

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА
Факультет електроніки і комп'ютерних технологій
Кафедра системного проектування

Звіт

про виконання лабораторної роботи № 4
з дисципліни «Прикладна статистика та ймовірнісні процеси»
«Модельовання неперервних випадкових величин із заданим законом розподілу»

Виконав:

студент 2 курсу
групи Феп-22
Линва В.А.

Перевірив:

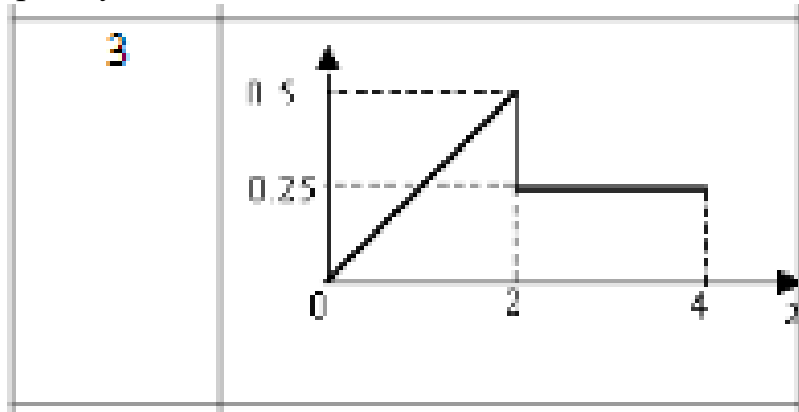
Доцент кафедри РКІ, к.ф.-м.н
Сас Н. Б.

Львів 2022

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

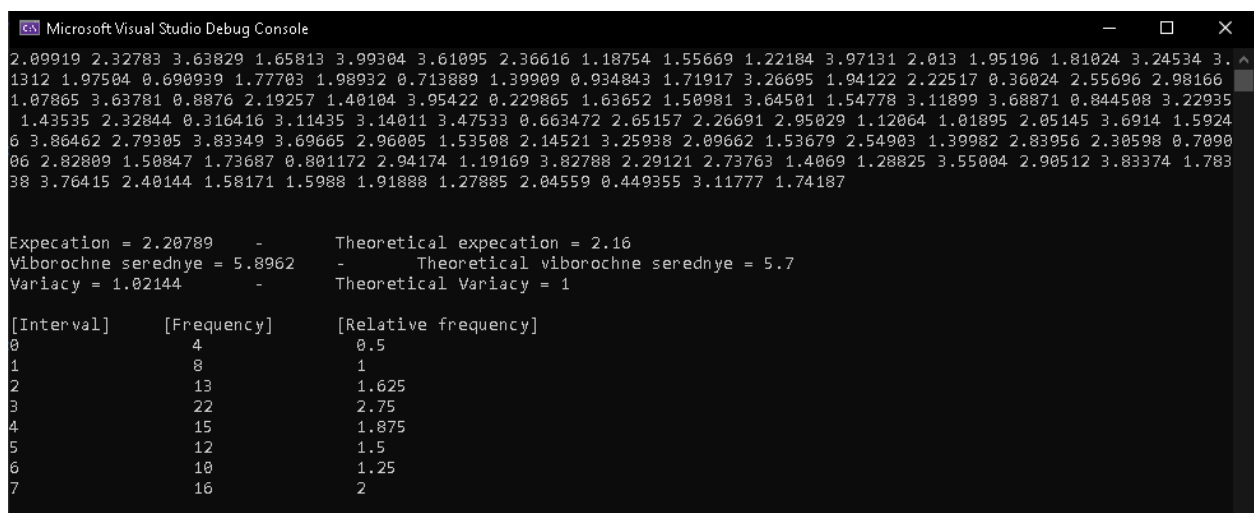
Хід роботи

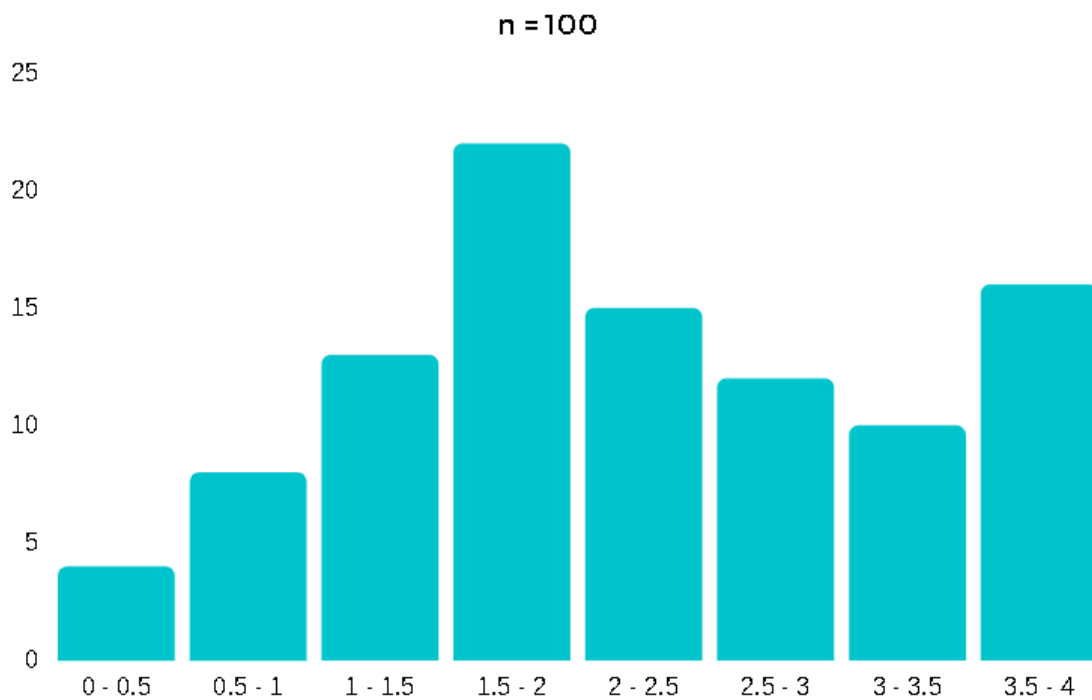
1. Змодельовав методом Неймана $N=100$ значень неперервної випадкової величини із заданою густиною розподілу ймовірності (таблиця 1).
Номер варіанту - 3.



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.3 для $N=1000$. Порівняв результати.

Результат роботи та гістограма для $N=100$





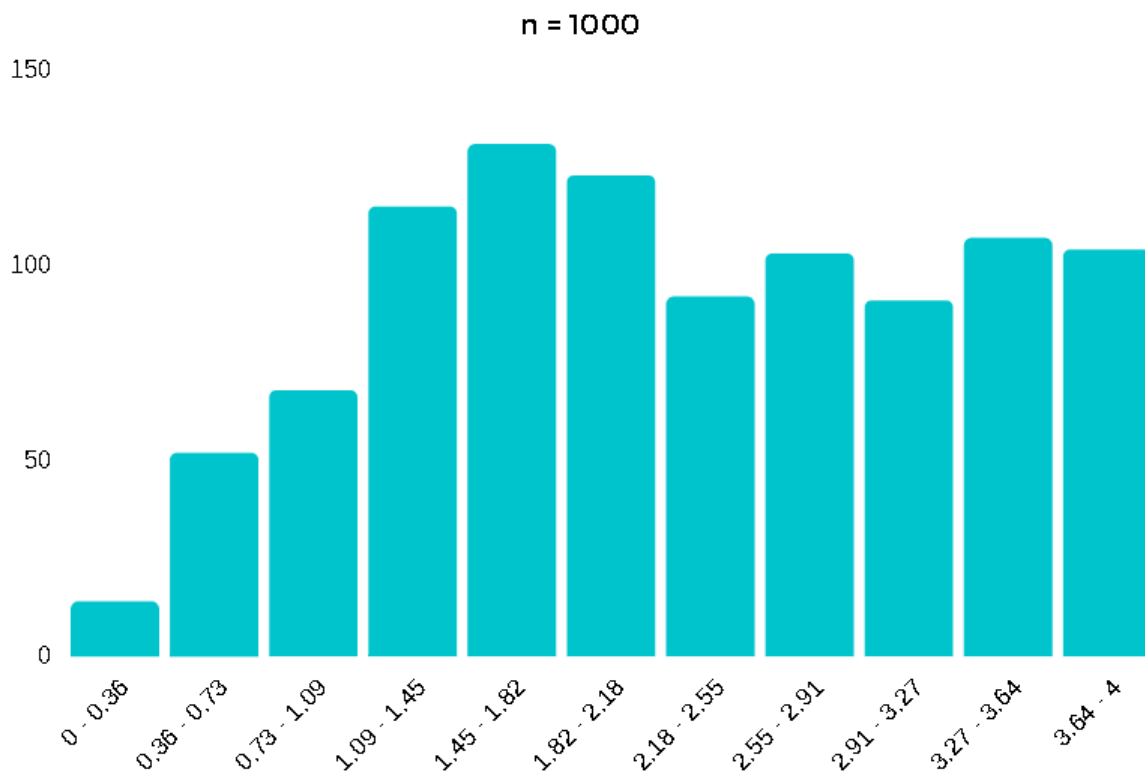
Результат роботи та гістограма для N=1000

```

Microsoft Visual Studio Debug Console
76 3.5725 2.76827 2.82138 2.18268 1.96051 0.377941 1.45647 2.98837 3.74511 1.70171 1.85199 2.60762 2.51436 3.53587 2.894
5 1.01126 1.86688 2.7297 1.6906 1.67364 2.20527 3.03867 2.30342 1.9245 3.70739 2.0174 1.77276 1.48796 0.573748 3.14658 3
.3375 1.7315 2.00739 1.42228 2.57393 3.96509 2.98337 0.977935 0.64919 2.25581 1.58098 2.26496 3.18027 1.22587 1.47172 3.
66808 1.38041 2.18061 1.16349 3.53392 2.74642 2.0235 1.87469 3.4254 1.08499 1.46086 1.99957 2.72762 1.93231 1.47893 3.78
283 1.61773 1.90619 3.86767 0.625141 2.73177 3.67821 0.874783 3.08786 2.57332 1.59087 3.76879 3.95141 1.60015 0.100345 2
.24421 2.8309 1.77386 0.893704 3.99316 2.14313 1.49614 1.03519 1.09586 2.38227 1.40239 1.9881 1.1442 2.8088 2.44551 3.92
541 3.19053 3.36802 1.70904 1.98859 1.30155 1.88885 2.84933 0.743797 2.55061

Expectation = 2.24503 - Theoretical expectation = 2.16
Viborochne serednye = 6.03304 - Theoretical viborochne serednye = 5.7
Variacy = 0.992863 - Theoretical Variacy = 1

[Interval] [Frequency] [Relative frequency]
0 14 1.27273
1 52 4.72727
2 68 6.18182
3 115 10.4545
4 131 11.9091
5 123 11.1818
6 92 8.36364
7 103 9.36364
8 91 8.27273
9 107 9.72727
10 104 9.45455
  
```



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 $f(x)$ того варіанту. На проміжку $x \in [0;2]$ частота попадання збільшується і сама функція збільшується (пряма прямує вгору). На проміжку $x \in [2;4]$ бачимо паралельну пряму до осі Ox , що свідчить, що показники частоти мають бути \pm рівні. Це ми можемо побачити на гістограмах. Показники стають чіткіші при збільшенні 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 probability = 1.0 / (double)size;
    double res = 0;
    for (int i = 0; i < size; i++)
        res += mass[i] * probability;
    return res;
}

double vib_ser(double* mass, int size) // Математичне вибіркове середнє
{
    double result = 0;
    double probability = 1.0 / (double)size;
    for (int i = 0; i < size; i++)
        result += pow(mass[i], 2) * probability;
    return result;
}

double variacy(double* mass, int size) // Математична дисперсія
{
    double result;
    double probability = 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 ∈ [0;4]
    double b = 4; // x ∈ [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 << "Expection = " << expectation(arr, size) << "\t\tTheoretical expection = 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;
}

}

```