Міністерство освіти і науки України

Національний авіаційний університет

Факультет кібербезпеки, комп’ютерної та програмної інженерії

Кафедра інженерії програмного забезпечення

**Лабораторна робота № 1.5**

«Дослідження точності моделювання методом Монте-Карло»

**Варіант: 14**

**Виконано студентом групи:**

ПІ-322, Царук С. О.

**Перевірено:**

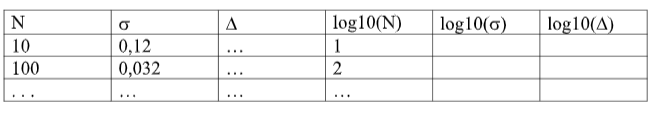
Волкогон В.О.

Київ – 2021

**Мета лабораторної роботи –** дослідити залежність точності моделювання від кількості випробувань.

**Завдання**

1. Обрахувати аналітичне значення інтегралу функції (згідно заданому варіанту) – IA .
2. Розробити та налагодити програму обрахування інтеграла методом Монте-Карло. Для генерація випадкових чисел застосувати метод NextDouble() класу Random. При необхідності зробити перетворення діапазону з [0..1] до [a..b].
3. Дослідити, як залежить точність обрахування інтегралу від кількості випробувань N =10, 100, …, 1 000 000. Кожне обрахування для заданого N провести K=10 разів, зайти оцінки математичного очікування M(I), дисперсії і середньоквадратичного відхилення D(I), , похибки
4. За допомогою Excel обробити результати досліджень. Побудувати таблицю з 6-ю стовпчиками.



**Варіант**

|  |  |  |
| --- | --- | --- |
| **№** | **Функція** | **Інтервал інтегрування** |
| 14 |  |  |

**Хід роботи**

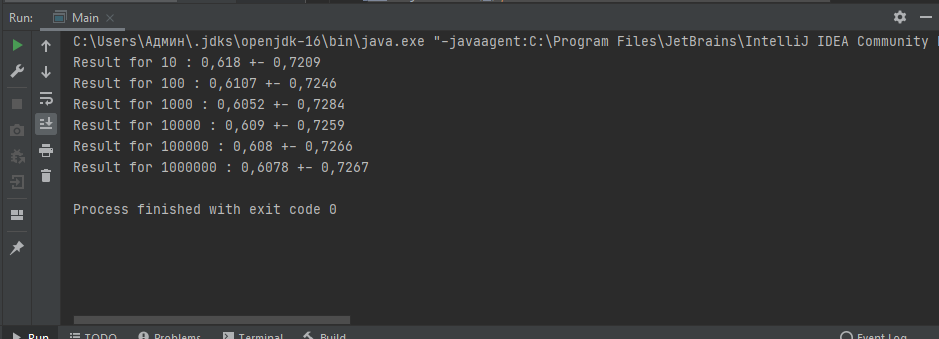
1. **Аналітична частина.**

Визначимо аналітичне значення інтегралу.

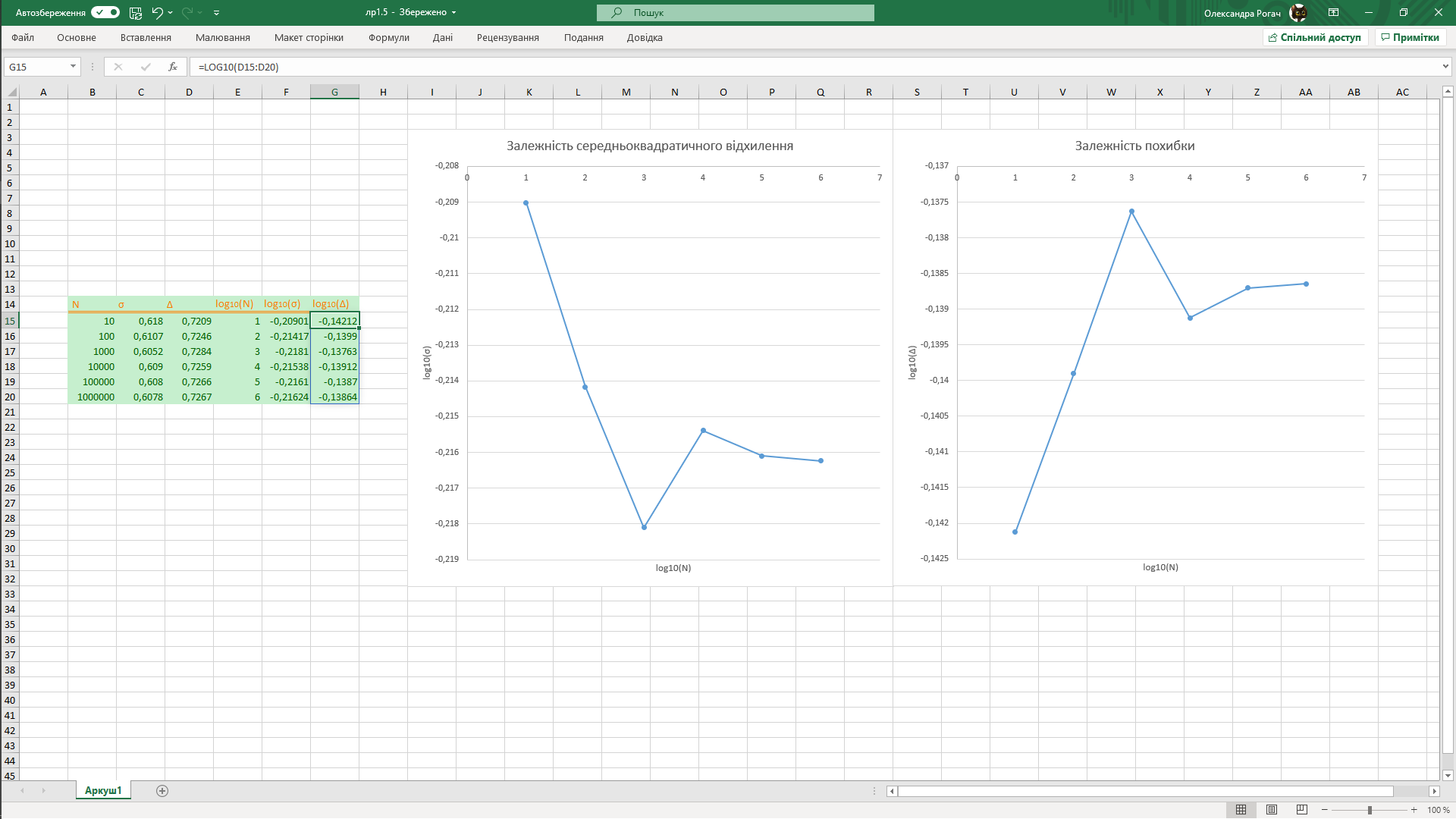
1. **Програмний код.**

package com.company;  
  
import java.text.DecimalFormat;  
import java.util.Random;  
  
public class Main {  
  
 public static void main(String[] args) {  
 for(int i = 10; i < 1000001;) //проведення обчислення з кількістю спроб від 100 до 1000000  
 {  
 *Integral*(i);  
 i \*= 10;  
 }  
 }  
  
 public static void Integral(int n) {  
 int k = 10;  
 double[] i\_arr = new double[k], i\_square\_arr = new double[k];  
 double x, a = 0, b = Math.*PI* / 2;  
 double i, i\_square, res, m, d, sigma, delta;  
  
 for (int j1 = 0; j1 < k; j1++) //цикл для повторення випробування k разів  
 {  
 i = 0;  
 i\_square = 0;  
  
 for (int j2 = 0; j2 < n; ++j2) //підрахунок інтегралів  
 {  
 x = *rand*(a, b);  
 res = *function*(x);  
 i += res;  
 i\_square += Math.*pow*(res, 2);  
 }  
  
 i\_arr[j1] = i/n; //масив з 10 інтегралів  
 i\_square\_arr[j1] = i\_square/n; //масив з 10 квадратів інтегралів  
 }  
  
 m = *sum*(i\_arr)/k; //математичне очікування  
 d = (*sum*(i\_square\_arr) - 2\*m\**sum*(i\_arr) + k\*m)/(k-1); //дисперсія  
 sigma = Math.*sqrt*(Math.*abs*(d)); //середньоквадратичне відхилення  
 delta = Math.*abs*(2 - m); //похибка  
  
 System.*out*.println("Result for " + n + " : " + *format*(sigma) + " +- " + *format*(delta));  
 }  
  
 public static double rand(double min, double max) //рандомізація значення х  
 {  
 Random random = new Random();  
 return min + random.nextDouble() \* (max - min);  
 }  
  
 public static double function(double x) //функція  
 {  
 return Math.*cos*(x) + Math.*sin*(x);  
 }  
  
 public static String format(double x) //округлення результату  
 {  
 DecimalFormat decimalFormat = new DecimalFormat( "#.####" );  
 return decimalFormat.format(x);  
 }  
  
 public static double sum(double[] arr) //розрахунок суми масивів  
 {  
 double res = 0;  
  
 for (double v : arr) {  
 res += v;  
 }  
  
 return res;  
 }  
}

1. **Результати виконання програми.**



1. **Графік.**



**Висновки**

Під час виконання лабораторної роботи я ознайомився з дослідженням залежності точності моделювання від кількості випробувань.