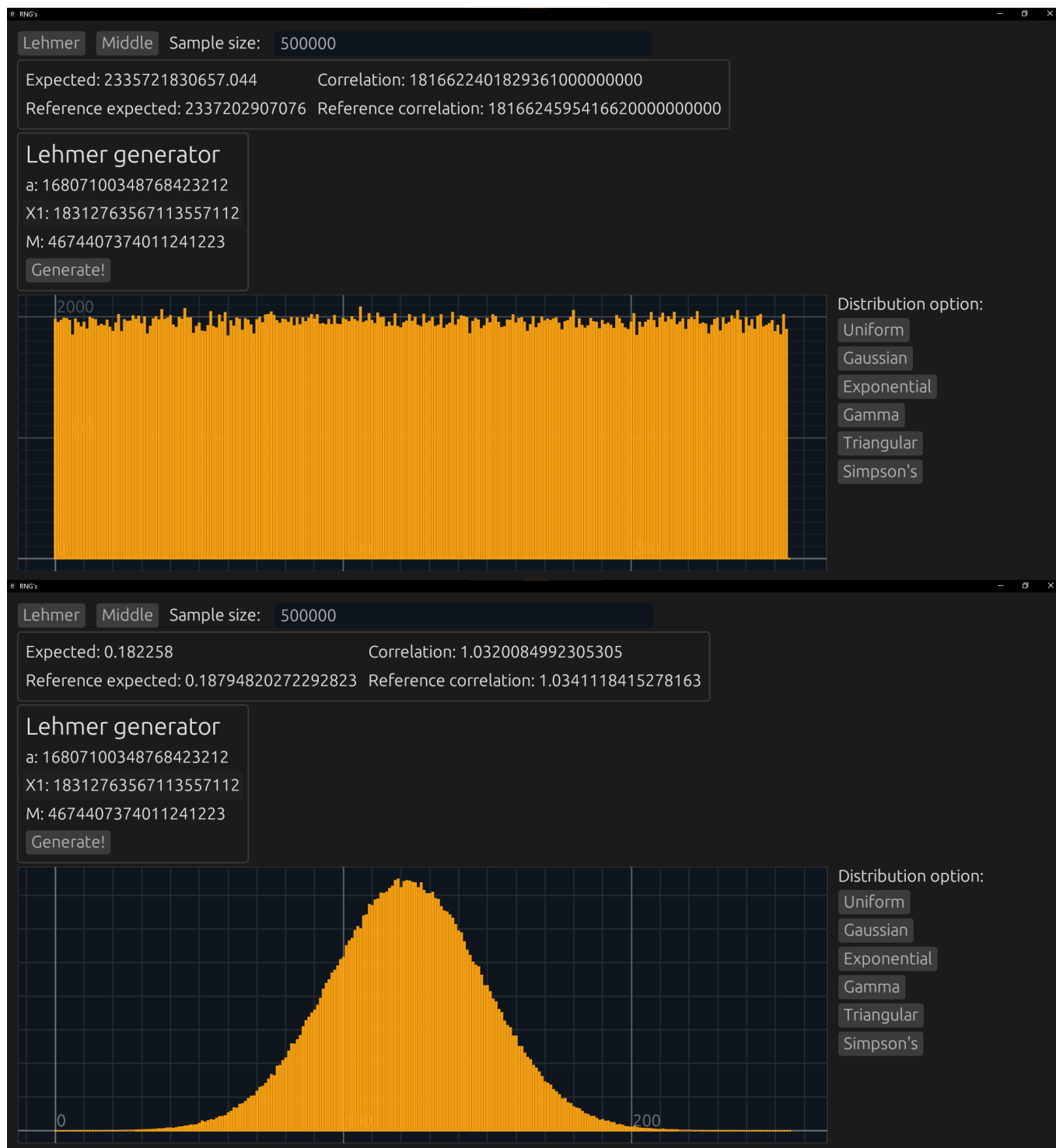
$$x = a + (b - a) \max(R_1, R_2), a = 452, b = 4248925791982749$$

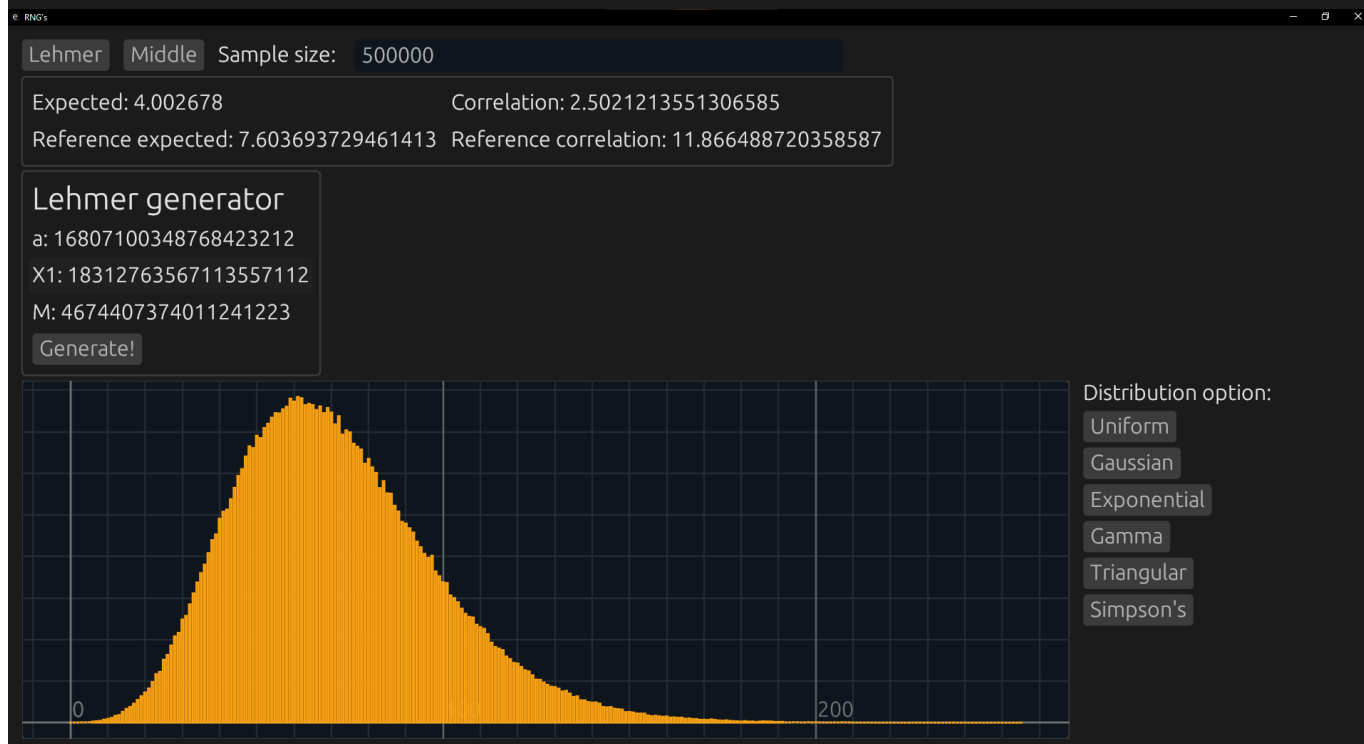
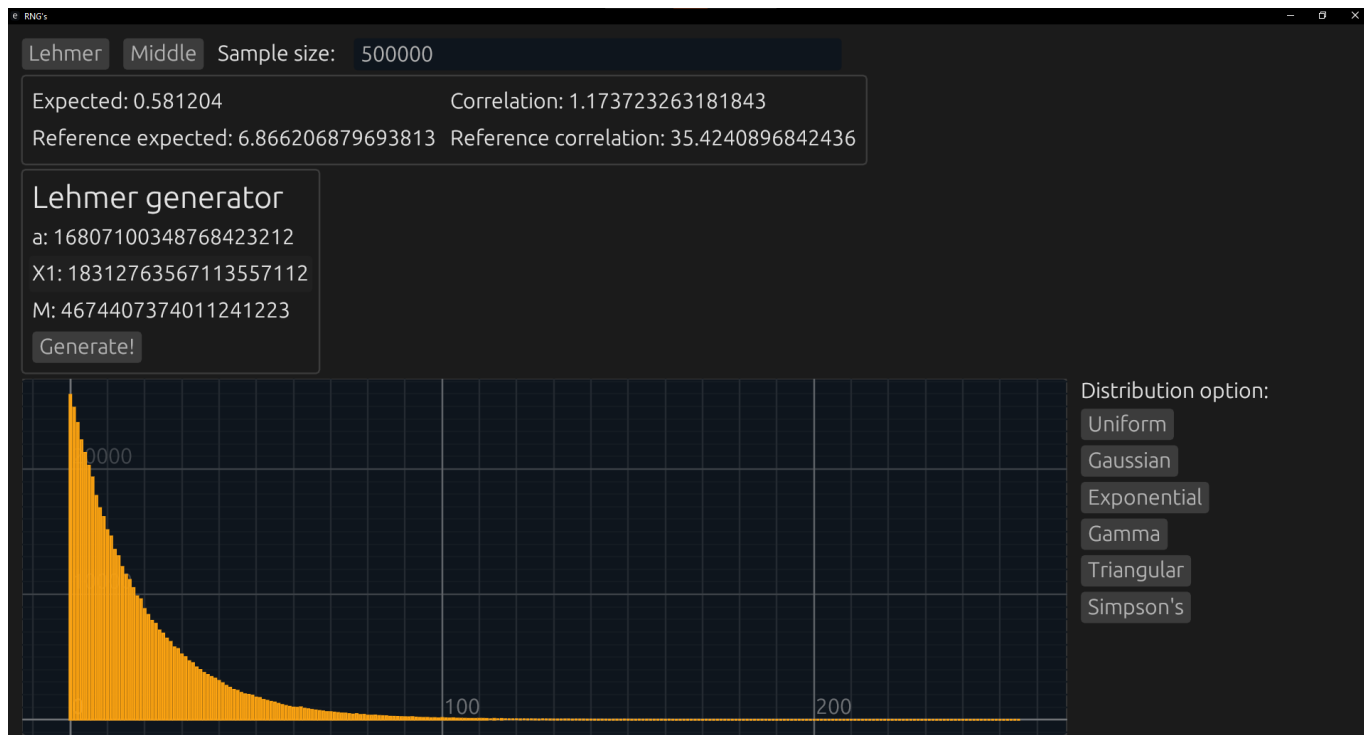


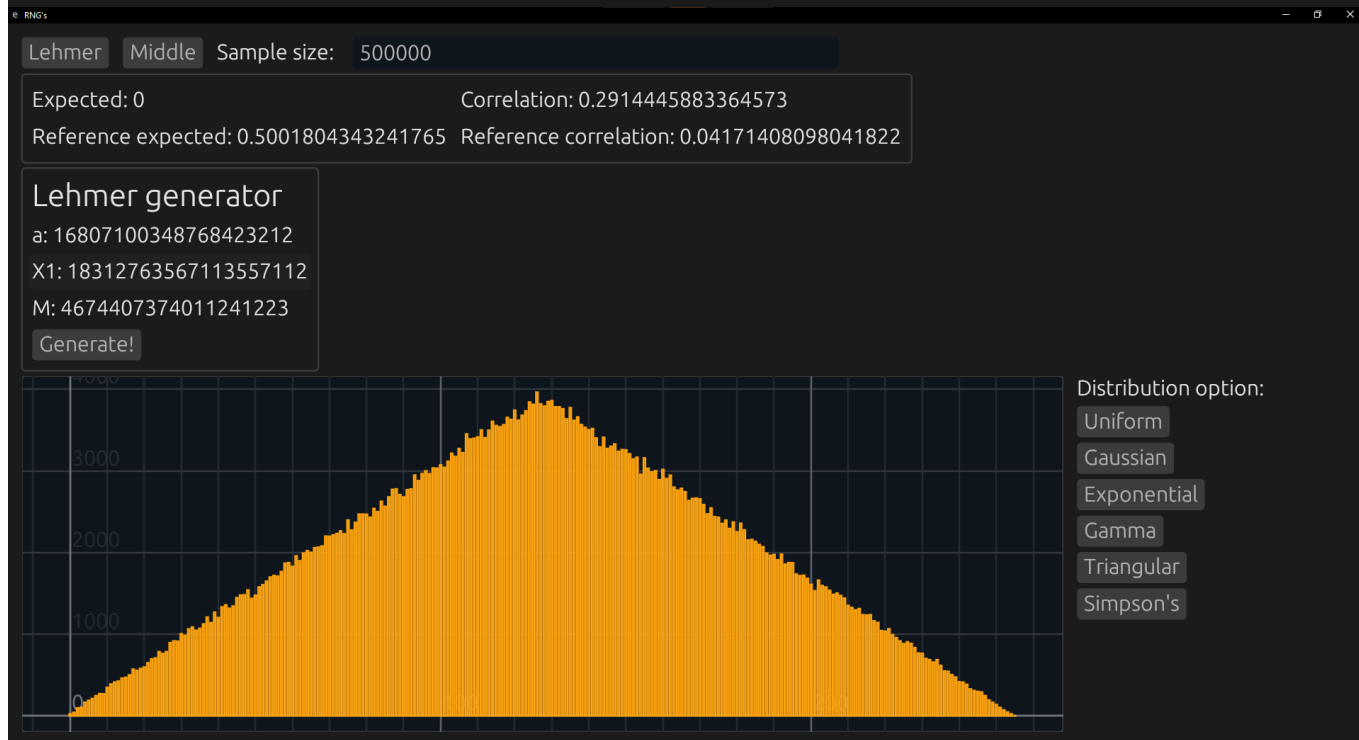
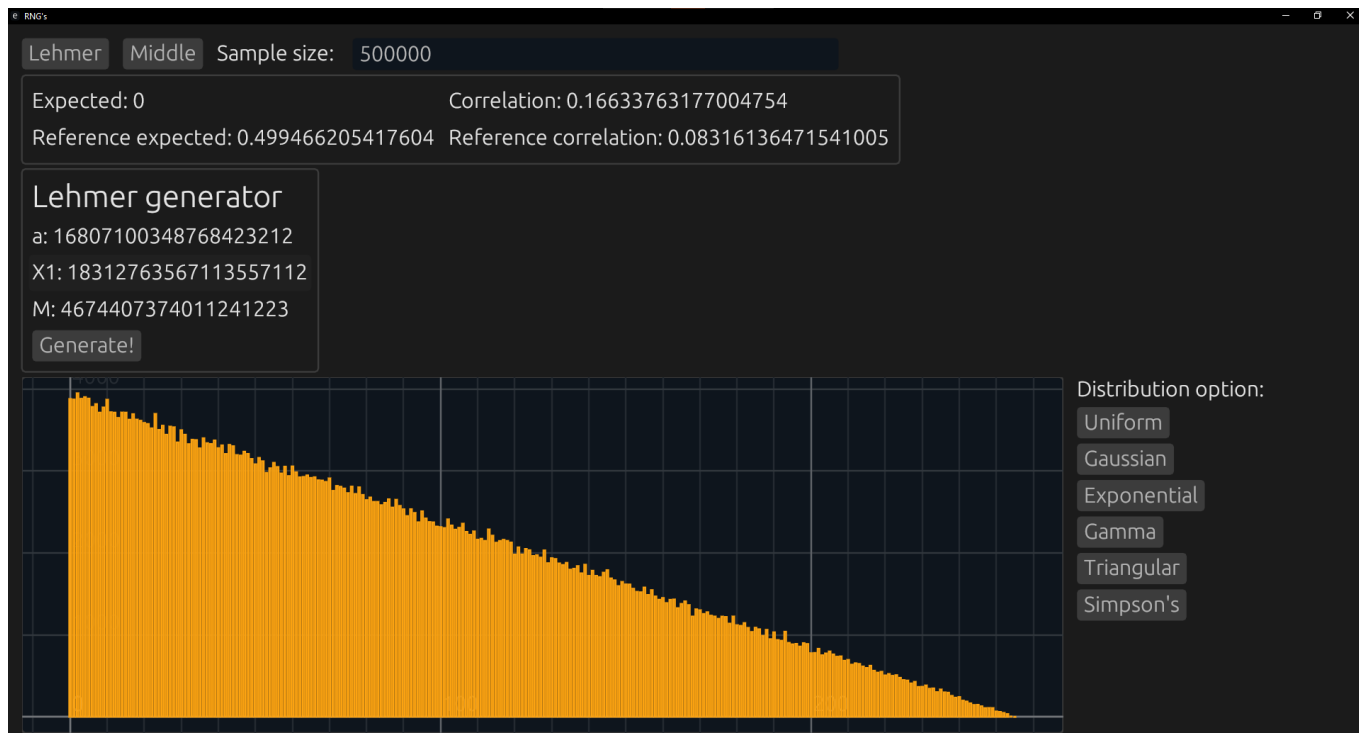
$$X = y+z$$



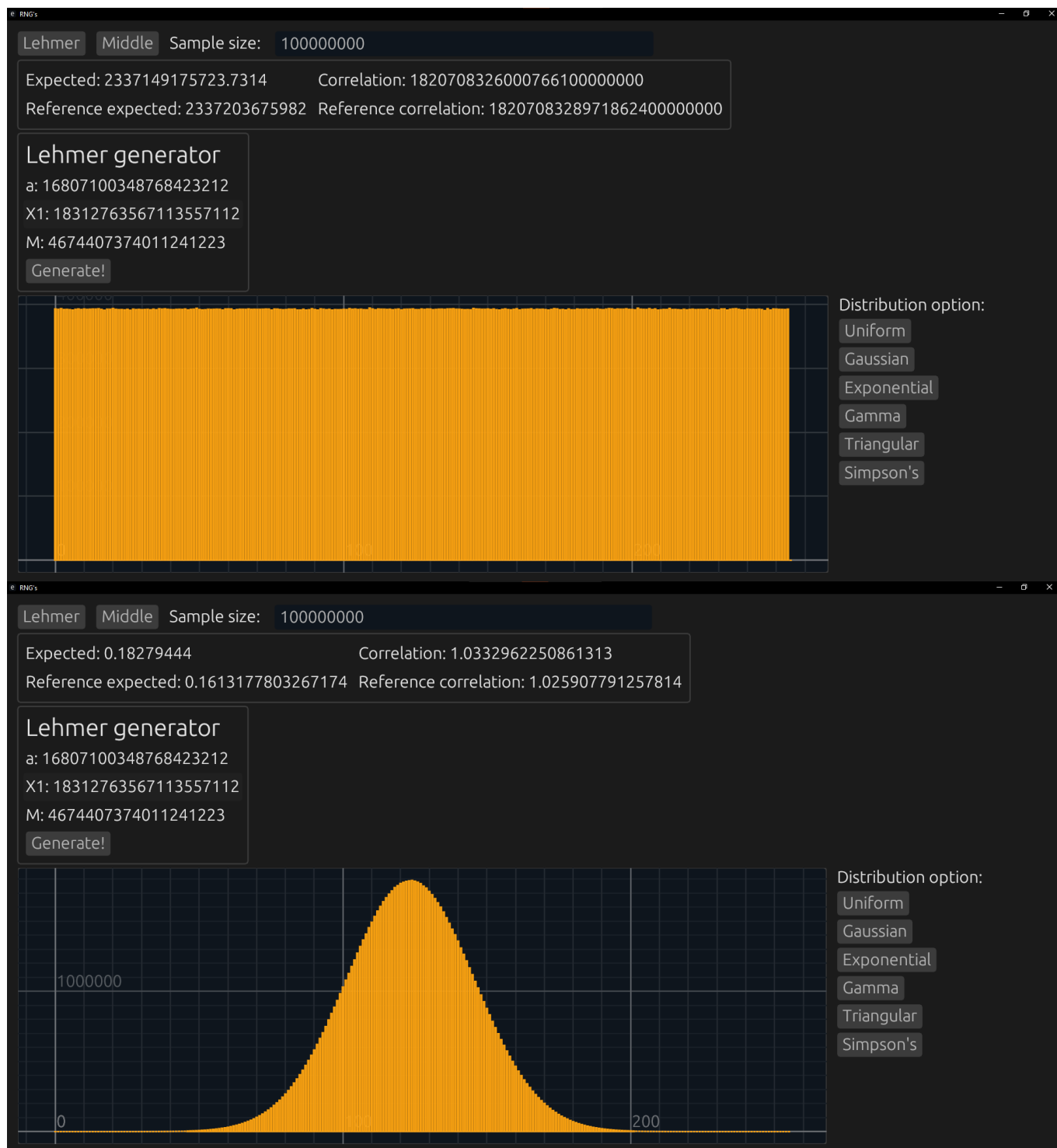
Генератор с выборкой равной $n = 500000$

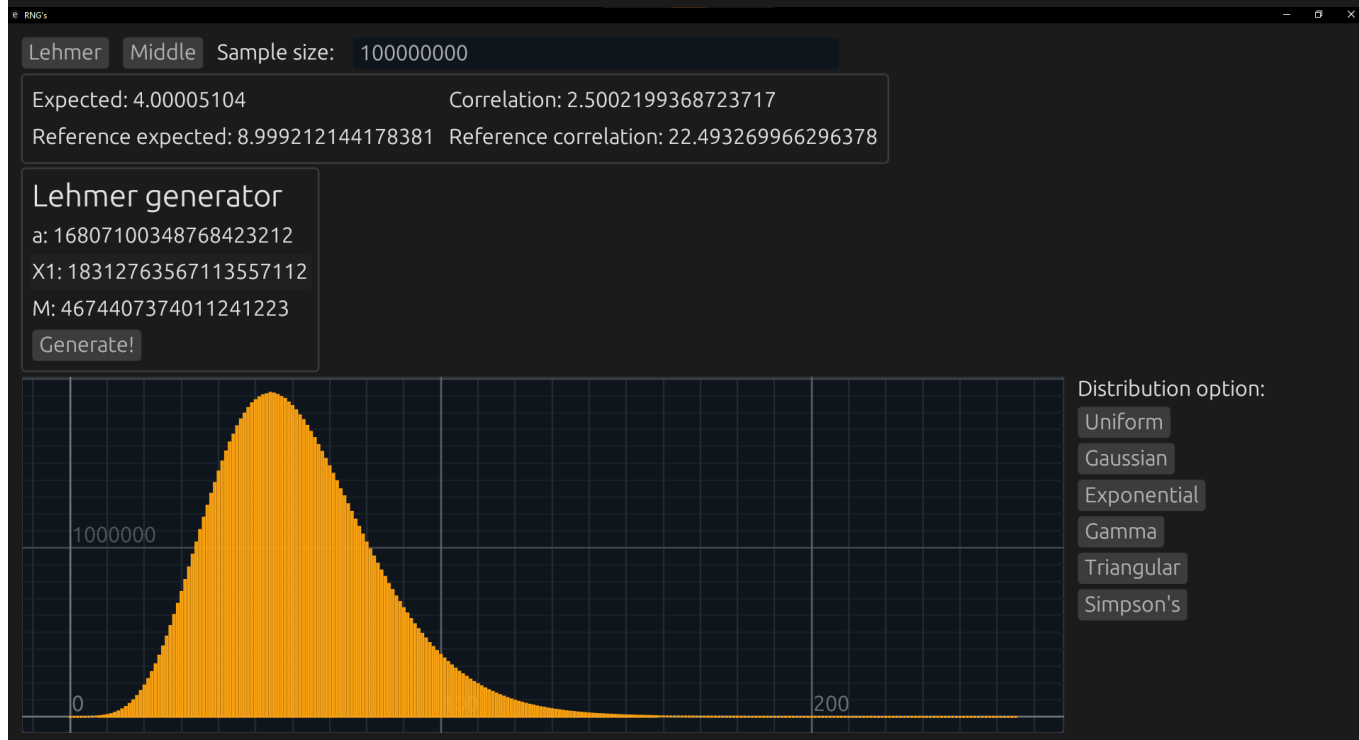


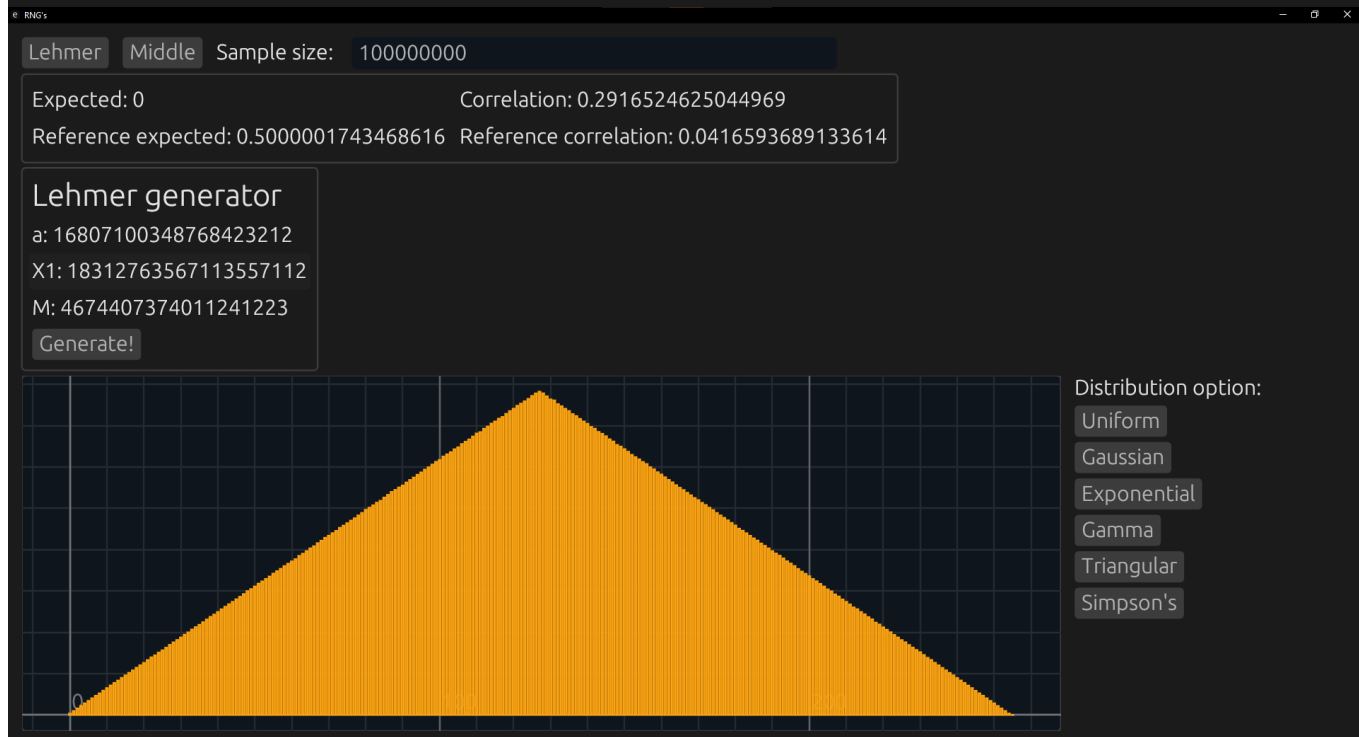
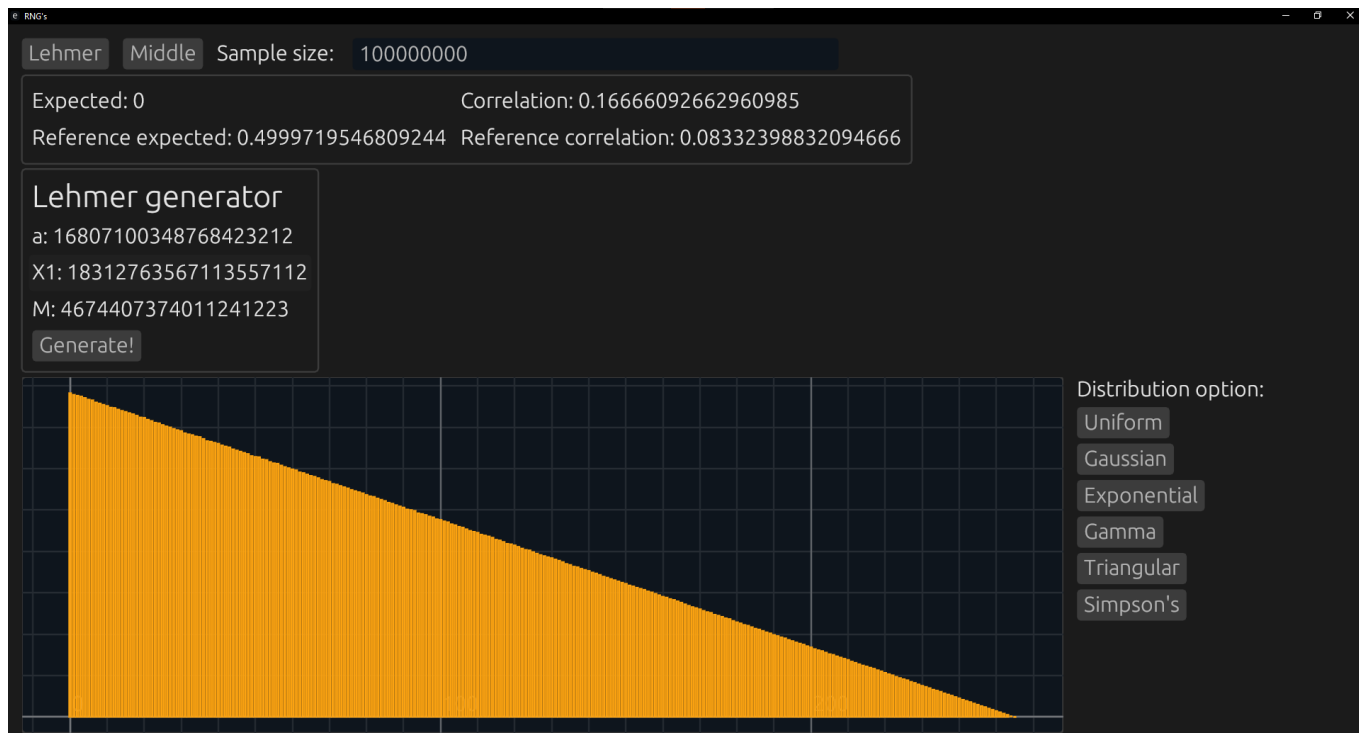




Генератор с выборкой равной $n = 100000000$







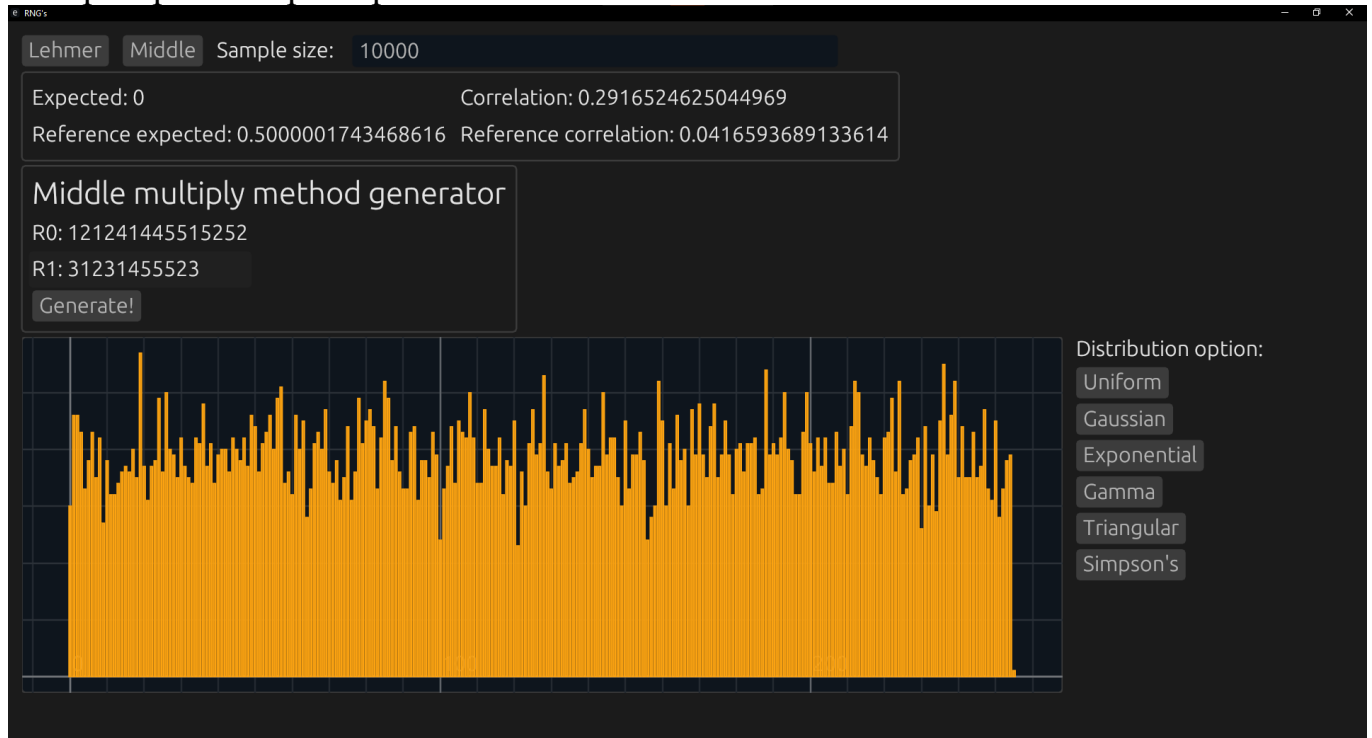
3.2 Генератор срединных произведений

В качестве начальных значений использовались следующие числа:

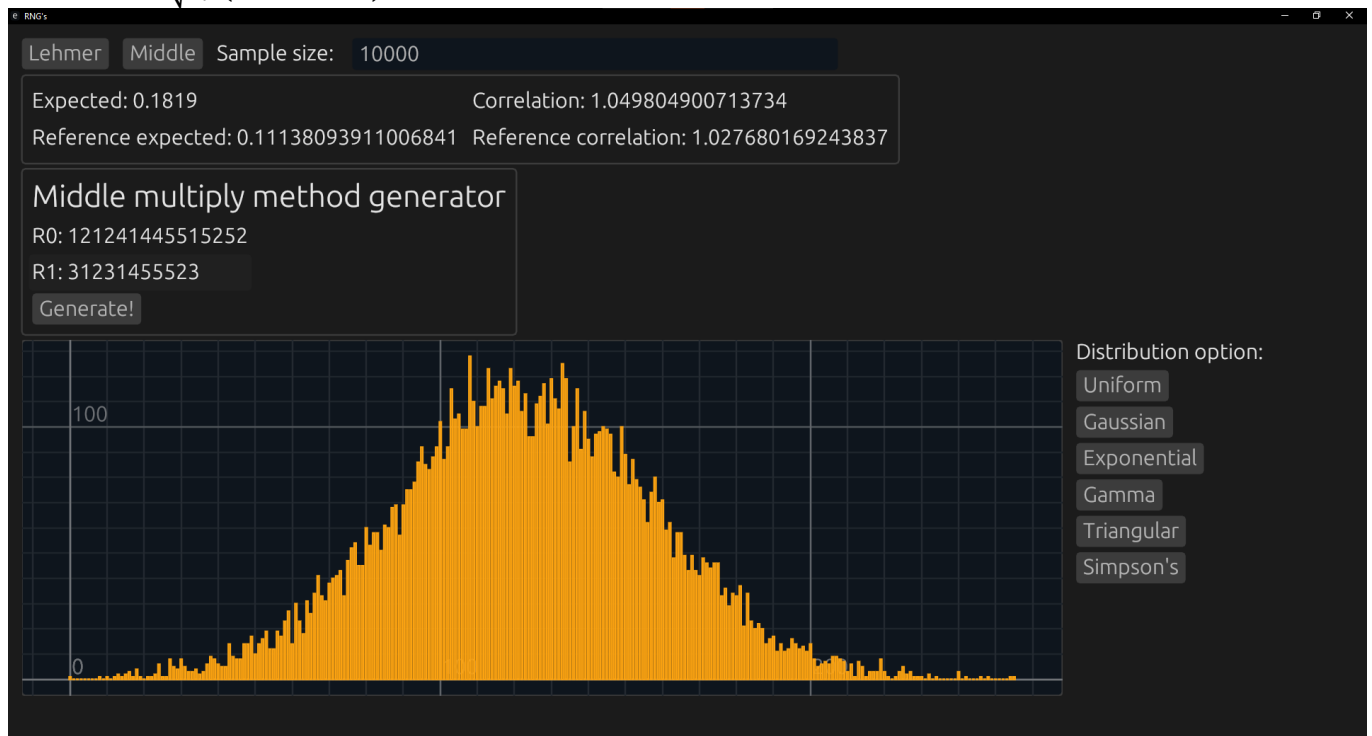
R0: 1.212.414.455.152.521.232

R1: 3.123.145.552.312.332.141

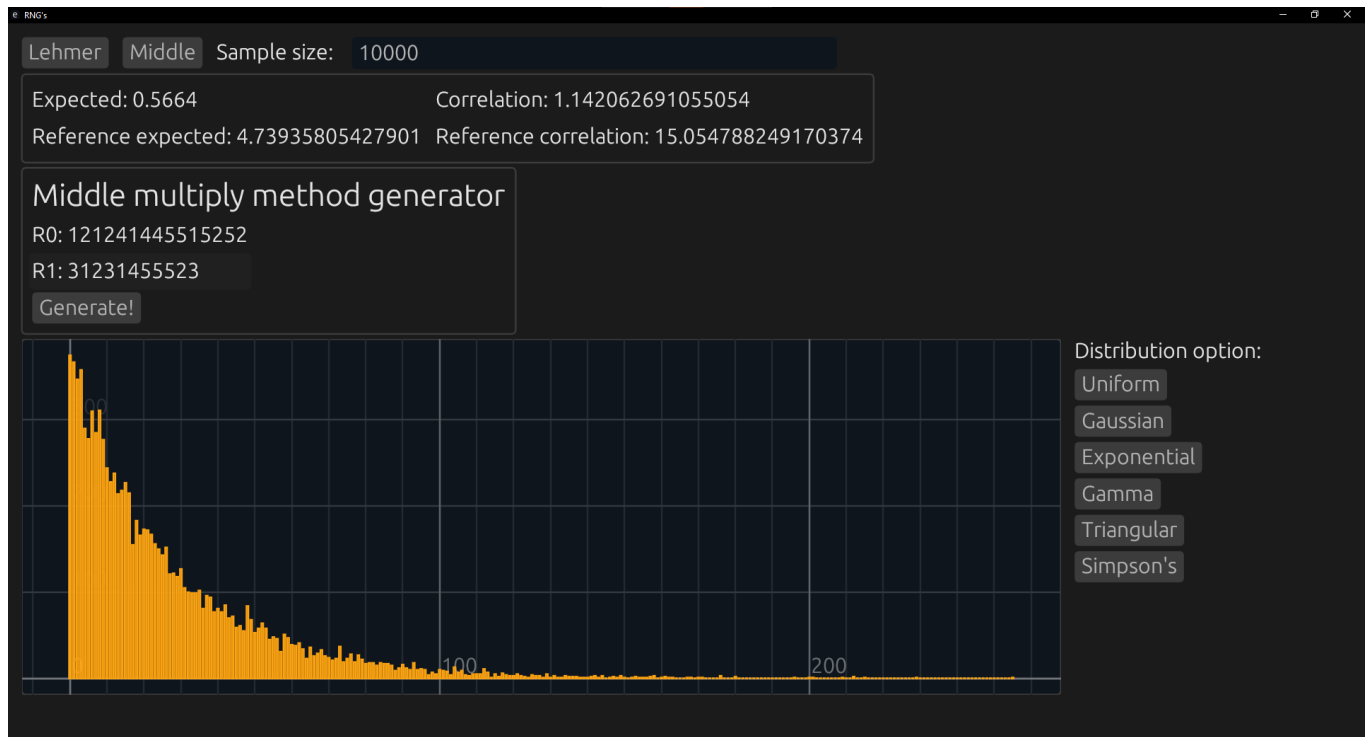
Генератор с выборкой равной $n = 10000$



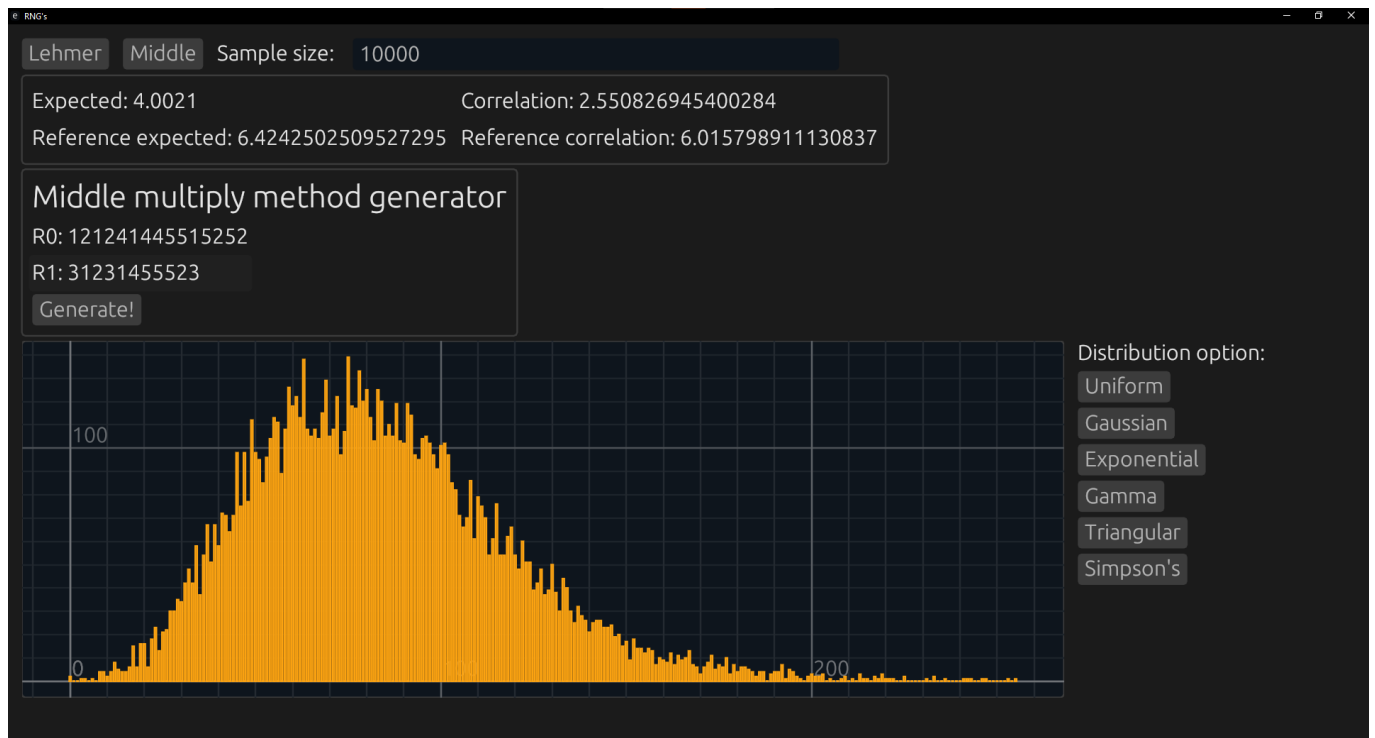
$$x = m_x + \sigma_x \sqrt{\frac{12}{n}} \left(\sum_{i=1}^n R_i - \frac{n}{2} \right), \sigma_x = 39, n = 6$$



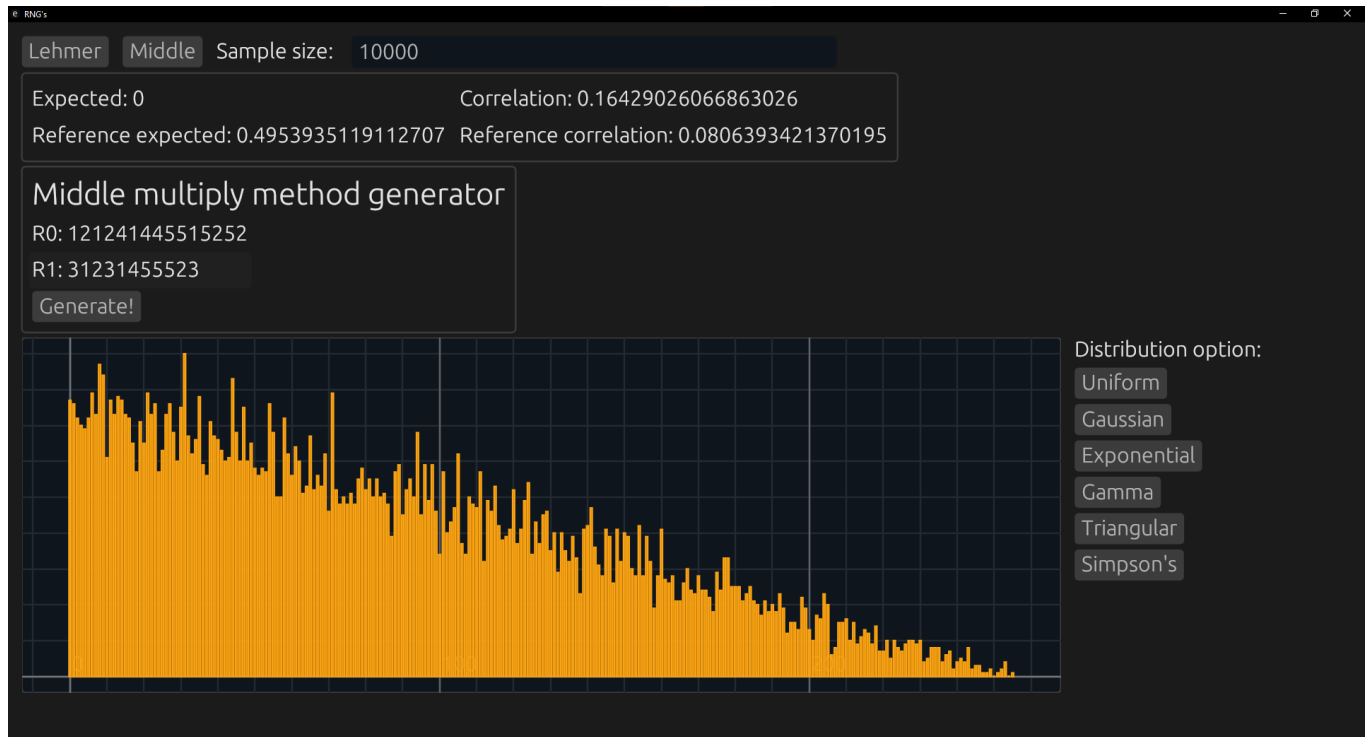
$$X = -\frac{1}{\lambda} \ln R. \lambda = 2,71828$$



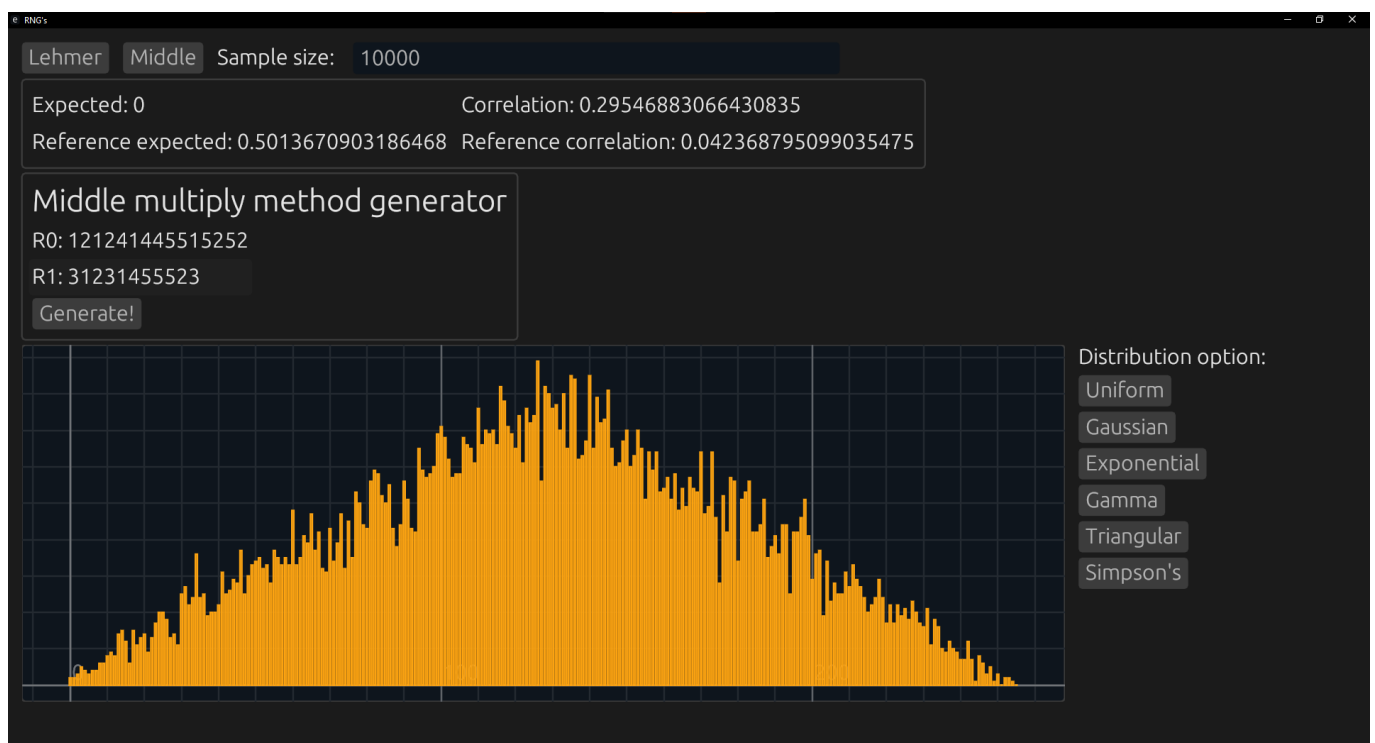
$$X = -\frac{1}{\lambda} \ln(\prod_{i=1}^{\eta} R_i), \lambda = 2.71828$$



$$x = a + (b - a) \max(R_1, R_2), a = 19, b = 449325284319556245$$

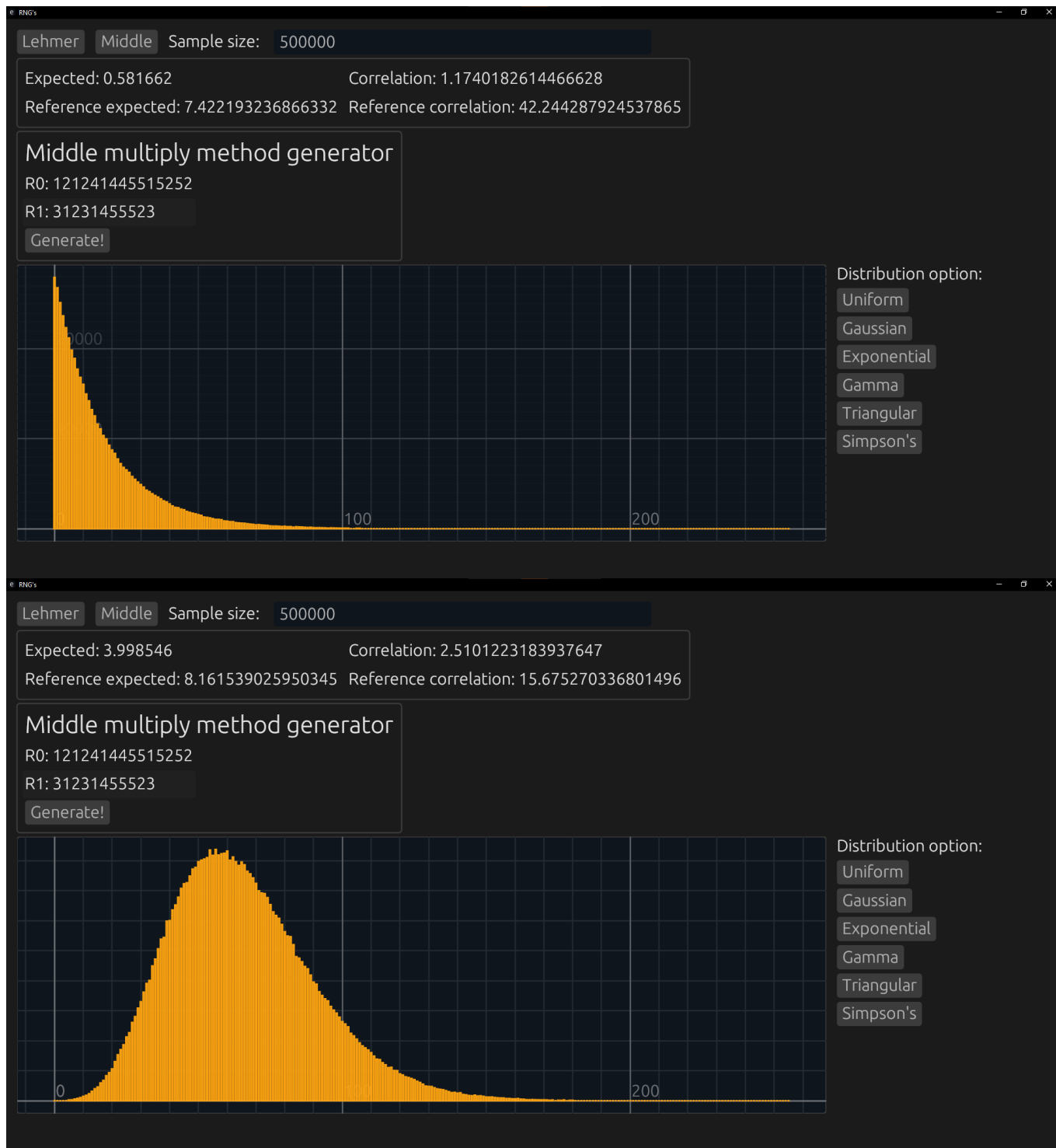


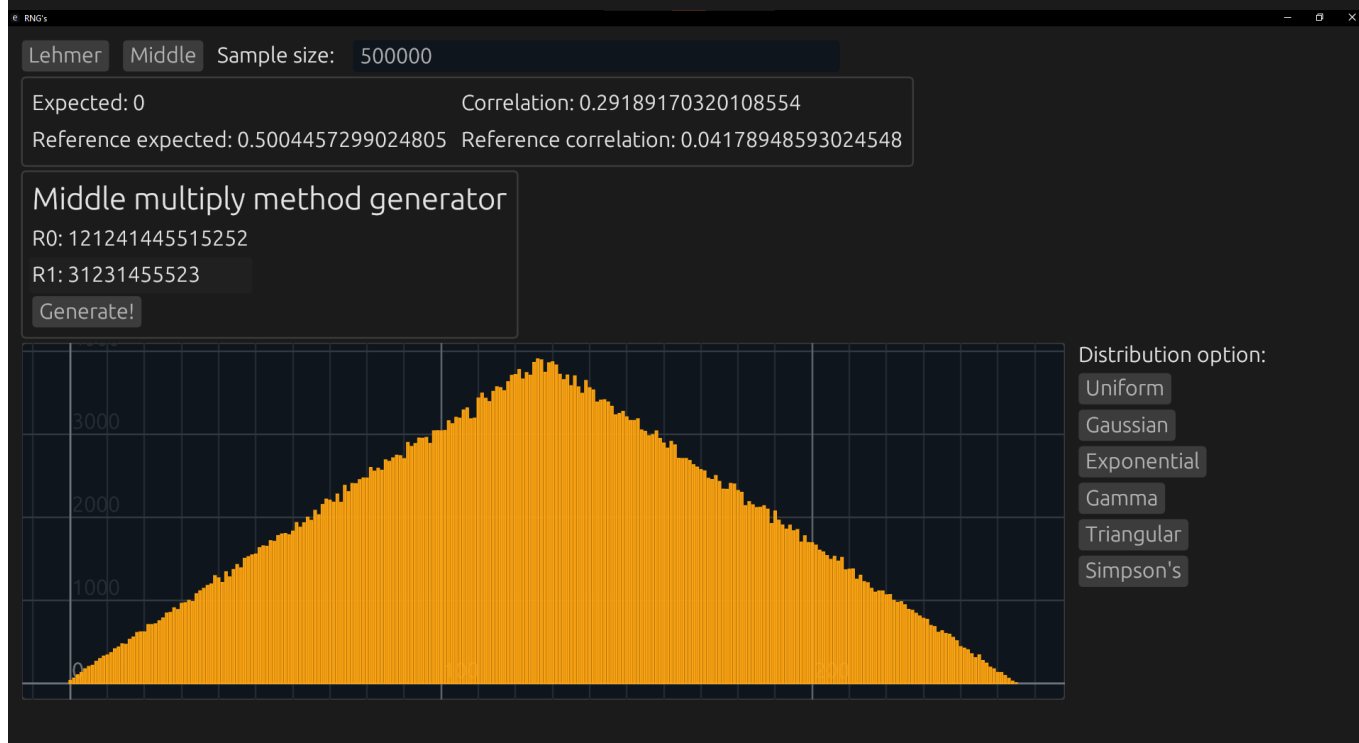
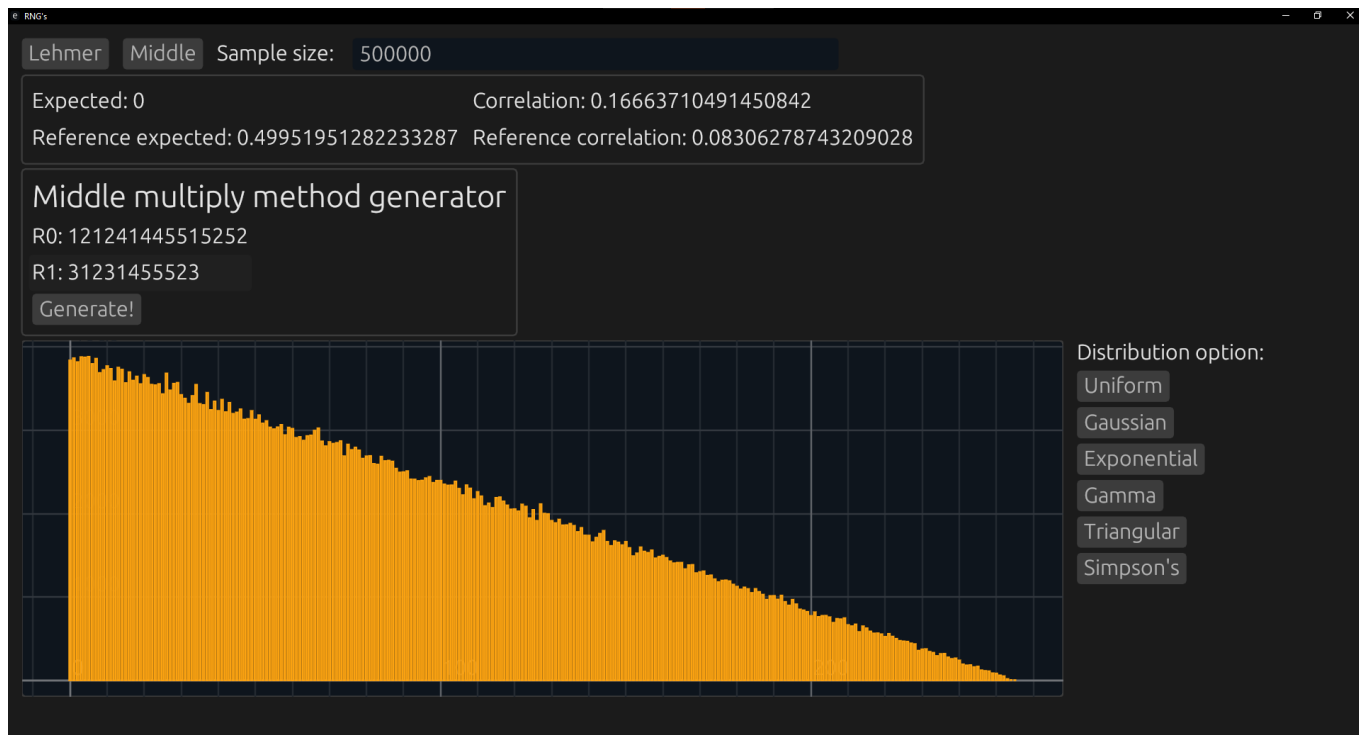
$$X = y + z$$



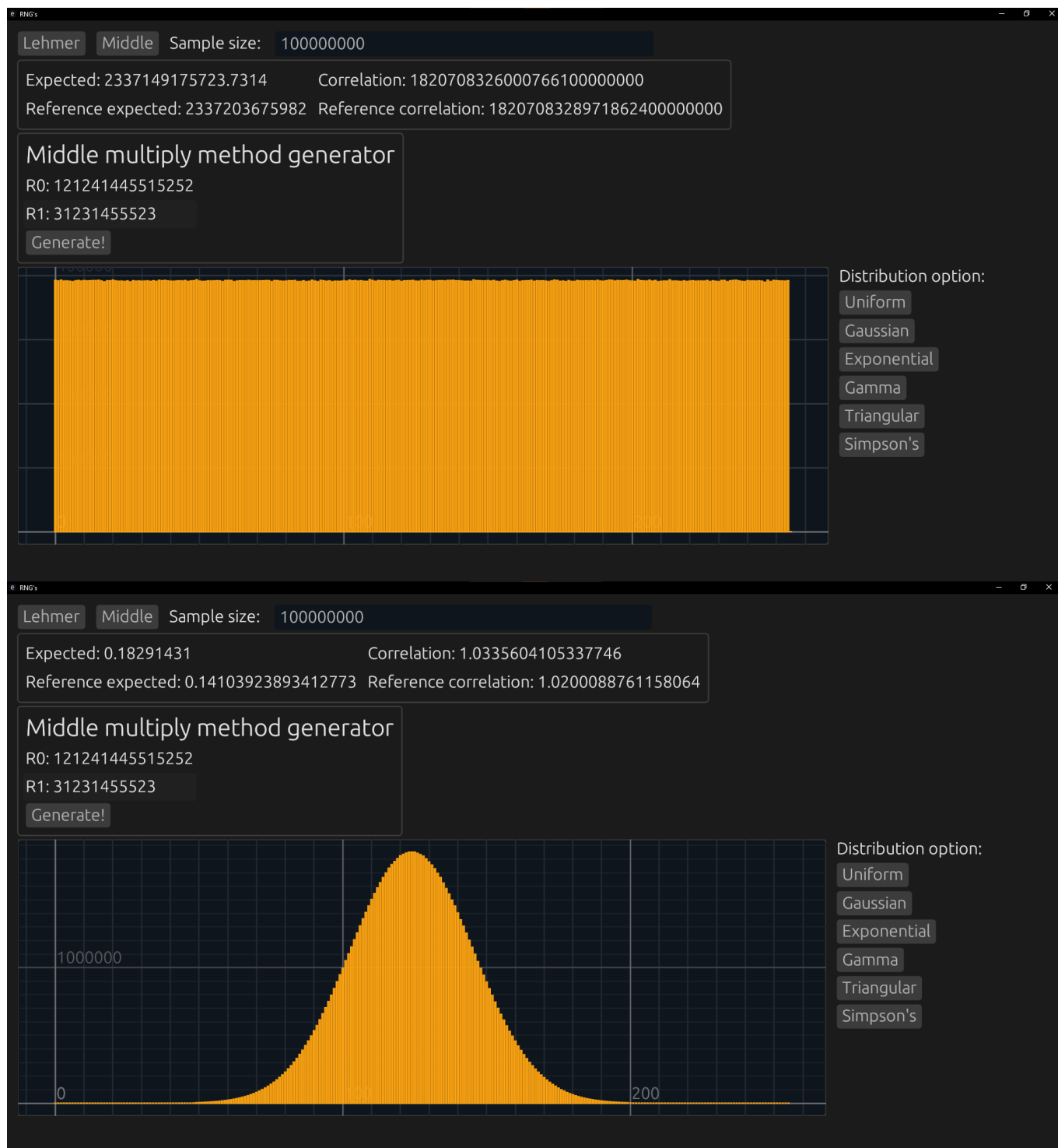
Генератор с выборкой равной $n = 500000$

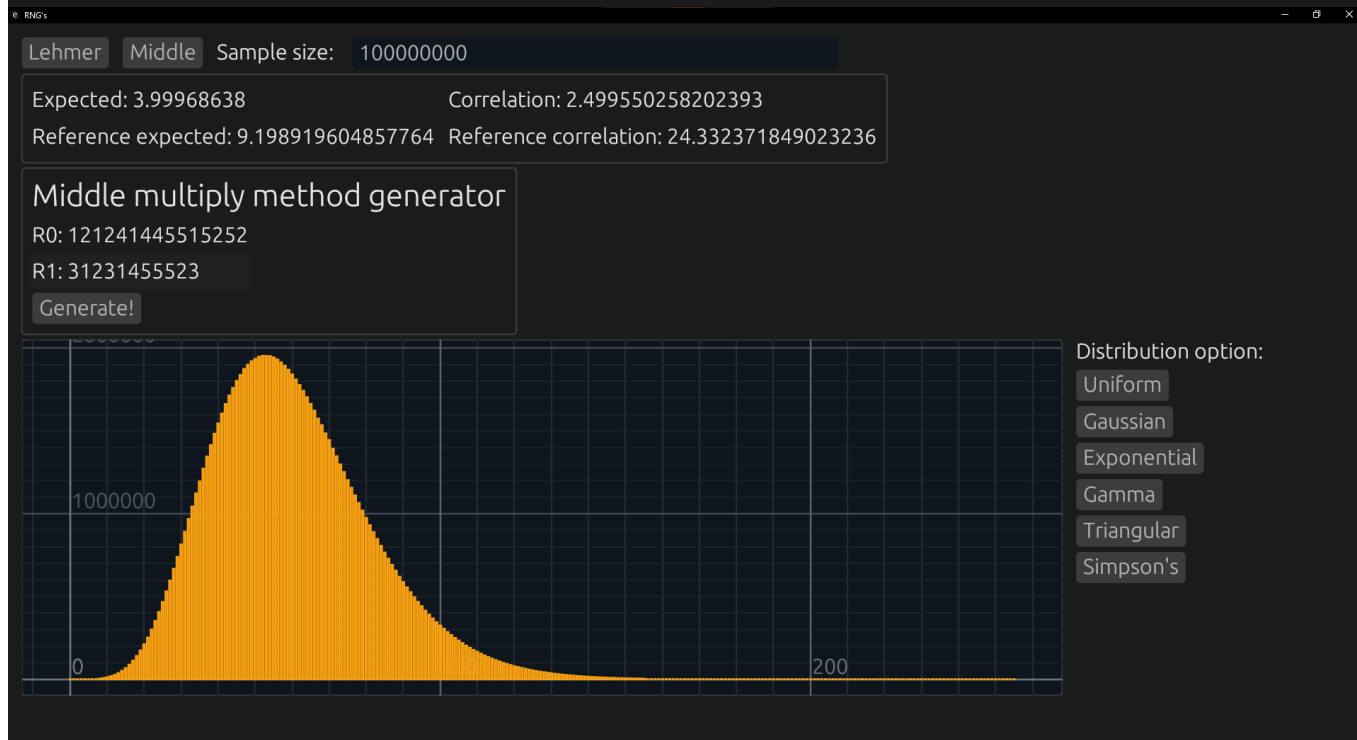
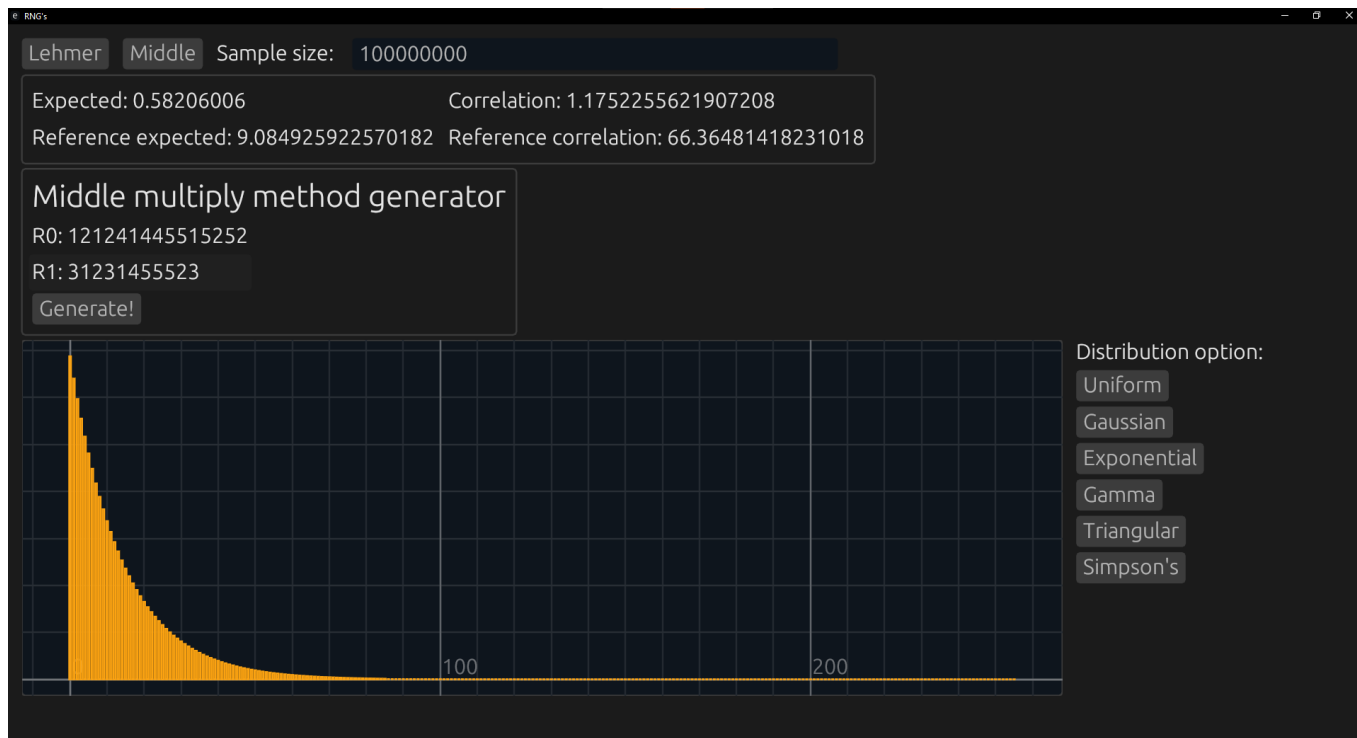


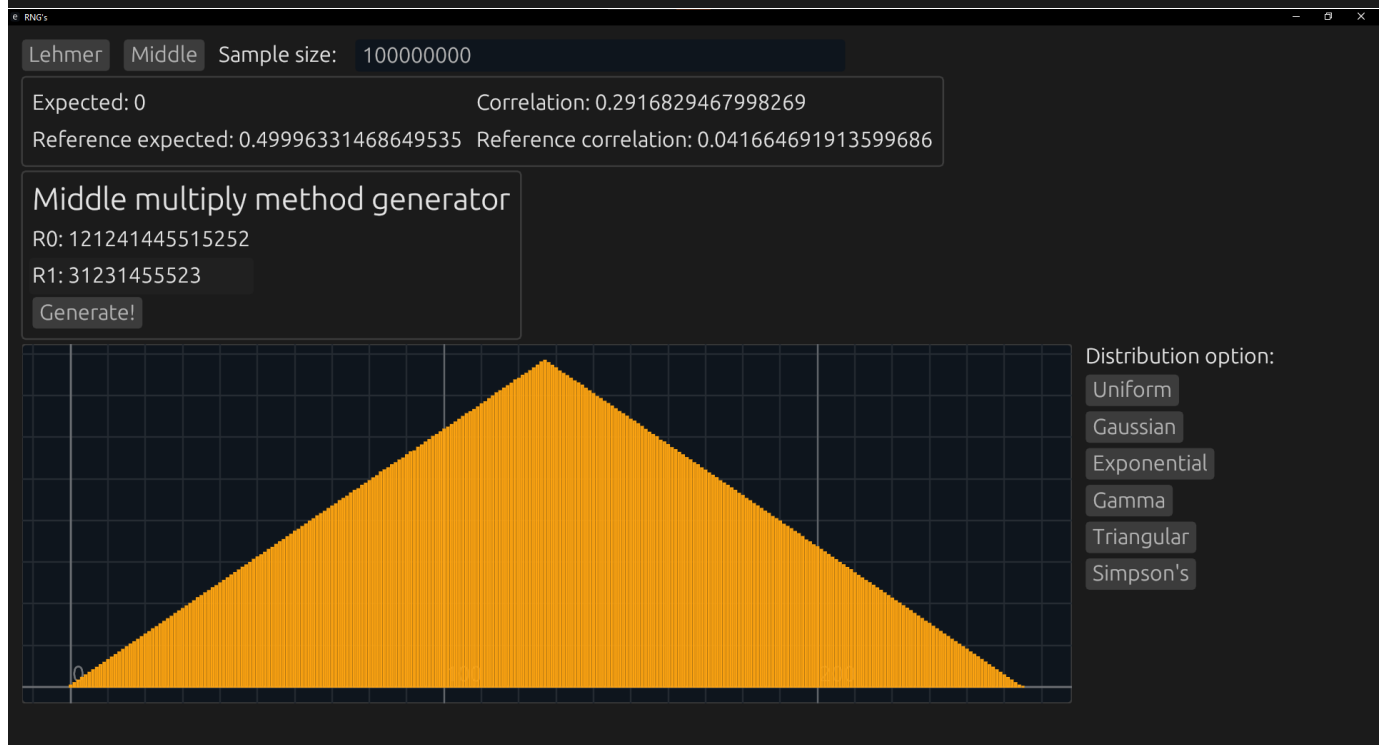
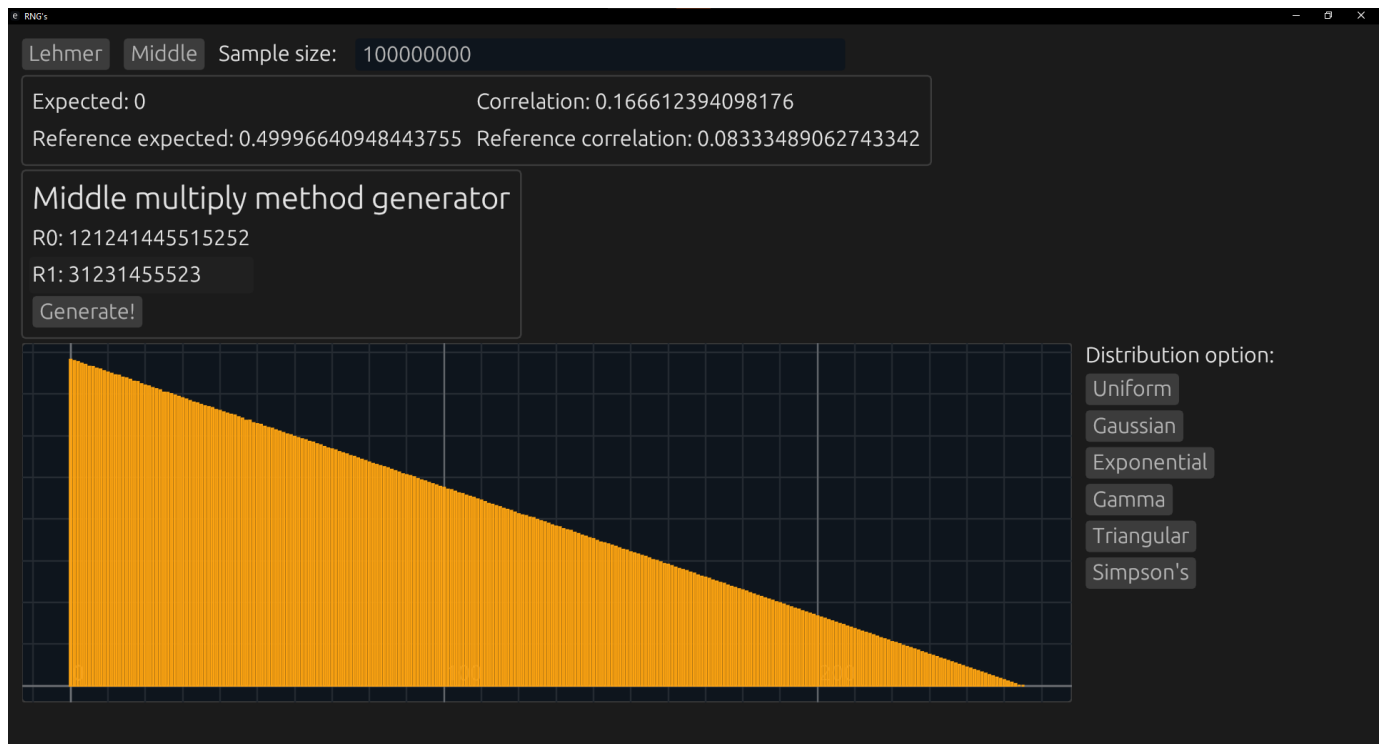




Генератор с выборкой равной $n = 100000000$







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 срединных 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);  
    }  
}
```