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 искусственого интеллекта

Направление подготовки: 01.03.04 – Прикладная математика

ОТЧЁТ

По дисциплине «Численные методы»
на тему:
«Вычисление интеграла с помощью квадратурных формул»

Выполнил:
студент группы 09-222
Фаррахова Л. Ф.
Проверил:
ассистент Глазырина О.В.

Казань, 2024 год

Содержание

1 Постановка задачи

Необходимо изучить и сравнить различные способы приближённого вычисления функции

$$Si(x) = \int_0^x \frac{\sin(t)}{t}\,dt \tag{1}$$

1. Протабулировать Si(x) на отрезке [a,b] с шагом h и точностью ε, основываясь на ряде Тейллора, предварительно вычислив его. Получив таким образом таблицу из 11 точек вида:

   x_0 x_1 x_2 ...
   f_0 f_1 f_2 ...

   $f_i = \mathrm{Si}(x_i), \quad x_i = a + ih, \quad i = 0, \ldots, n.$

2. Вычислить Si(x) при помощи пяти составных квадратурных формул при

   $$h = (x_{i+1} - x_i) = \frac{b-a}{n} = \frac{\sin(t)}{t}$$

   2.1. Формула правых прямоугольников:

   $$J_N(x) = \sum_{i=1}^{n} hg(x_i) \tag{2}$$

   2.2. Формула центральных прямоугольников:

   $$J_N(x) = \sum_{i=1}^{n} hg\left(\frac{x_i + x_{i+1}}{2}\right) \tag{3}$$

   2.3. Формула трапеции:

   $$J_N(x) = \sum_{i=1}^{n} h\frac{g(x_i) + g(x_{i+1})}{2} \tag{4}$$

   2.4. Формула Симпсона:

   $$J_N(x) = \sum_{i=1}^{n} \frac{h}{6}\left[g(x_i) + 4g\left(\frac{x_i + x_{i+1}}{2}\right) + g(x_{i+1})\right] \tag{5}$$

   2.5. Формула Гаусса:

   $$J_N(x) = \sum_{i=1}^{n} \frac{h}{2}\left[g\left(x_i + \frac{h}{2}\left(1 - \frac{1}{\sqrt{3}}\right)\right) + g\left(x_i + \frac{h}{2}\left(1 + \frac{1}{\sqrt{3}}\right)\right)\right] \tag{6}$$

Вычисления проводятся от начала интегрирования до каждой из 11 точек, увеличивая количество разбиений между точками в 2 раза до тех пор, пока погрешность больше ε.

2 Ход работы

Для того чтобы найти значение функции в точке, необходимо протабулировать искомый интеграл на отрезке [a, b] с шагом $h = 0.4$ и точностью ε.

Для этого найдём разбиение Si(x) в ряд Тейлора:

1) Воспользуемся стандартным разбиением функции $sin(x) = t - \frac{t^3}{3!} + \frac{t^5}{5!} - \frac{t^7}{7!} + \dots$ в ряд Тейлора, сделав в нем замену $x = t$ и разделив на t:

$$\frac{sin(t)}{t} = 1 - \frac{t^2}{3!} + \frac{t^4}{5!} - \frac{t^6}{7!} + \dots \tag{7}$$

2) Проинтегрируем полученное выражение на интеграле [0, x]:

$$\begin{aligned}\int_0^x \frac{\sin(t)}{t}\,dt = \int_0^x \left(1 - \frac{t^2}{3!} + \frac{t^4}{5!} - \frac{t^6}{7!} + \dots\right) dt = \\ = x - \frac{x^3}{3 \cdot 3!} + \frac{x^5}{5 \cdot 5!} - \frac{x^7}{7 \cdot 7!} + \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)(2n+1)!}\end{aligned} \tag{8}$$

Выделим два общих члена a_n, a_{n+1} из $Si(x)$ и найдём $q_n = \dfrac{a_{n+1}}{a_n}$:

$$a_n = \frac{(-1)^n x^{2n+1}}{(2n+1)(2n+1)!}, \quad a_{n+1} = \frac{(-1)^{n+1} x^{2n+3}}{(2n+3)(2n+3)!}. \tag{9}$$

$$q_n = \frac{-x^2(2n+1)}{(2n+2)(2n+3)^2}. \tag{10}$$

Вычислим по ней значения функции в 11 узлах интерполяции (Таблица 1).

x_i	$erf(x_i)$
0,0	0,0000000000
0,4	0,3964614570
0,8	0,7720957994
1,2	1,1080472469
1,6	1,3891806602
2,0	1,6054129601
2,4	1,7524855137
2,8	1,8320965767
3,2	1,8514009714
3,6	1,8219480515
4,0	1,7582031488

Таблица 1 - точки x_i и значения разложения функции $Si(x_i)$

Далее вычислим $Si(x)$ с помощью пяти составных квадратурных формул и составим для каждой формулы таблицу, в которой первый столбец - одиннадцать точек разбиения, второй - значение интеграла, третий - значение интеграла, вычисленного соответсвующим методом. В четвертом столбце находятся значения погрешности. В последнем - количество разбиений, необходимых для подсчета интеграла с заданной точностью.

1. Правые прямоугольники:

x_i	$J_0(x_i)$	$J_(x_i)$	$\lvert J_0(x_i) - J_N(x_i)\rvert$	N
0,0	0,0000000000	0,0000000000	0,0000000000	2
0,4	0,3964614570	0,3964561820	0,0000052750	1024
0,8	0,7720957994	0,7720555663	0,0000402331	1024
1,2	1,1080472469	1,1079159975	0,0001312494	1024
1,6	1,3891806602	1,3888883591	0,0002923012	1024
2,0	1,6054129601	1,6048804522	0,0005325079	1024
2,4	1,7524855137	1,7516450882	0,0008404255	1024
2,8	1,8320965767	1,8308923244	0,0012042522	1024
3,2	1,8514009714	1,8498086929	0,0015922785	1024
3,6	1,8219480515	1,8199745417	0,0019735098	1024
4,0	1,7582031488	1,7558794022	0,0023237467	1024

Таблица 2 - таблица значений для формулы правых прямоугольников

2. Центральные прямоугольники:

x_i	$J_0(x_i)$	$J_(x_i)$	$\lvert J_0(x_i) - J_N(x_i)\rvert$	N
0,0	0,0000000000	0,0000000000	0,0000000000	2
0,4	0,3964614570	0,3964616656	0,0000002086	64
0,8	0,7720957994	0,7720960975	0,0000002980	256
1,2	1,1080472469	1,1080478430	0,0000005960	512
1,6	1,3891806602	1,3891806602	0,0000000000	1024
2,0	1,6054129601	1,6054127216	0,0000002384	1024
2,4	1,7524855137	1,7524878979	0,0000023842	1024
2,8	1,8320965767	1,8320971727	0,0000005960	1024
3,2	1,8514009714	1,8514013290	0,0000003576	512
3,6	1,8219480515	1,8219479322	0,0000001192	1024
4,0	1,7582031488	1,7582037449	0,0000005960	512

Таблица 3 - таблица значений для формулы центральных прямоугольников

3. Формула трапеций:

x_i	$J_0(x_i)$	$J_(x_i)$	$\lvert J_0(x_i) - J_N(x_i)\rvert$	N
0,0	0,0000000000	0,0000000000	0,0000000000	2
0,4	0,3964614570	0,3964614570	0,0000000000	128
0,8	0,7720957994	0,7720955014	0,0000002980	256
1,2	1,1080472469	1,1080472469	0,0000000000	512
1,6	1,3891806602	1,3891803026	0,0000003576	1024
2,0	1,6054129601	1,6054120064	0,0000009537	512
2,4	1,7524855137	1,7524851561	0,0000003576	1024
2,8	1,8320965767	1,8320960999	0,0000004768	1024
3,2	1,8514009714	1,8514009714	0,0000000000	1024
3,6	1,8219480515	1,8219457865	0,0000022650	1024
4,0	1,7582031488	1,7582020760	0,0000010729	1024

Таблица 4 - таблица значений для формулы трапеций

4. Формула Симпсона

4.1. Вывод формулы Симпсона через интегральный полином Лагранжа:
Формула для полинома Лагранжа:

$$L_n(x) = \sum_{i=0}^{n} f(x_i) \prod_{i \neq j, j=0}^{n} \frac{x - x_j}{x_i - x_j} \tag{11}$$

По трём узлам $(x_1 = a, x_2 = \dfrac{a+b}{2}, x_3 = b) : L_2 = f(a) \left(\dfrac{x - \dfrac{a+b}{2}}{a - \dfrac{a+b}{2}} \right) \left(\dfrac{x-b}{a-b} \right) +$
$f\left(\dfrac{a+b}{2}\right) \left(\dfrac{x-a}{\dfrac{a+b}{2} - a} \right) \left(\dfrac{x-b}{\dfrac{a+b}{2} - b} \right) + f(b) \left(\dfrac{x - \dfrac{a+b}{2}}{b - \dfrac{a+b}{2}} \right) \left(\dfrac{x-b}{b-a} \right).$

Проинтегрируем выражение по интервалу [a,b]:

$$\int_a^b L_2(x)\mathrm{d}x = f(a)c_1 + f\left(\frac{a+b}{2}\right) c_2 + f(b)c_3 \tag{12}$$

где $c_1 = \dfrac{b-a}{6}, c_2 = \dfrac{2}{3}(b-a), c_3 = \dfrac{b-a}{6}.$

Тогда:

$$\int_a^b L_2(x)\mathrm{d}x = \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right) \tag{13}$$

4.2. Значения, полученные для формулы Симпсона:

x_i	$J_0(x_i)$	$J_(x_i)$	$\|J_0(x_i) - J_N(x_i)\|$	N
0,0	0,0000000000	0,0000000000	0,0000000000	2
0,4	0,3964614570	0,3964614868	0,0000000298	4
0,8	0,7720957994	0,7720957398	0,0000000596	8
1,2	1,1080472469	1,1080472469	0,0000000000	8
1,6	1,3891806602	1,3891803026	0,0000003576	16
2,0	1,6054129601	1,6054130793	0,0000001192	16
2,4	1,7524855137	1,7524855137	0,0000000000	16
2,8	1,8320965767	1,8320968151	0,0000002384	16
3,2	1,8514009714	1,8514008522	0,0000001192	32
3,6	1,8219480515	1,8219482899	0,0000002384	16
4,0	1,7582031488	1,7582032681	0,0000001192	16

Таблица 5 - таблица значений для формулы Симпсона

5. Формула Гаусса:

x_i	$J_0(x_i)$	$J_(x_i)$	$\|J_0(x_i) - J_N(x_i)\|$	N
0,0	0,0000000000	0,0000000000	0,0000000000	2
0,4	0,3964614570	0,3964614868	0,0000000298	4
0,8	0,7720957994	0,7720956802	0,0000001192	4
1,2	1,1080472469	1,1080472469	0,0000000000	8
1,6	1,3891806602	1,3891805410	0,0000001192	16
2,0	1,6054129601	1,6054128408	0,0000001192	16
2,4	1,7524855137	1,7524856329	0,0000001192	16
2,8	1,8320965767	1,8320965767	0,0000000000	16
3,2	1,8514009714	1,8514007330	0,0000002384	16
3,6	1,8219480515	1,8219481707	0,0000001192	16
4,0	1,7582031488	1,7582032681	0,0000001192	16

Таблица 6 - таблица значений для формулы Гаусса

3 Выводы

В ходе работы были изучены численные методы вычисления интегралов с применением пяти различных квадратурных формул. Анализ результатов показал, что методы Гаусса и Симпсона являются наиболее эффективными. Они обеспечивает высокую точность при минимальном числе разбиений благодаря специально подобранным узлам для приближенного вычисления интеграла.

4 Листинг программы

```
import math

STEPS = 1024
LEFT_BORDER = 0
EPSILON = 1e-6
E = 2.71828182846

def Tfunc(x):
    n = 0
    node_0 = x
    ans = x
    while abs(node_0) > 1e-6:
        q = ((-1) * x * x * (2 * n + 1)) / ((2 * n + 2) * (2 * n + 3)
            * (2 * n + 3))
        node_0 *= q
        ans += node_0
        n += 1
    return ans

def func(t):
    if t != 0:
        return math.sin(t) / t
    return 1

def leftRectangles(func, a, b, steps):
    result = 0.0
    stepSize = (b - a) / steps
    for i in range(steps):
        x_i = a + stepSize * i
        result += stepSize * func(x_i)
    return result

def rightRectangles(func, a, b, steps):
    result = 0.0
    stepSize = (b - a) / steps
    for i in range(1, steps + 1):
        x_i_1 = a + stepSize * i
        result += stepSize * func(x_i_1)
    return result

def middleRectangles(func, a, b, steps):
```

```
    result = 0.0
    stepSize = (b - a) / steps
    for i in range(steps):
        x_i = a + stepSize * i
        x_i_1 = a + stepSize * (i + 1)
        result += stepSize * func((x_i + x_i_1) / 2)
    return result

def trapezeFormula(func, a, b, steps):
    result = func(a) + func(b)
    stepSize = (b - a) / steps
    for i in range(1, steps):
        x_i_1 = a + stepSize * i
        result += 2 * func(x_i_1)
    result *= stepSize / 2
    return result

def SypmsonsFormula(func, a, b, steps):
    stepSize = (b - a) / steps
    result = 0
    x = a
    for i in range(steps):
        result += (func(x) + 4 * func(x + stepSize / 2) + func(x +
           stepSize)) * stepSize / 6
        x += stepSize
    return result

def GaussFormula(func, a, b, steps):
    stepSize = (b - a) / steps
    ad1 = (1 - 1.0 / math.sqrt(3)) * stepSize / 2
    ad2 = (1 + 1.0 / math.sqrt(3)) * stepSize / 2
    result = 0
    x = a
    for i in range(steps):
        result += (func(x + ad1) + func(x + ad2)) * stepSize / 2
        x += stepSize
    return result

def calculateFunc(points, function):
    for point in points:
        steps = 1
        lastResult = 0.0
        currentResult = 0.0
```

```
        while True:
            steps *= 2
            lastResult = currentResult
            currentResult = function(func, LEFT_BORDER, point, steps)
            if abs(lastResult - currentResult) <= EPSILON or steps >= STEPS:
                break

        difference = abs(Tfunc(point) - currentResult)

        print(f"x_i = {point:.1f} | J_o = {Tfunc(point):.10f} | J_n = {currentResult:.10f} | |J_o - J_n| = {difference:.10f} | N = {steps}")

if __name__ == "__main__":
    points = [0.0, 0.4, 0.8, 1.2, 1.6, 2.0, 2.4, 2.8, 3.2, 3.6, 4.0]
    print("Правые прямоугольники")
    calculateFunc(points, rightRectangles)
    print("Левые прямоугольники")
    calculateFunc(points, leftRectangles)
    print("Центральные прямоугольники")
    calculateFunc(points, middleRectangles)
    print("Трапеции")
    calculateFunc(points, trapezeFormula)
    print("Симпсон")
    calculateFunc(points, SypmsonsFormula)
    print("Гаус")
    calculateFunc(points, GaussFormula)
```