МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «Крымский федеральный университет имени В. И. Вернадского»
Физико-технический институт
Кафедра компьютерной инженерии и моделирования

Лабораторная работа №1
по курсу «Алгоритмы и методы вычислений»
на тему: «Численное интегрирование»

Выполнил:
студент 2 курса
группы ИВТ-202(1)
Шор К.А

Проверила:
старший преподаватель
кафедры компьютерной
инженерии и моделирования
Горская И.Ю.

Симферополь, 2022

Лабораторная работа № 1

Тема: Численное интегрирование

Цель работы:

1. Изучить и научиться использовать на практике наиболее эффективные алгоритмы численного интегрирования.
2. Изучить приемы контроля точности вычисляемых интегралов и оптимизации быстродействия используемых алгоритмов.
3. Написать программу, реализующую два метода численного интегрирования в следующих комбинациях: метод Симпсона с контролем погрешности по формуле Рунге; метод Гаусса - Кронрода, или Чебышева, или Монте-Карло.

Перед выполнением лабораторной работы:

1. Изучить презентацию лектора курса: «Введение в вычислительную математику и физику», «Численное интегрирование (Численные квадратуры)», материалы в электронном виде доступны на One Drive.

2. Прочитать соответствующие разделы в книгах: Каханер Д., Моулер К., Нэш С. Численные методы и математическое обеспечение: пер. с англ.- М.: Мир, 1998. - 575 с., Мухамадиев. Конспект лекций по вычислительной математике, 2007., Х.Гулд, Я.Тобочник. Компьютерное моделирование в физике, в 2-ух томах.-М.:Мир.-1990г. (метод Моне-Карло).

В соответствии с индивидуальным заданием, решены следующие задачи:

1. Изучены и закреплены на практике способы использования наиболее эффективных алгоритмов численного интегрирования.

2. Изучены приемы контроля точности вычисляемых интегралов и оптимизации быстродействия используемых алгоритмов.

3. Реализована программа, реализующая два метода численного интегрирования: методом Симпсона, с контролем погрешности по формуле Рунге, а также методом Гаусса-Крондора

Ход работы

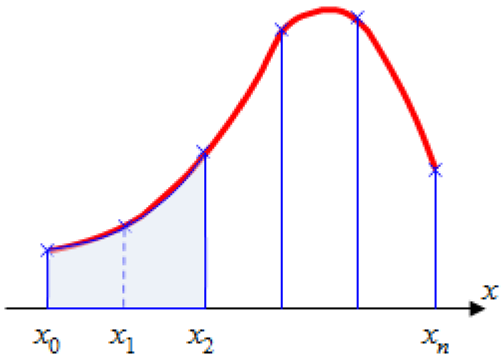
Вариант 12

12	$x^4 \ln\left(x + \sqrt{x^2 - 0{,}36}\right)$	1,25	2,45	10^{-4}

Суть метода заключается в приближении подынтегральной функции на отрезке *[a, b]* интерполяционным многочленом второй степени $p_2(x)$, то есть приближение графика функции на отрезке параболой.

$$J \approx \frac{b-a}{3n}\left[y_0 + 4(y_1 + y_3 + \ldots + y_{n-1}) + 2(y_2 + y_4 + \ldots + y_{n-2}) + y_n\right],$$

коэффициенты меняются по схеме: 1 - 4 - 2 - 4 ... 4 - 1, число разбиений отрезка (a,b) должно быть четным, а число точек нечетным.

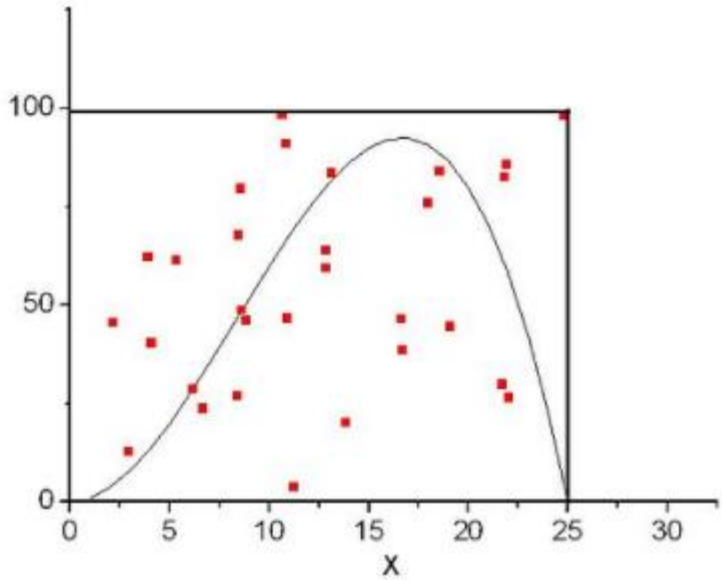


Для практической оценки погрешности дискретизации, которая возникает при применении численного интегрирования, воспользуемся правилом Рунге, заключающееся в последовательном увеличении (например, удвоении) числа узловых точек n и соответствующем уменьшении шага дискретизации h. Оценка по правилу Рунге имеет вид:

$$J - J_{h/2} \approx \frac{J_{h/2} - J_h}{2^k - 1},$$

Метод Монте-Карло - общее название группы методов, основанных на получении большого числа реализаций стохастического процесса, так, что вероятностные характеристики совпадают со свойствами модели.

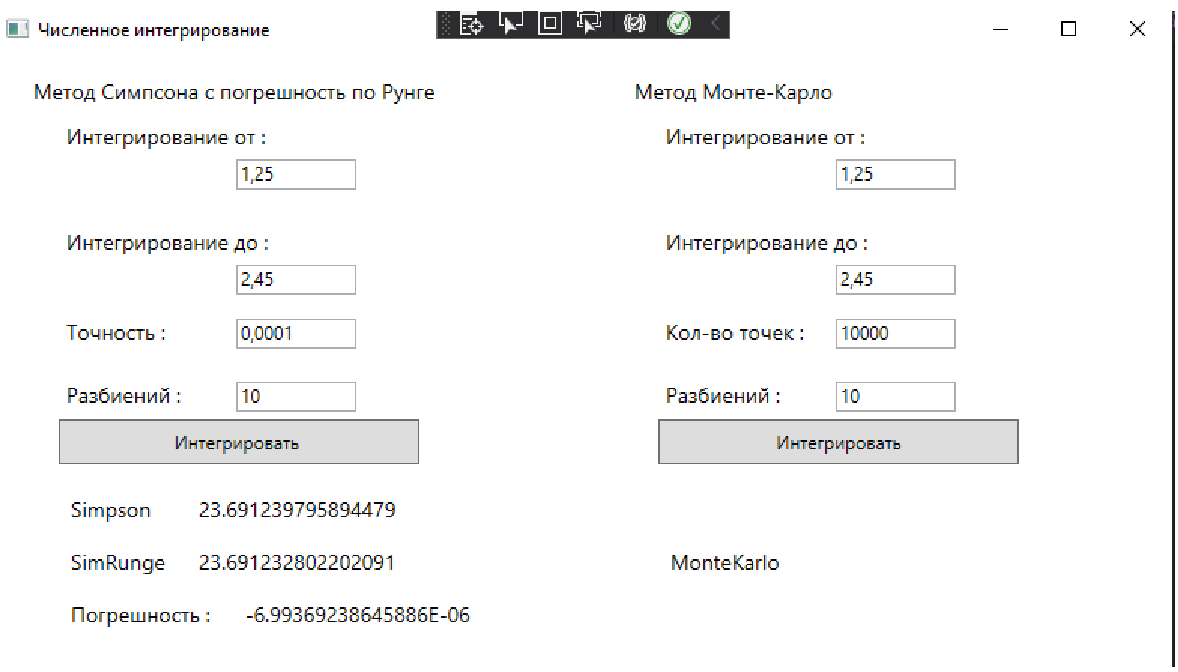
Суть метода заключается в том, что по X мы ограничиваемся пределами интегрирования a и b. По Y ищем максимальное значение и проводим через него касательную параллельную оси X, она и будет ограничением. Потом рандомно бросаем в этот прямоугольник точки. Считаем сколько точек попало в график, а сколько нет. Находим их отношение и умножаем на площадь ограничивающей фигуры — это и будет значением интеграла.



Листинг-1 Функция численного интегрирования методом Симпсона

```
        static double Simpson(double a, double b, double N)
        {
            double f(double x) =>/* Math.Sin(x)*/Math.Abs((x * x * x * x) *
Math.Log(x + Math.Sqrt((x * x) - 0.36)));

            double h = (b - a) / N;
            double sum = f(a) + f(b);
            int k;
            for (int i = 1; i <= N - 1; i++)
            {
                k = 2 + 2 * (i % 2); // если нечетное -> 4, четное -> 2
                sum += k * f(a + i * h);
            }
            sum *= h / 3;

            return sum;
        }
```

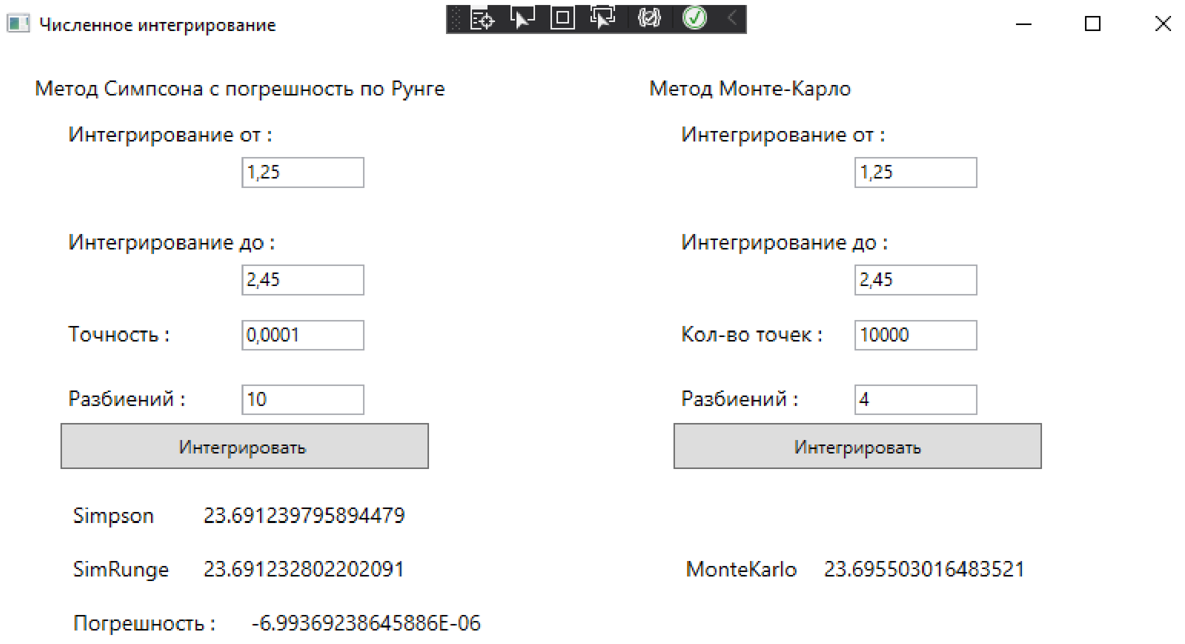
Работа алгоритма начинается с объявления функции и ключевых переменных, после чего, в цикле считается сумма по формуле, в зависимости от коэффициента, который стоит при разбиение. В конце всё подставляется в конечную формулу и возвращается.



Листинг-2 Функция численного интегрирования методом Монте-Карло

```
        static double MonteKarlo(double min, double max, double N)
        {
            double f(double x) =>/* Math.Sin(x)*/Math.Abs((x * x * x * x) *
Math.Log(x + Math.Sqrt((x * x) - 0.36)));

            Random random = new Random();
            double Y, F;
            double counter = 0;

            double X = random.NextDouble() * (max * min) + min;
            double max_heigth = f(X)*2-max*min;

            for(int i = 0; i < N; i++)
            {
                X = random.NextDouble() * (max - min) + min;
                Y = random.NextDouble() * max_heigth;
                F = f(X);

                if (F > Y)
                {
                    counter++;
                }
            }
            double area = max_heigth * (max - min);
            double hits = counter / N;

            return area * hits;
        }
```

Работа алгоритма начинается с объявления функции и ключевых переменных, после чего, задаются границы фигуры, в которую будут забрасываться точки. Затем в цикле проверяется сколько точек попало. В конце по формуле рассчитывается искомое значение.



Преподавателю представлена программа, реализующая два метода численного интегрирования в электронной форме, продемонстрирована их работоспособность, разъяснены детали программного кода