МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет електроніки і комп’ютерних технологій

Кафедра системного проєктування

**Звіт**

про виконання лабораторної роботи № 4

з дисципліни «Прикладна статистика та ймовірнісні процеси»

«Моделювання неперервних випадкових величин із заданим законом розподілу»

**Виконав:**

студент 2 курсу

групи Феп-22

Линва В.А.

**Перевірив:**

Доцент кафедри РКИ, к.ф.-м.н

Сас Н. Б.

**Львів 2022**

**Мета:** Ознайомитись із темою «Моделювання випадкових чисел із заданим законом розподілу». Виконати практичні завдання.

**Хід роботи**

1. Змоделював методом Неймана N=100 значень неперервної випадкової величини із заданою густиною розподілу ймовірності (таблиця 1). Номер варіанту - 3.

|  |
| --- |
|  |

2. Оцінив вибіркові математичне сподівання і дисперсію отриманої випадкової величини.

3. Побудував гістограму та оцінив за її допомогою закон розподілу отриманої випадкової величини.

4. Повторив п.1 -п.3для N=1000. Порівняв результати.

Результат роботи та гістограма для N=100

|  |
| --- |
|  |
|  |

Результат роботи та гістограма для N=1000

|  |
| --- |
|  |
|  |

Отож, ми бачимо схожість між теоретичними та практичними розрахунками. Із збільшенням розміру вибірки – похибка стає меншою.

Аналізуючи гістограму бачимо, що вони дійсно відповідають закону розподілу, який зображений у вигляді функції f(x) мого варіанту. На проміжку x Є [0;2] частота попадання збільшується і сама функція збільшується (пряма прямує вверх). На проміжку x Є [2;4] бачимо паралельну пряму до осі Ox, що свідчить, що показники частоти мають бути +- рівні. Це ми можемо побачити на гістограмах. Показники стають чіткіші при збільшені розміру вибірки.

**Висновок:** Ознайомився із темою «Моделювання випадкових чисел із заданим законом розподілу». Виконав практичні завдання.

**Додаток:**

#include <iostream>

#include <cmath>

#include <time.h>

#include <random>

#include <math.h>

using namespace std;

double func(double x)

{

double y = 0;

if (x >= 0 && x <= 2) // [0;2]

return y = 0.25 \* x;

else if (x > 2 && x <= 4) // (2;4]

return y = 0.25;

else

return 0;

}

double expectation(double\* mass, int size) // Математичне сподівання

{

double probality = 1.0 / (double)size;

double res = 0;

for (int i = 0; i < size; i++)

res += mass[i] \* probality;

return res;

}

double vib\_ser(double\* mass, int size) // Математичне вибіркове середнє

{

double result = 0;

double probality = 1.0 / (double)size;

for (int i = 0; i < size; i++)

result += pow(mass[i], 2) \* probality;

return result;

}

double variacy(double\* mass, int size) // Математична дисперсія

{

double result;

double probality = 1.0 / (double)size;

result = (vib\_ser(mass, size) - pow(expectation(mass, size), 2));

return result;

}

int main()

{

srand(time(NULL));

const int size = 1000;

double arr[size] = {};

double r1, r2;

double X0 = 0;

double n = 0;

int idx = 0;

double M = 0.5; // y - Максимум

double a = 0; // x Є [0;4]

double b = 4; // x Є [0;4]

//Інтервали

int count\_of\_intervals = ceil((1.0 + 3.332 \* log10(size)));

double distance = (b - a) / count\_of\_intervals; // розмір кожного інтервала

double\* interval = new double[count\_of\_intervals];

for (int i = 0; i < count\_of\_intervals; i++) // заповнюємо масив значеннями інтервалів

interval[i] = 0;

//Заповнення масиву

for (int i = 0; i < size; i++)

{

r1 = (double)rand() / RAND\_MAX;

r2 = (double)rand() / RAND\_MAX;

X0 = a + r1 \* (b - a);

n = r2 \* M;

if (n <= func(X0)) // Якщо випадкове число менше за значення функції в даному інтервалі

{

arr[i] = X0;

idx = floor((X0 - a) / distance); // позиція в масиві

interval[idx]++;

}

else // Якщо випадкове число більше за значення функції в даному інтервалі

i--;

}

// Виведення масиву

for (int i = 0; i < size; i++)

{

cout << arr[i] << " ";

}

// Виведення результатів

cout << "\n\n\n";

cout << "Expecation = " << expectation(arr, size) << "\t-\tTheoretical expecation = 2.16" << endl;

cout << "Viborochne serednye = " << vib\_ser(arr, size) << "\t-\tTheoretical viborochne serednye = 5.7" << endl;

cout << "Variacy = " << variacy(arr, size) << "\t-\tTheoretical Variacy = 1"<<endl;

cout << endl << "[Interval] [Frequency] [Relative frequency]" << endl;

for (int i = 0; i < count\_of\_intervals; i++)

{

cout << i << "\t\t " << interval[i] << "\t\t " << (double)interval[i] / (double)count\_of\_intervals << endl;

}

}