

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ»

Институт вычислительной математики и информационных технологий
Кафедра прикладной математики и искусственного интеллекта

ОТЧЕТ

Исследование приближения функций при помощи квадратурных формул

Выполнил:
студент группы 09-222 Шпак В.С.
Проверил:
ассистент Глазырина О.В.

Казань, 2024 год

Оглавление

ПОСТАНОВКА ЗАДАЧИ	3
ХОД РАБОТЫ	4
ВЫВОДЫ ПО РАБОТЕ	8
ЛИСТИНГ	9

ПОСТАНОВКА ЗАДАЧИ

Одна из специализированных функций математической физики – интегральный синус, определяется следующим образом $\text{Si}(x) = \int_0^x \frac{\sin t}{t} dx$

Цель задания – вычислить интеграл с помощью квадратурных формул.

1. Левых прямоугольников
2. Правых прямоугольников
3. Трапеции
4. Симпсона
5. Гаусса
6. Центральных прямоугольников

где $h = (x_{i+1} - x_i)$, $f(x) = \frac{\sin(x)}{x}$;

ХОД РАБОТЫ

Выберем одиннадцать равноотстоящих друг от друга точек на отрезке $[a, b]$ с шагом h . Для каждой точки найдём значение $Si(x)$. Составим таблицу из полученных результатов:

X	Si(x)
0	0
0.4	0.396461
0.8	0.772096
1.2	1.10805
1.6	1.38918
2	1.60541
2.4	1.75249
2.8	1.8321
3.2	1.8514
3.6	1.82195
4	1.7582

Для каждой точки будем вычислять значение до тех пор пока не выполнится условие $|J_N(x) - J_{2N}(x)| < \varepsilon$. Далее находим погрешность как модуль разности полученного данным методом значения и найденного через разложение в ряд Тейлора. Составляем таблицу, где $J_0(x)$ - найденное значение интеграла, $J_N(x)$ - настоящее значение интеграла, N - количество разбиений отрезка.

Левых прямоугольников: $J_N(x) = \sum_{i=0}^{n-1} h f(x_i)$

x_i	$J_0(x)$	$J_N(x)$	$ J_0(x) - J_N(x) $	N
0.4	0.396461	0.396467	5.16524e-06	1024
0.8	0.772096	0.772136	4.03406e-05	1024
1.2	1.10805	1.10818	0.000130801	1024
1.6	1.38918	1.38947	0.000293099	1024
2	1.60541	1.60595	0.00053243	1024
2.4	1.75249	1.75333	0.000841865	1024
2.8	1.8321	1.8333	0.00120339	1024
3.2	1.8514	1.85299	0.00159075	1024
3.6	1.82195	1.82392	0.00197365	1024
4	1.7582	1.76053	0.00232251	1024

Правых прямоугольников: $J_N(x) = \sum_{i=1}^n h f(x_i)$

x_i	$J_0(x)$	$J_N(x)$	$ J_0(x) - J_N(x) $	N
0.4	0.396461	0.396456	5.16841e-06	1024
0.8	0.772096	0.772055	4.03664e-05	1024
1.2	1.10805	1.10792	0.00013088	1024
1.6	1.38918	1.38889	0.000293255	1024
2	1.60541	1.60488	0.00053271	1024
2.4	1.75249	1.75164	0.000842253	1024
2.8	1.8321	1.83089	0.00120385	1024
3.2	1.8514	1.84981	0.00159125	1024
3.6	1.82195	1.81997	0.00197412	1024
4	1.7582	1.75588	0.0023228	1024

Трапеции: $J_N(x) = \sum_{i=0}^{n-1} h \frac{f(x_i) + f(x_{i+1})}{2}$

x_i	$J_0(x)$	$J_N(x)$	$ J_0(x) - J_N(x) $	N
0.4	0.396461	0.396461	1.06701e-07	128
0.8	0.772096	0.772096	2.03634e-07	256
1.2	1.10805	1.10805	1.57927e-07	512
1.6	1.38918	1.38918	3.27103e-07	512
2	1.60541	1.60541	1.40058e-07	1024
2.4	1.75249	1.75249	1.93854e-07	1024
2.8	1.8321	1.8321	2.29813e-07	1024
3.2	1.8514	1.8514	2.50917e-07	1024
3.6	1.82195	1.82195	2.37024e-07	1024
4	1.7582	1.7582	1.43659e-07	1024

Весьма интересно выводится формула Симпсона

ВСТАВИТЬ ЗДЕСЬ ВЫВОД ФОРМУЛЫ СИМПСОНА

Симпсона: $J_N(x) = \sum_{i=0}^{n-1} \frac{h}{6} \left[f(x_i) + 4f\left(\frac{x_i + x_{i+1}}{2}\right) + f(x_{i+1}) \right]$

x_i	$J_0(x)$	$J_N(x)$	$ J_0(x) - J_N(x) $	N
0.4	0.396461	0.396461	2.80595e-09	4
0.8	0.772096	0.772096	4.94881e-09	8
1.2	1.10805	1.10805	3.54972e-08	8
1.6	1.38918	1.38918	1.35524e-08	16
2	1.60541	1.60541	1.84437e-08	16
2.4	1.75249	1.75249	3.86081e-08	16
2.8	1.8321	1.8321	6.33216e-08	16
3.2	1.8514	1.8514	6.24795e-08	16
3.6	1.82195	1.82195	2.6337e-08	16
4	1.7582	1.7582	2.6013e-08	16

Гаусса: $J_N(x) = \sum_{i=0}^{n-1} \frac{h}{2} \left[f\left(x_i + \frac{h}{2} \left(1 - \frac{1}{\sqrt{3}}\right)\right) + f\left(x_i + \frac{h}{2} \left(1 + \frac{1}{\sqrt{3}}\right)\right) \right]$

x_i	$J_0(x)$	$J_N(x)$	$ J_0(x) - J_N(x) $	N
0.4	0.396461	0.396461	1.73704e-09	4
0.8	0.772096	0.772096	5.50997e-08	4
1.2	1.10805	1.10805	2.34466e-08	8
1.6	1.38918	1.38918	1.40096e-10	16
2	1.60541	1.60541	1.50446e-08	16
2.4	1.75249	1.75249	2.49499e-08	16
2.8	1.8321	1.8321	3.14221e-08	16
3.2	1.8514	1.8514	4.44494e-08	16
3.6	1.82195	1.82195	4.36041e-08	16
4	1.7582	1.7582	2.39866e-08	16

Центральных прямоугольников: $J_N(x) = \sum_{i=0}^{n-1} h f\left(\frac{x_i + x_{i+1}}{2}\right)$

x_i	$J_0(x)$	$J_N(x)$	$ J_0(x) - J_N(x) $	N
0.8	0.772096	0.772096	1.01524e-07	256
1.2	1.10805	1.10805	3.16249e-07	256
1.6	1.38918	1.38918	1.71809e-07	512
2	1.60541	1.60541	2.75169e-07	512
2.4	1.75249	1.75249	9.76376e-08	1024
2.8	1.8321	1.8321	1.24621e-07	1024
3.2	1.8514	1.8514	1.22943e-07	1024
3.6	1.82195	1.82195	9.50707e-08	1024
4	1.7582	1.7582	2.99268e-07	512

ВЫВОДЫ ПО РАБОТЕ

:

Из представленных 6 методов самыми эффективными оказались методы Гаусса и Симпсона. Для метода Гаусса потребовалось меньше итераций, следовательно, он является самым результативным.

ЛИСТИНГ

```
#include <iostream>
#include <Math.h>
#include <string>
#include <vector>

static double step = 0.4;
static double Limit(double x)
{
    if (x != 0) return (sin(x) / x);
    return 1; //первый замечательный предел
}

static double Tabulate(double x)
{
    double a = x;
    double res = x;
    double q;
    int n = 0;
    do
    {
        q = (-1) * x * x * (2 * n + 1) / ((2 * n + 2) * (2 * n + 3) * (2 * n + 3));
        a *= q;
        res += a;
        n++;
    } while (abs(a) > 0.000001);
    return res;
}

static std::vector<double> Tabulate(std::vector<double> x)
{
    std::vector<double> result;
    for (int i = 0; i < x.size(); i++)
    {
        result.push_back(Tabulate(x[i]));
    }
    return result;
}

static double LeftRectangleMethod(int N, double x0)
{
    double h = x0 / N;
    double result = 0;
    double x = 0;
    for (int i = 0; i < N; i++)
    {
        result += h * Limit(x);
        x += h;
    }
    return result;
}
```

```

}
staticdoubleRightRectangleMethod(intN, doublex0)
{
    double h = x0 / N;
    double sum = 0;
    double x = h;
    for (int i = 0; i < N; i++)
    {
        sum += h * Limit(x);
        x += h;
    }
    return sum;
}
staticdoubleCentralRectanglesMethod(intN, doublex0)
{
    double h = x0 / N;
    double sum = 0;
    double x = h / 2;
    for (int i = 0; i < N; i++)
    {
        sum += h * Limit(x);
        x += h;
    }
    return sum;
}
staticdoubleSimpsonmethod(intN, doublex0)
{
    double h = x0 / N;
    double sum = 0;
    double x = 0;
    for (int i = 0; i < N; i++)
    {
        sum += (Limit(x) + 4 * Limit(x + h / 2) + Limit(x + h)) * h / 6;
        x += h;
    }
    return sum;
}
staticdoubleTrapezoidmethod(intN, doublex0)
{
    double h = x0 / N;
    double result = 0;
    double x = 0;
    for (int i = 0; i < N; i++)
    {
        result += h * (Limit(x) + Limit(x + h)) / 2;
        x += h;
    }
    return result;
}
staticdoubleGaussmethod(intN, doublex0)
{
    double h = x0 / N;
    double ad1 = (1 - 1.0 / sqrt(3)) * h / 2;
    double ad2 = (1 + 1.0 / sqrt(3)) * h / 2;
    double sum = 0;
    double x = 0;
    for (int i = 0; i < N; i++)
    {
        sum += (Limit(x + ad1) + Limit(x + ad2)) * h / 2;
        x += h;
    }
}

```

```

    }
    return sum;
}
static void CalculateAndWrite(double x, double y, double (*IntegralVariant)(int, double))
{
    double lastJ = 0;
    double J = 0;
    int n = 1;
    do
    {
        if (n == 1024)
        {
            break;
        }
        n *= 2;
        lastJ = J;
        J = IntegralVariant(n, x);
    } while (abs(lastJ - J) > 0.000001);
    double error = abs(J - y);
    std::cout << | < x << "/t| < y << "/t| < J << "/t| < error << "/t| < n << "/n";
}

```