Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Программное обеспечение ЭВМ и информационные технологии (ИУ-7)

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

НА ТЕМУ:

Исследование псевдослучайных последовательностей

Студент ИУ7-78 А.В. Иванников
(Группа) (И.О.Фамилия)
Вариант 4

Преподаватель И.В. Рудаков
(И.О.Фамилия)

2020 г.

Выбор языка программирования приложения

В качестве языка программирования приложения был выбран язык C#.

Псевдослучайные числа

Псевдослучайное число - число, полученное в соответствии с заданным алгоритмом, используемое в качестве случайного числа.

Последовательности из таких чисел обязательно образуют циклы. Повторяющиеся циклы называют периодом. Период последовательности псевдослучайных чисел – одно из ключевых её свойств, которое оценивается в первую очередь.

Существует три способа получения последовательностей псевдослучайных чисел:

- алгоритмический;

- табличный;

- ручной ввод.

Алгоритмический способ

Способ получения последовательностей псевдослучайных чисел, основанный на формировании случайных чисел в ЭВМ с помощью специальных алгоритмов и реализующих их программ.

Каждое псевдослучайное число вычисляется с помощью соответствующей программы по мере возникновения потребностей при моделировании системы на ЭВМ.

Генератор псевдослучайных чисел (ГПСЧ) - алгоритм, генерирующий последовательность чисел, элементы которой почти независимы друг от друга и подчиняются заданному распределению. Зачастую используется равномерное распределение.

Генераторы псевдослучайных чисел могут работать по разным алгоритмам. Одним из простейших генераторов является так называемый линейный конгруэнтный генератор, который для вычисления очередного числа x_i использует формулу: $x_i = (a \cdot x_{i-1} + b) \bmod c$, где a, b, m — некоторые константы, x_{i-1} — предыдущее псевдослучайное число. Для получения k_1 задается начальное значение seed.

В качестве примера были приняты a = 5, b = 3, m = 8 и seed = 1. Количество элементов в последовательности – 10.

Табличный способ

Псевдослучайные числа оформляются в виде таблицы и помещаются во внешнюю или оперативную память ЭВМ в предварительно сформированном файле (массиве чисел).

Данный способ получения псевдослучайных чисел при моделировании систем на ЭВМ обычно рационально использовать при сравнительно небольшом объеме таблицы и, соответственно, файла чисел, когда для хранения можно применять оперативную память. Хранение файла во внешней памяти при частном обращении в процессе статистического моделирования нерационально, так как вызывает увеличение затрат машинного времени при моделировании системы из-за необходимости обращения к внешнему накопителю. Возможны промежуточные способы

организации файла, когда он переписывается в оперативную память периодически по частям. Это уменьшает время на обращение к внешней памяти, но сокращает объем оперативной памяти, который можно использовать для моделирования процесса функционирования системы.

Критерий оценки случайности последовательности

В качестве критерия оценки случайности последовательности был выбран критерий сериальной корреляции:

$C = \frac{n(x_0x_1 + x_1x_2 + \cdots + x_{n-2}x_{n-1} + x_{n-1}x_0) - (x_0 + x_1 + x_2 + \cdots + x_{n-1})^2}{n(x_0^2 + x_1^2 + x_2^2 + \cdots + x_{n-1}^2) - (x_0 + x_1 + x_2 + \cdots + x_{n-1})^2}$, где n – число элементов последовательности.

Данный коэффициент является мерой зависимости между x_{i+1} от x_i.

Коэффициент сериальной корреляции лежит в пределах от -1 до 1. Когда он равен 0 или очень мал, это означает, что величины x_{i+1} и x_i линейно независимы друг от друга. Если значение коэффициента равно ±1, то между величинами x_{i+1} и x_i полная линейная зависимость.

Последовательность обладает более высокой случайностью, если коэффициент сериальной корреляции мал.

Пример работы приложения

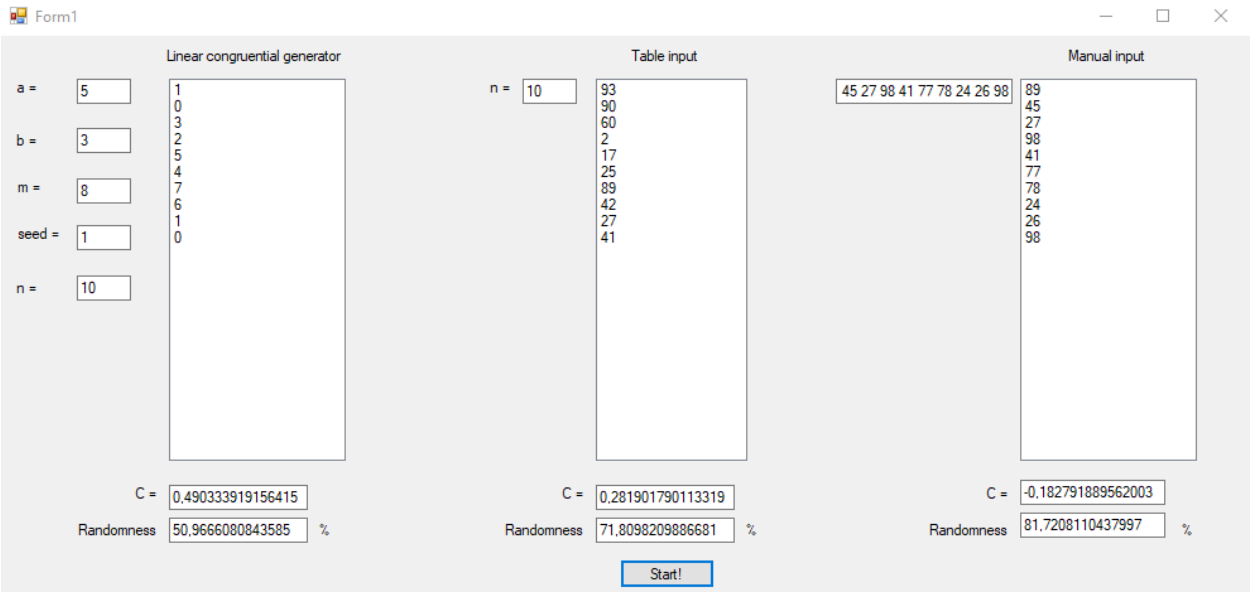

Рисунок 1 – Пример работы приложения

Вывод

Наилучшей случайностью обладает последовательность псевдослучайных чисел, полученная ручным способом. Наихудшей – полученная алгоритмическим способом.

Листинг программного кода приложения

```
using System;
using System.IO;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace ГПСЧ
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            textBox2.Text = "5";
            textBox3.Text = "3";
            textBox4.Text = "8";
            textBox5.Text = "10";
            textBox6.Text = "1";
        }

        public int[] LCG()
        {
            int a = Convert.ToInt32(textBox2.Text);
            int b = Convert.ToInt32(textBox3.Text);
            int m = Convert.ToInt32(textBox4.Text);
            int n = Convert.ToInt32(textBox5.Text);
            int seed = Convert.ToInt32(textBox6.Text);

            int[] x = new int[n];

            x[0] = seed;

            for (int i = 1; i < n; i++)
            {
                x[i] = (a * x[i - 1] + b) % m;
            }

            for (int i = 0; i < n; i++)
            {
                listBox1.Items.Add(x[i].ToString());
            }

            double C = 0;

            int num1 = 0, num2 = 0, denom = 0;

            num2 = x[0];
            denom = n * x[0] * x[0];

            for (int i = 0; i < n - 1; i++)
            {
                num1 = num1 + n * x[i] * x[i + 1];
                num2 = num2 + x[i + 1];
                denom = denom + n * x[i + 1] * x[i + 1];
```

```
            }

            C = (double)(num1 - num2 * num2 + n * x[n - 1] * x[0]) / (double)(denom -
num2 * num2);

            textBox13.Text = (C.ToString());
            textBox1.Text = (((1 - Math.Abs(C))*100).ToString());

            return x;
        }

        public void tableInput()
        {
            string path = "Random digits table.txt";

            string[] sNums = File.ReadAllText(path).Split(new char[] { ' ', '\n' });

            int[] x = new int[sNums.Length - 1];

            for (int i = 0; i < x.Length; i++)
            {
                x[i] = Convert.ToInt32(sNums[i]);
            }

           int n = Convert.ToInt32(textBox9.Text);

            for (int i = 0; i < n; i++)
            {
                listBox3.Items.Add(x[i].ToString());
            }

            double C = 0;

            int num1 = 0, num2 = 0, denom = 0;

            num2 = x[0];
            denom = n * x[0] * x[0];

            for (int i = 0; i < n - 1; i++)
            {
                num1 = num1 + n * x[i] * x[i + 1];
                num2 = num2 + x[i + 1];
                denom = denom + n * x[i + 1] * x[i + 1];
            }

            C = (double)(num1 - num2 * num2 + n * x[n - 1] * x[0]) / (double)(denom -
num2 * num2);

            textBox11.Text = (C.ToString());
            textBox10.Text = (((1 - Math.Abs(C)) * 100).ToString());
        }

        public int[] manualInput()
        {
            string[] sNums = textBox8.Text.Split(' ');

            int[] x = new int[sNums.Length];

            for (int i = 0; i < x.Length; i++)
            {
                x[i] = Convert.ToInt32(sNums[i]);
            }

            for (int i = 0; i < x.Length; i++)
            {
```

```
                listBox2.Items.Add(x[i].ToString());
            }

            double C = 0;

            int num1 = 0, num2 = 0, denom = 0;

            num2 = x[0];
            denom = x.Length * x[0] * x[0];

            for (int i = 0; i < x.Length - 1; i++)
            {
                num1 = num1 + x.Length * x[i] * x[i + 1];
                num2 = num2 + x[i + 1];
                denom = denom + x.Length * x[i + 1] * x[i + 1];
            }

            C = (double)(num1 - num2 * num2 + x.Length * x[x.Length - 1] * x[0]) /
(double)(denom - num2 * num2);

            textBox12.Text = (C.ToString());
            textBox7.Text = (((1 - Math.Abs(C)) * 100).ToString());

            return x;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            listBox1.Items.Clear();
            listBox2.Items.Clear();
            listBox3.Items.Clear();

            LCG();
            tableInput();
            manualInput();
        }
    }
}
```