Университет ИТМО

Факультет ПИиКТ

**Вычислительная математика**

Лабораторная работа 2

Вариант – 2 (Метод трапеций)

Выполнил: Давтян Давид Арменович

Группа: P3201

Преподаватель: Перл Ольга Вячеславовна

Санкт-Петербург, 2020 г.

**Описание используемого метода**

Путь есть функция , заданная на отрезке . Разьём этот отрезок на отрезков , где и , .

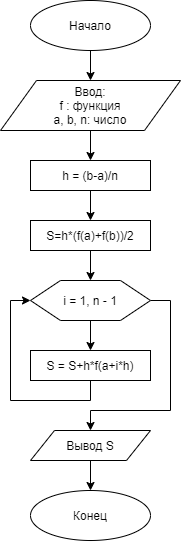
На каждом из этих отрезков выберем произвольную точку и для каждого отрезка посчитаем , тогда интегральной суммой будет сумма всех : .

Чтобы найти определённый интеграл, нужно найти предел суммы всех при .

Суть метода трапеций в том, что подынтегральную функцию на каждом отрезке заменяют интерполяционным многочленом первой степени: .

Это значит, что график функции будет представляться в виде ломаной, соединяющей точки . Если принять, что , тогда площадь фигуры можно посчитать так:

**Блок-схема**

****

**Листинг численного метода**

package ru**.**david**.**compmath2**.**math**.**integration**;**

**import** ru**.**david**.**compmath2**.**math**.**expression**.**Expression**;**

**import** ru**.**david**.**compmath2**.**math**.**expression**.**VariableExpression**;**

public class Integrator **{**

private static final double VERY\_SMALL\_NUMBER **=** 0.0000001**;**

private static final int SPLITS\_LIMIT **=** 100\_000\_000**;**

public static IntegrationResult solve**(**

Expression expression**,**

VariableExpression variable**,**

IntegrationLimits limits**,**

double accuracy

**)** **{**

int splits **=** 4**;**

double previousResult**;**

double result **=** solve**(**expression**,** variable**,** limits**,** 2**);**

double error**;**

**do** **{**

previousResult **=** result**;**

result **=** solve**(**expression**,** variable**,** limits**,** splits**);**

error **=** Math**.**abs**(**previousResult **-** result**)/**3**;**

splits **\*=** 2**;**

**}** **while** **(**

Math**.**abs**(**error**)** **>** Math**.**abs**(**accuracy**)** **&&**

Double**.**isFinite**(**result**)** **&&**

splits **<** SPLITS\_LIMIT

**);**

**if** **(**Double**.**isFinite**(**result**)** **&&** splits **<** SPLITS\_LIMIT**)**

**return** **new** IntegrationResult**(**result**,** error**,** splits**);**

**return** **new** IntegrationResult**(**IntegrationResult**.**Status**.**GAP**);**

**}**

private static double solve**(**

Expression expression**,**

VariableExpression variable**,**

IntegrationLimits limits**,**

int splits**)** **{**

double deltaXSize **=**

**(**limits**.**getHighLimit**()** **-** limits**.**getLowLimit**())/**splits**;**

double result **=** 0**;**

**for** **(**int i **=** 0**;** i **<** splits**;** i**++)** **{**

variable**.**setValue**(**

limits**.**getLowLimit**()** **+** i**\***deltaXSize

**);**

double leftY **=** expression**.**value**();**

**if** **(!**Double**.**isFinite**(**leftY**))** **{**

**if** **(**i **==** 0**)**

**return** Double**.**POSITIVE\_INFINITY**;**

variable**.**setValue**(**

limits**.**getLowLimit**()** **+**

i**\***deltaXSize **+**

VERY\_SMALL\_NUMBER

**);**

leftY **=** expression**.**value**();**

**}**

variable**.**setValue**(**limits**.**getLowLimit**()** **+** **(**i**+**1**)\***deltaXSize**);**

double rightY **=** expression**.**value**();**

**if** **(!**Double**.**isFinite**(**rightY**))** **{**

**if** **(**i **==** splits **-** 1**)**

**return** Double**.**POSITIVE\_INFINITY**;**

variable**.**setValue**(**

limits**.**getLowLimit**()** **+**

**(**i**+**1**)\***deltaXSize **-**

VERY\_SMALL\_NUMBER

**);**

rightY **=** expression**.**value**();**

**}**

double avgY **=** **(**leftY**+**rightY**)/**2**;**

result **+=** avgY**\***deltaXSize**;**

**}**

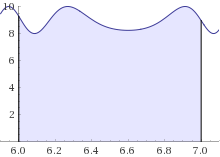
**return** result**;**

**}**

**}**

**Примеры**

**Вывод программы График**

****Введите любое выражение:

y = **sin(x + x^2\*cos(x)) + 9**

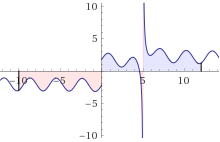
Пределы интегрирования: **6 7**

Точность: **0.00001**

Результат: **8.9755387813312720**

Погрешность: **0.0000057403610961**

Сделано разбиений: **256**

Введите любое выражение:

y = **sin(x\*2) + 2\*abs(x)/x + 1/(x-5)**

Пределы интегрирования: **-10 12**

Точность: **0.00001**

Не получилось посчитать интеграл.

Возможно, он не сходится или

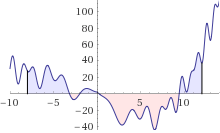
указана слишком высокая точность.

Введите любое выражение:

y = **-5\*abs(x) + 18\*sin(x^2/4)\*cosx + (x^2)/1.5 + (x^3)/20 - 6\*x + 2**

Пределы интегрирования: **-8 12**

Точность: **0.000000001**

****Результат: **-28.1638511862336900**

Погрешность: **0.0000000007469628**

Сделано разбиений: **2097152**

**Вывод**

В данной лабораторной работе я реализовал алгоритм вычисления интеграла с помощью метода трапеций и парсер математических выражений, позволяющий ввести любое выражение с переменной и посчитать его интеграл.

В методе трапеций подынтегральную функцию функцию на каждом заменяют интерполяционным многочленом первой степени. Интерполяция кусочно-линейная, поэтому график исходной функции представляется как ломаная, которая соединяет точки . Площадь всей фигуры состоит из суммы площадей всех трапеций. Погрешность метода трапеций выше, чем у метода средних прямоугольников, но ниже, чем у методов левых и правых прямоугольников. Это объясняется тем, что в методе средних прямоугольников используется значение в средней точке каждого отрезка, а в методе трапеций – полсумма двух крайних точек.

Метод прямоугольников в описанном выше виде неприменим в общем случае к функциям, значения которых мы знаем в конечном числе точек, так как, например, мы не всегда можем разбить отрезок интегрирования на подотрезки, серединами которых являются точки, в которых нам известно значение функции; в методе трапеций можно взять в качестве узлов интегрирования данные точки.

В методе Симпсона (парабол) разбиваем отрезок на четное число n равных частей с шагом, равным h. На каждом отрезке подынтегральную функцию заменяют интерполяционным многочленом второй степени. Тогда общая формула: . Точность метода Симпсона выше точности метода прямоугольников и трапеций.