

Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

ОТЧЕТ

ПО ПРЕДМЕТУ “МЕТОДЫ ЧИСЛЕННОГО АНАЛИЗА”

НА ТЕМУ “ Приближенное вычисление интегралов”

студентки 2 курса 2 группы

Курец Любви Олеговны

Преподаватель

Горбачева Юлия Николаевна

Постановка задачи

- Вычислить интеграл $\int_a^b f(x)dx$ с точностью $\varepsilon = 10^{-4}$, используя квадратурные формулы, указанные в варианте задания, и правило Рунге оценки погрешности. Результаты оформить в виде таблицы:

- Вычислить интеграл $\int_a^b f(x)dx$ по квадратурной формуле Гаусса с 2, 3 и 4 узлами и единичной весовой функцией на $[a, b]$. Результаты оформить в виде таблицы:

$f(x)$	a	b	Квадратурные формулы
$\frac{\sin x}{1+x^2}$	0	$\frac{\pi}{2}$	Левых прямоугольников, средних прямоугольников, Симпсона.

Краткие теоретические сведения:

Формула для левых прямоугольников:

$$\int_a^b f(x) dx \approx \sum_{i=0}^{n-1} f(x_i)(x_{i+1} - x_i).$$

Формула для средних прямоугольников:

$$\int_a^b f(x) dx \approx \sum_{i=0}^{n-1} f\left(\frac{x_i + x_{i+1}}{2}\right)(x_{i+1} - x_i) = \sum_{i=1}^n f\left(\frac{x_{i-1} + x_i}{2}\right)(x_i - x_{i-1}).$$

Формула Симпсона:

$$\int_a^b f(x)dx \approx \sum_{i=1}^N \frac{f(x_{i-1}) + 4f\left(\frac{x_{i-1}+x_i}{2}\right) + f(x_i)}{6}(x_i - x_{i-1})$$

Для оценки погрешности вычислений используется правило Рунге.

Погрешность вычисления значения интеграла при числе шагов, равном $2n$, определяется по формуле Рунге:

$\Delta_{2n} \approx \Theta |I_{2n} - I_n|$, для формул прямоугольников и трапеций $\Theta = \frac{1}{3}$, а для формулы Симпсона $\Theta = \frac{1}{15}$.

n	x	A
2	-0.57735027, 0.57735027	1, 1
3	-0.77459667, 0, 0.77459667	5/9, 8/9, 5/9
4	-0.86113631, -0.33998104, 0.33998104, 0.86113631	0.34785484, 0.65214516, 0.65214516, 0.34785484

Для вычисления интеграла непрерывной функции на интервале от a до b квадратурная формула Гаусса может быть записана следующим образом

$$\int_a^b p(x)f(x) dx \approx \sum_{k=1}^n A_k f(x_k),$$

значения коэффициентов A_k и узлов x_k приведены в таблице.

Листинг:

```
import math
a = 0
b = math.pi/2
eps = 0.0001
import math
import numpy as np
import matplotlib.pyplot as plt

def f(x):
    return math.sin(x) / (1 + x * x)
def leftRec(n):
    h = (b - a) / n
    x = np.arange(a, b, h);
    ans = 0
    for i in range(n):
        ans += f(x[i])
    ans *= h
    return ans
def midRec(n):
    h = (b - a) / n
    x = np.arange(a + h / 2, b, h)
    ans = 0
    for i in range(n):
        ans += f(x[i])
    ans *= h
    return ans

def simpson(n):
    if (n % 2 == 1):
        n += 1

    h = (b - a) / n

    even = np.arange(a + 2 * h, b, 2 * h)
    odd = np.arange(a + 2 * h + h, b, 2 * h)

    ans = x(A) + 4 * x(A + h) + x(B)
    n = int(n / 2) - 1
    for i in range(n):
        ans += 2 * f(even[i])
        ans += 4 * f(odd[i])

    ans *= h / 3
```

```

    return ans
def runge(function, v1, v2):
    if (function == leftTriang or function == midTriang):
        return abs(v1 - v2) / 3
    if (function == simpson):
        return abs(v1 - v2) / 15
def answer(function):
    n = 4
    error = 1
    while (error >= EPS):
        i1 = function(n)
        i2 = function(2 * n)
        error = runge(function, i1, i2)
        h = (b - a) / n
        h2 = (b - a) / (2 * n)
        print("h = {:.6f}".format(h), "h/2 = {:.6f}".format(h2),
              "I1 = {:.6f}".format(i1), "I2 = {:.6f}".format(i2),
              "error = {:.6f}".format(error))
        n *= 2
answer(leftRec)
answer(midTRec)
answer(simpson)
def lezhandr(x, n):
    l = np.zeros(n)
    l[0] = 0
    l[1] = x
    for i in range(n - 2):
        l[i + 2] = (2 * n + 1) / (n + 1) * x * l[i + 1] - n / (n + 1) * l[i]
    return l
a = np.array([[1, 1], [5 / 9, 8 / 9, 5 / 9], [0.3478, 0.6521, 0.6521, 0.3478]])
l = np.array([[-0.5773, 0.5773], [-0.7746, 0, 0.7746], [-0.8611, -0.3340, 0.3340, 0.8611]])
def gaussianQuadrature(n):
    if (n != 2 and n != 3 and n != 4):
        return 0
    ans = 0.
    for i in range(n):
        t = 0.5 * ((b - a) * l[n - 2][i] + b + a)
        ans += a[n - 2][i] * f(t)
    ans *= (b - a) / 2.
    return ans
for i in range(2, 5):
    print("i =", i, "i = {:.6f}".format(gaussianQuadrature(i)))

```

Результаты:

Квадратурная формула	Шаг	Приближенное значение интеграла	Погрешность
Левых прямоугольников	$h = 0.392699$ $h/2 = 0.196350$ $h/4 = 0.098175$ $h/8 = 0.049087$ $h/16 = 0.024544$ $h/32 = 0.012272$ $h/64 = 0.006136$ $h/128 = 0.003068$ $h/256 = 0.001534$	$I_h = 0.453877$ $I_{h/2} = 0.494597$ $I_{h/4} = 0.511808$ $I_{h/8} = 0.519647$ $I_{h/16} = 0.523376$ $I_{h/32} = 0.525193$ $I_{h/64} = 0.526090$ $I_{h/128} = 0.526535$ $I_{h/256} = 0.526757$	0.013573 0.005737 0.002613 0.001243 0.000606 0.000299 0.000148 0.000074
Средних прямоугольников	$h = 0.392699$ $h/2 = 0.196350$ $h/4 = 0.098175$ $h/8 = 0.049087$ $h/16 = 0.024544$	$I_h = 0.535317$ $I_{h/2} = 0.52901$ $I_{h/4} = 0.527486$ $I_{h/8} = 0.527105$ $I_{h/16} = .527010$	0.002100 0.000511 0.000127 0.000032
Симпсона	$h = 0.392699$ $h/2 = 0.196350$	$I_h = 0.528426$ $I_{h/2} = 0.527046$	0.000092

Количество узлов	Приближенное значение интеграла
2	0.523512
3	0.525788
4	0.528350

Вывод:

Из результатов видно, что метод Симпсона обладает лучшей скоростью сходимости, чем метод средних треугольников и метод левых треугольников; в свою очередь метод средних треугольников, сходится быстрее, чем метод левых прямоугольников.

С помощью квадратуры Гаусса можно получить высокую точность, используя небольшое число узлов, но не легко оценить погрешность полученного интерграла.