**ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ**

Во многих случаях, когда функция задана аналитически, определенный интеграл вычисляется по формуле Ньютона – Лейбница, которая состоит в том, что определенный интеграл равен приращению первообразной на отрезке интегрирования:

Однако на практике не всегда можно вычислить интеграл с помощью этой формулы. Основными причинами являются:

1. Невозможность выразить первообразную функцию через элементарные функции.

2. Первообразная вычисляется через элементарную функцию, но выражение довольно сложное для вычисления.

3. Функция задана таблицей.

В случае, когда нельзя применить формулу Ньютона–Лейбница, обращаются к методам численного интегрирования.

Методы численного интегрирования основаны на аппроксимации подынтегральной функции некоторыми более простыми выражениями, т.е. подынтегральную функцию заменяют другой приближенной функцией так, чтобы, во-первых, приближенная функция была близка к и, во вторых, интеграл от нее легко вычислялся. Формулы численного интегрирования называются *квадратурами*.

Например, можно заменить подынтегральную функцию интерполяционным многочленом.

К методам численного интегрирования относят такие методы как: метод прямоугольников (левых, правых, средних), трапеций, парабол (метод Симпсона). Рассмотрим каждый из методов подробнее.

**Метод прямоугольников**

Метод прямоугольниковявляется одним из простейших методов. Данный метод использует замену определённого интеграла интегральной суммой. Формулу прямоугольников можно получить из геометрической интерпретации интеграла. Будем интерпретировать следующий интеграл:



как площадь криволинейной трапеции, ограниченной графиком функции , осью абсцисс и прямыми и*.*

Разобьем отрезок ] на равных частей длиной так, что При этом получим точки (рис.7.1).

;

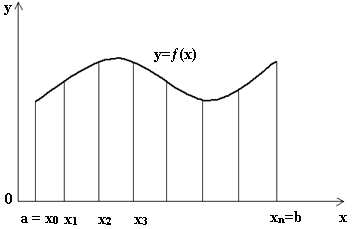


Рис. 7.1

Заменим приближенно площадь криволинейной трапеции площадью ступенчатой фигуры, как показано на рис.7.2.

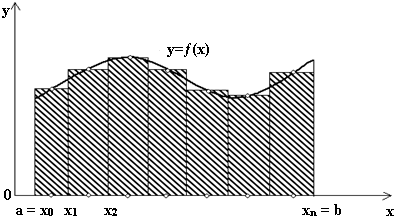
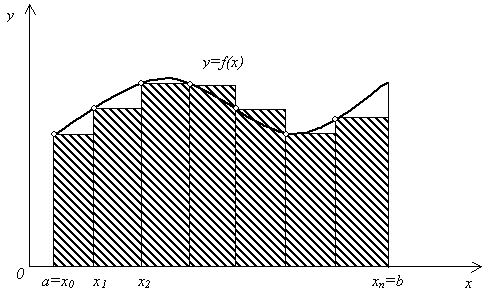


Рис. 7.2.

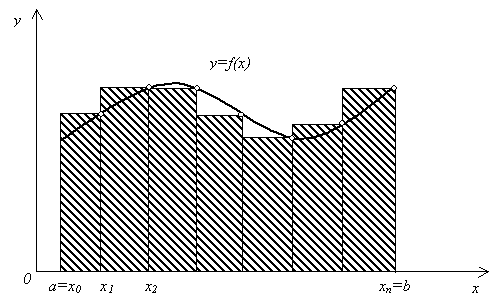
Эта фигура состоит из прямоугольников. Основание *n*-го прямоугольника образует отрезок  длины,  тогда получим квадратурные формулы.

Для левых прямоугольников с постоянным шагом:

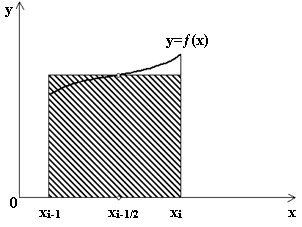




Для правых прямоугольников:



Более точной является формула прямоугольников, использующая функции в средних точках участков:





**Метод трапеций**

Метод трапеций использует линейную интерполяцию, т.е. график функции  представляется в виде ломаной, соединяющей точки  В этом случае площадь всей фигуры складывается из площадей элементарных трапеций (рис. 7.3):

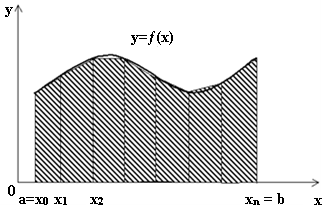


Рис. 7.3

Площадь каждой такой трапеции равна произведению полусуммы оснований на высоту. Складывая эти равенства, получим следующую формулу для вычислений:



Точность интегрирования зависит от степени многочлена, количества участков и расположение точек.

Во многих случаях формула средних прямоугольников дает лучшую точность, чем формула трапеций. Это на первый взгляд неожиданно, т.к. формула прямоугольников использует интерполяцию нулевого порядка, а формула трапеций – нелинейную. Здесь все дело в особом рассположении точек, которое повышает точность.

**Метод парабол (Симпсона)**

Метод парабол (Симпсона) подразумевает разбиение отрезка интегрирования , на четное число равных частей с шагом . На каждом отрезке подынтегральную функцию заменяют многочленом второй степени, т.е. уравнением квадратичной параболы.

На каждом участке строится парабола, находится площадь фигуры, ограниченной полиномом 2–й степени и графиком подынтегральной функции. Далее находится сумма этих площадей.

В результате получим следующую формулу для вычислений:

**Лабораторная работа 5 (практика)**

Вычисление определенных интегралов с помощью

численных методов

**Цель работы**: изучить методы численного интегрирования, алгоритм каждого метода, формулы для вычисления, написать программу на языке программирования для реализации данных методов.

К методам численного интегрирования относят метод прямоугольников (левых, правых, средних), метод трапеций и метод парабол (метод Симпсона).

**Алгоритм метода прямоугольников:**

– соответственно нижний и верхний пределы интегрирования;

количество участков разбиения;

– шаг;

подынтегральная функция.

Ввод

Fo r to step

Next

Печать

**Алгоритм метода трапеций:**

–соответственно нижний и верхний пределы интегрирования;

количество участков разбиения;

– шаг;

подынтегральная функция.

Ввод

For to step

Next

Печать

**Алгоритм метода Симпсона:**

– соответственно нижний и верхний пределы интегрирования;

количество участков разбиения (должно быть четным);

– шаг;

подынтегральная функция.

Ввод

For to step

Next

Печать

ВАРИАНТЫ ЗАДАНИЙ

|  |  |  |
| --- | --- | --- |
| № варианта | Интеграл | Шаг интегрирования |
| 1 | 2 | 3 |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 1 | 2 | 3 |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |

Входные данные, после запуска программы – значения границ .

Выходные данные – вычисленное значение интеграла по каждому из четырех методов.

**Реализация алгоритмов на языке C#**

При вычислении *определенных интегралов с помощью численных методов* в основной части программы определяются верхний и нижний пределы интегрирования, количество интегралов разбиения, а для вычисления подынтегральной функции используется отдельный метод:

Static double func(double x)

{

return (Math. Sqrt(1.5 \* x + 0.6)) / (1.6 + Math. Sqrt(0.8 \* Math. Pow(x, 2) + 2));

}

Поскольку реализация алгоритма метода прямоугольников (левых и правых) и алгоритма метода трапеций одинакова, то вычисление можно организовать в одном цикле, при этом метод средних прямоугольников требует дополнительных условий.

int n = 50;

double H = (b - a) / n;

for (double x = a; x < b; x += H)

{

result1 += func(x) \* H; //левых прямоугольников

result2 += func(x + H) \* H; //правых прямоугольников

result3 += H \* (func(x) + func(x + H)) / 2; //трапеций

}

for (double i = a; i< b - H; i = i + H)

{

result4 = result4 + (func(i) + func(i + H)) / 2;

//средних прямоугольников

}

Формула для вычисления функции по методу Симпсона достаточно громоздка, поэтому для удобства следует разбить ее на части, каждую из которых можно вычислить с помощью цикла с параметром FOR. Реализация данного метода представлена ниже:

**//**метод Симпсона

Static double Simpson (double a, double b, int n)

{

Int i;

double S = 0;

double[] x = new double[n + 1];

double h = (b - a) / n;

**//**значение функции для равноотстоящих точек

for (i = 0; i < n; i++)

{

x[i] = a + h \* i;

}

**//**для подсчета h/3 \* (y0+yn)

S = h \* (func(x[0])+func(x[n])) / 3;

**//**для подсчета h/3 \* 4(y1+y3+...+y(n-1))

for (i = 1; i< (n - 1); i = i + 2)

{

S = S + h \* 4 \* func(x[i]) / 3;

}

**//**для подсчета h/3 \* 2(y2+y4+...+y(n-2))

for (i = 2; i< (n - 2); i = i + 2)

{

S = S + h \* 2 \* func(x[i]) / 3;

}

return S;

}

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Особенности численного интегрирования.

2. Метод левых, правых и средних прямоугольников.

3. Метод трапеций.

4. Метод парабол.

5. Число участков разбиения и шаг интегрирования.

6. Способ повышения точности при численном интегрировании.