Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет: «Информатика и системы управления»

Кафедра: «Программное обеспечение ЭВМ и информационные технологии (ИУ-7)»

Лабораторная работа №5

Тема: Построение и программная реализация алгоритмов численного интегрирования.

Выполнил: **Варин Д.В.**

Группа: **ИУ7-46Б**

Оценка (баллы): ___

Преподаватель: **Градов В. М.**

Москва,
2021 г.

Цель работы: Получение навыков построения алгоритма вычисления двукратного интеграла с использованием квадратурных формул Гаусса и Симпсона.

Задание

Построить алгоритм и программу для вычисления численного интеграла при фиксированном значении параметра τ

$$\epsilon(\tau) = \frac{4}{\pi}\int_0^{\frac{\pi}{2}} d\phi \int_0^{\frac{\pi}{2}} [1 - \exp(-\tau\frac{1}{R})]\ cos\theta\ sin\theta\ d\theta\ d\varphi$$

$$\frac{1}{R} = \frac{2cos\theta}{1-cos^2\theta sin^2\varphi}$$

Применить метод последовательного интегрирования.
По одному направлению использовать формулу Гаусса, а по другому - формулу Симпсона.

Входные данные

1. Количество узлов сетки: N, M;
2. Значение параметра τ ;
3. Методы для направлений при последовательном интегрировании.

Выходные данные

1. Значение интеграла при заданном параметре;
2. График зависимости $\epsilon(\tau),\ t\in[0.05, 10]$;

Описание алгоритма

Полагая, что:

$$\int_{-1}^{1} t^k\,dt = \sum_{i=1}^{n} A_i t_i^k, k=0,1,\ldots,2n-1$$

Данная система даёт 2n соотношений для определения 2n неизвестных $A_i u t_i$

$$\int_{-1}^{1} t^k\,dt = \frac{1-(-1)^{k+1}}{k+1} = \begin{cases} \dfrac{2}{k+1}, \textit{при k четном} \\ 0. \textit{при k нечётном} \end{cases}$$

Система:

$$\begin{cases} \sum_{i=1}^{n} A_i = 2 \\ \sum_{i=1}^{n} A_i t_i = 0 \\ \sum_{i=1}^{n} A_i t_i^2 = \frac{2}{3} \\ \ldots\ldots\ldots \\ \sum_{i=1}^{n} A_i t_i^{2n-1} = 0 \end{cases}$$

Указанная система является нелинейной, и её решения находятся довольно трудно. Воспользуемся полиномами Лежандра:

$$P_n(x)=\frac{1}{2^n n!}\frac{d^n}{dx^n}\left[(x^2-1)^n\right], n=0,1,2,\ldots$$

Узлами формулы Гаусса являются нули данного полинома, а A_i можно найти из заданной выше системы уравнений.

При вычислении интеграла на произвольном интервале $[a,b]$, для применения квадратурной формулы Гаусса необходимо выполнить преобразование переменной следующим образом:

$$x=\frac{b+a}{2}+\frac{b-a}{2}t$$

В таком случае получаем конечную формулу для произвольного интервала интегрирования $[a,b]$:

$$\int_a^b f(x)dx=\frac{b-a}{2}\sum_{i=1}^{n} A_i f(x_i)$$

Также, для произвольного интервала формула Симпсона:

$$\int_a^b f(x)dx\approx\frac{h}{3}\sum_{i=0}^{\frac{N}{2}-1}(f_{2i}+4f_{2i+1}+f_{2i+2})$$

Указанные методы можно применять для приближённой оценки двукратного интеграла. Для прямоугольной области будем иметь:

$$I=\int_c^d\int_a^b f(x,y)dx\,dy=\int_a^b F(x)dx$$

При этом:

$$F(x)=\int_c^d f(x,y)dy$$

По каждой из осей введём некоторую сетку узлов. Каждый однократный интеграл будем вычислять по квадратурным формулам. Для разных направлений можем использовать квадратурные формулы разных порядков точности.
Для формулы Гаусса будем иметь:

$$I=\iint_G f(x,y)dx\,dy=\sum_{i=1}^{n}\sum_{j=1}^{m} A_i B_{ij} f(x_i,y_i)$$

$$A_i, B_{ij} - \text{известные постоянные}$$

Результаты работы программы

Алгоритм вычисления n корней полинома Лежандра n-ой степени.

Процедура вычисления корней полиному Лежандра произвольной степени выполняется численным методом, например, можно применить метод половинного деления. Сам полином строиться по рекуррентной формуле:

$$P_m(x)=\frac{1}{m}[(2m-1)xP_{m-1}(x)-(m-1)P_{m-2}(x)]$$

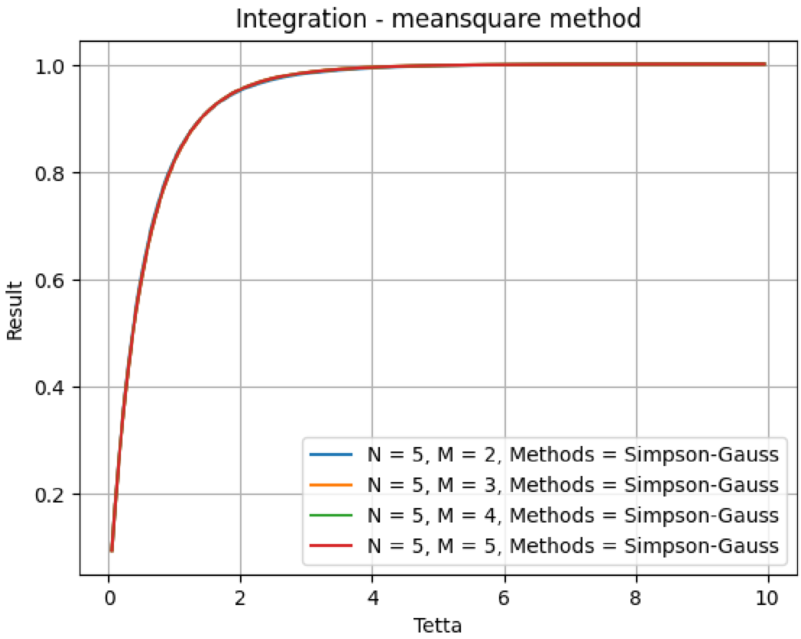
Это рекуррентное соотношение является одним из полезных свойств полинома Лежандра.
Для начала процесса используются полиномы $P_0(t) и\, P_1(t)$:

$$P_0(x)=1$$
$$P_1(x)=x$$

Процедура повторяется до тех пор, пока не будут найдены все n корней полинома. При этом учитывается свойство полиномов, согласно которому все эти корни располагаются на интервале [-1; 1], и они все действительны и различны, то есть кратные корни отсутствуют.

Влияние количества выбираемых узлов сетки на точность(узлы по каждому направлению)

При использовании метода Симпсона как внешнего метода интегрирования и при задании для него 5 узлов, метод Гаусса с различным количеством узлов будет давать одни и те же результаты.



Если для метода Симпсона зададим меньшее количество узлов, то получим расхождение с физическим смыслом задачи — больший «вклад» будет вносить метод, который является внешним.

$\tau=1$

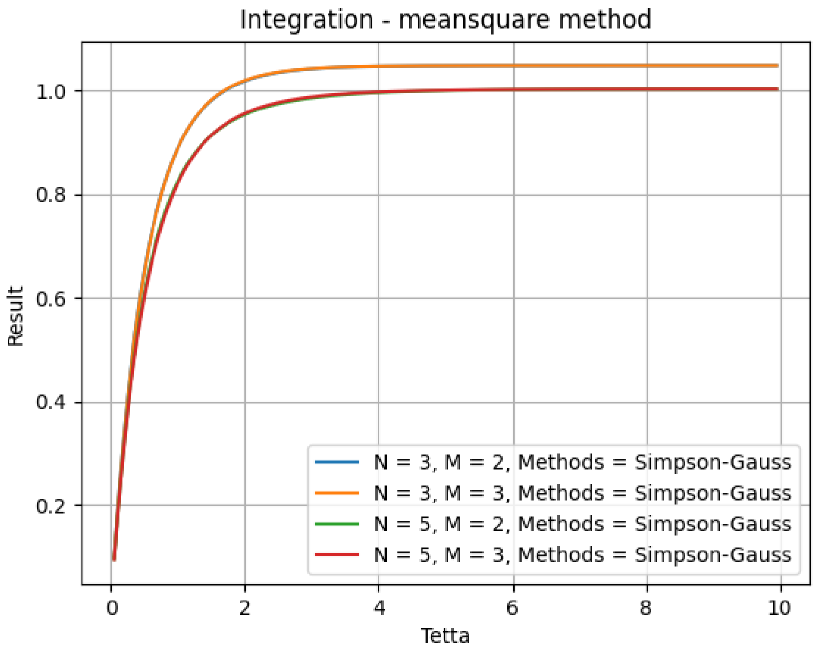
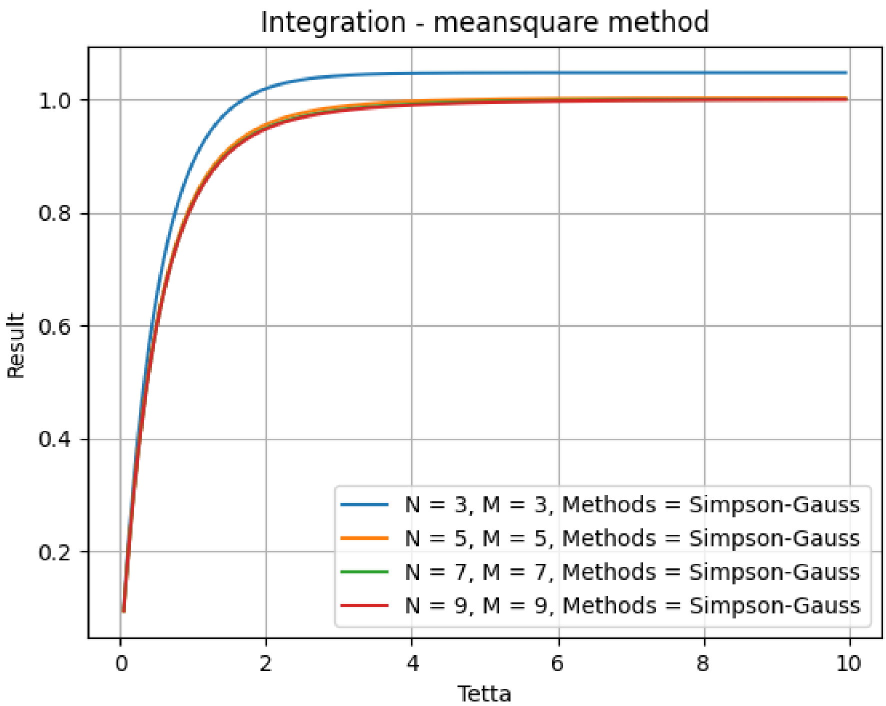


График зависимости $\varepsilon(\tau)$



Оптимальное значение достигнуто при размере сетки 5x5. $\tau=1$

Листинг кода

Программа написана на языке Python3 и состоит из 3х модулей.

main.py

```
from math import pi

import view
import integral


def main() -> None:
    tetta = [0.05, 0.05, 10.0]

    ns, ms = [], []
    md1s, md2s = [], []
    ints = []

    terminate = '0'
    while terminate == '0':
        try:
            ns.append(int(input("Input N: ")))
            ms.append(int(input("Input M: ")))
            param = float(input("Enter parameter: "))
            print("Entry integration mode (0 - Gauss, 1 - Simpson)")
            md1s.append(int(input("Outer integration mode: ")))
            md2s.append(int(input("Inner integration mode: ")))
        except ValueError:
            print("Invalid input data. Program is terminated.")
            return

        lm = [[0, pi / 2], [0, pi / 2]]

        ints.append(integral.Integral(lm, [ns[-1], ms[-1]], [md1s[-1],
md2s[-1]]))

        print("Result with {} as a parameter is {:.5f}".format(tetta, ints[-1]
(param)))

        terminate = input("If you want stop execution, entry not 0?: ")
    view.plot(ints, tetta, ns, ms, md1s, md2s)


if __name__ == '__main__':
    main()
```

view.py

```
import matplotlib.pyplot as plt
def plot(fs, sc, ns, ms, md1s, md2s) -> None:
    plt.clf()

    plt.title("Integration - meansquare method")
    plt.xlabel("Tetta")
    plt.ylabel("Result")
    plt.grid()

    for i in range(len(fs)):
        x, y = [], []
        j = sc[0]
        while j < sc[2]:
            x.append(j)
            y.append(fs[i](j))
            j += sc[1]

        plt.plot(x, y, label=get_label(ns[i], ms[i], md1s[i], md2s[i]))
    plt.legend()
    plt.show()
```

integral.py

```python
from math import cos, sin, exp, pi
from scipy.special import roots_legendre
from typing import Callable as Call, List


class Integral:
    def __init__(self, lm: List[List[float]], n: List[int], fn: List[int]):
        self.lm = lm
        self.n = n
        self.func_one = Integral.simpson if (fn[0]) else Integral.gauss
        self.func_two = Integral.simpson if (fn[1]) else Integral.gauss

    def __call__(self, p: float) -> float:
        f = Integral.__integrated(p)

        inner = lambda x: self.func_two(lambda val_one: f(x, val_one), self.lm[1][0],
self.lm[1][1], self.n[1])
        integ = lambda: self.func_one(inner, self.lm[0][0], self.lm[0][1], self.n[0])

        return integ()

    @staticmethod
    def simpson(f: Call[[float], float], a: float, b: float, n: int) -> float:
        if n < 3 or n % 2 == 0:
            raise Exception("Wrong value n")

        h = (b - a) / (n - 1.0)
        x = a
        res = 0.0
        for i in range((n - 1) // 2):
            res += f(x) + 4 * f(x + h) + f(x + 2 * h)
            x += 2 * h
        return res * h / 3

    @staticmethod
    def gauss(f: Call[[float], float], a: float, b: float, n: int) -> float:
        def pToV(p: float, c: float, d: float) -> float:
            return (d + c) / 2 + (d - c) * p / 2

        x, w = roots_legendre(n)
        return sum([(b - a) / 2 * w[i] * f(pToV(x[i], a, b)) for i in range(n)])

    @staticmethod
    def __integrated(p: float) -> Call[[float, float], float]:
        t = lambda x, y: 2 * cos(x) / (1 - sin(x) ** 2 * cos(y) ** 2)
        return lambda x, y: 4 / pi * (1 - exp(-p * t(x, y))) * cos(x) * sin(x)
```

Ответы на контрольные вопросы

1. В каких ситуациях теоретический порядок квадратурных формул численного интегрирования не достигается?

теоретический порядок квадратурных формул численного интегрирования не достигается в том случае, если подынтегральная функция не имеет соответствующих производных. В лекции нам был приведён пример о том, что, если на отрезке интегрирования не существуют 3-я и 4-я производные, то порядок точности формулы Симпсона будет только 2-ой, $O(h^2)$.

2. Построить формулу Гаусса численного интегрирования при одном узле.

$$\sum_{i=1}^{1} A_i = 2, \textit{значит } A_1 = 2$$
$$P_1(x) = x, \textit{значит } x_1 = 0 \ (1)$$

Для произвольного интервала будем иметь:
$$x = \frac{b+a}{2} + \frac{b-a}{2} t, \textit{где } t - \textit{нуль полинома Лежандра}$$
Таким образом, будем иметь:
$$\int_a^b f(x)\,dx = \frac{b-a}{2} \int_{-1}^{1} f\left(\frac{b+a}{2} + \frac{b-a}{2} t\right) dt$$
Также отметим, что:
$$\int_{-1}^{1} f(t)\,dt = A_1 f(x_1) \ (2)$$
При фактах (1) и (2) приходим к конечной формуле:
$$\int_a^b f(x)\,dx = (b-a) f\left(\frac{b+a}{2}\right)$$

3. Построить формулу Гаусса численного интегрирования при двух узлах.

В данном случае: $A_0 = 1, A_1 = 1, t_0 = \frac{-1}{\sqrt{3}}, t_1 = \frac{1}{\sqrt{3}}$

Получаем:
$$\int_a^b f(x)\,dx = \frac{b-a}{2} \left(f\left(\frac{b+a}{2} + \frac{b-a}{2}\left(\frac{-1}{\sqrt{3}}\right)\right) + f\left(\frac{b+a}{2} + \frac{b-a}{2}\frac{1}{\sqrt{3}}\right)\right)$$

4. Получить обобщенную кубатурную формулу, на основе формулы трапеций с тремя узлами по каждому направлению.

$$\int_c^d \int_a^b f(x,y)\,dxdy = \int_a^b F(x)\,dx$$

Где

$$F(x) = \int_c^d f(x,y)\,dy$$

Так будем иметь следующее:

$$\int_c^d \int_a^b f(x,y)\,dxdy = h_x \left(\frac{1}{2} F(x_0) + F(x_1) + \frac{1}{2} F(x_2)\right)$$

Где $h_x = \frac{b-a}{N}, N - \textit{количество узлов по направлению}$

Тогда:

$$\int_c^d \int_a^b f(x, y)\, dxdy = h_x h_y \cdot \left(\frac{1}{2}\right)\left(\frac{1}{2} f(x_0, y_0) + f(x_0, y_1) + \frac{1}{2} f(x_0, y_2)\right)$$

$$+\left(\frac{1}{2} f(x_1, y_0) + f(x_1, y_1) + \frac{1}{2} f(x_1, y_2)\right) + \frac{1}{2} \cdot \left(\frac{1}{2} f(x_2, y_0) + f(x_2, y_1) + \frac{1}{2} f(x_2, y_2)\right)$$

Где $h_y = \frac{d-c}{M}$, *где* $M -$ *количество узлов по направлению*