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»

Лабораторная работа № 5

Тема: Построение и программная реализация алгоритмов численного интегрирования.

Студент: Чаушев Александър

Группа: ИУ7-46Б

Оценка (баллы) _______________

Преподаватель : Градов В. М.

Москва.
2020 г.

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

Задание.

Работа основывается на материалах лекций №5 и 6.

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

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

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

θ, φ - углы сферических координат.

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

Краткие теоретические сведения и алгоритм.

Работа программы.
Вводяться значения количества узлов n и m, значение тау. Потом выбираем каким методом следует проинтегрировать(выбирается для 2 направлений).

Результаты. 1.
Описать алгоритм вычисления n корней полинома Лежандра n-ой степени при реализации формулы Гаусса

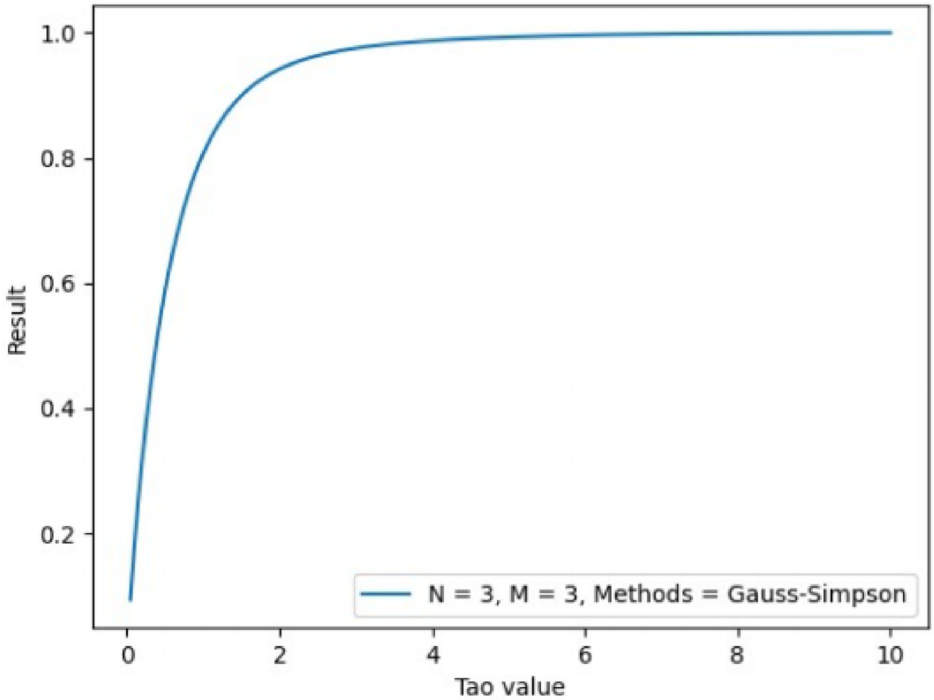
Все корни полинома лежат в интервале от [-1, 1], следовательно будет достаточно если мы рассмотрим тольку одну половину этого интервала. Корни полинома мы можем вычислить итеративно по методу Ньютона.

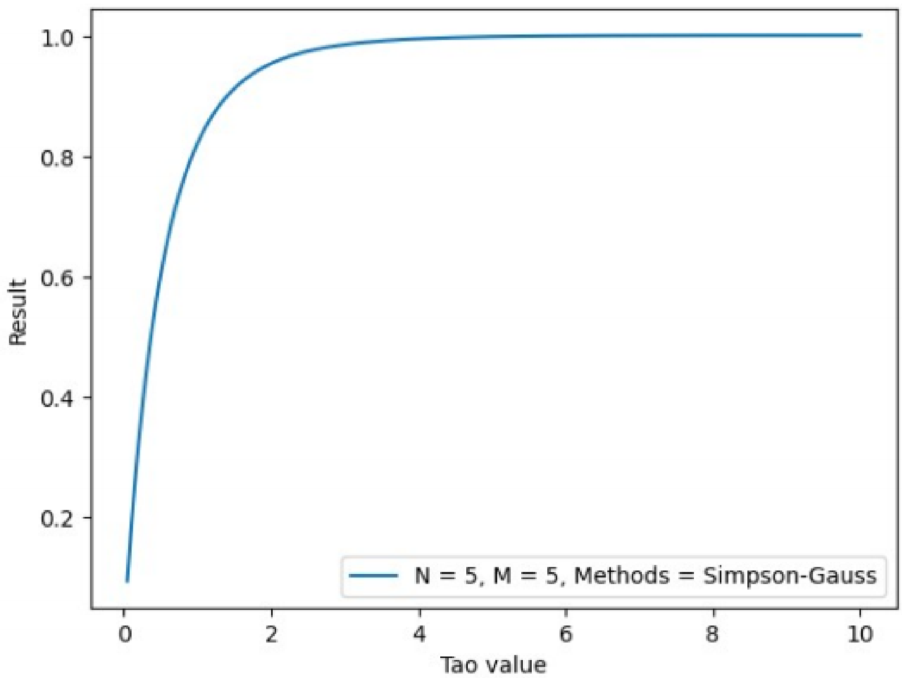
$$x_i^{(k+1)} = x_i^{(k)} — (P_n(x_i)^k) / (P_n^{'}(x_i)^k)$$

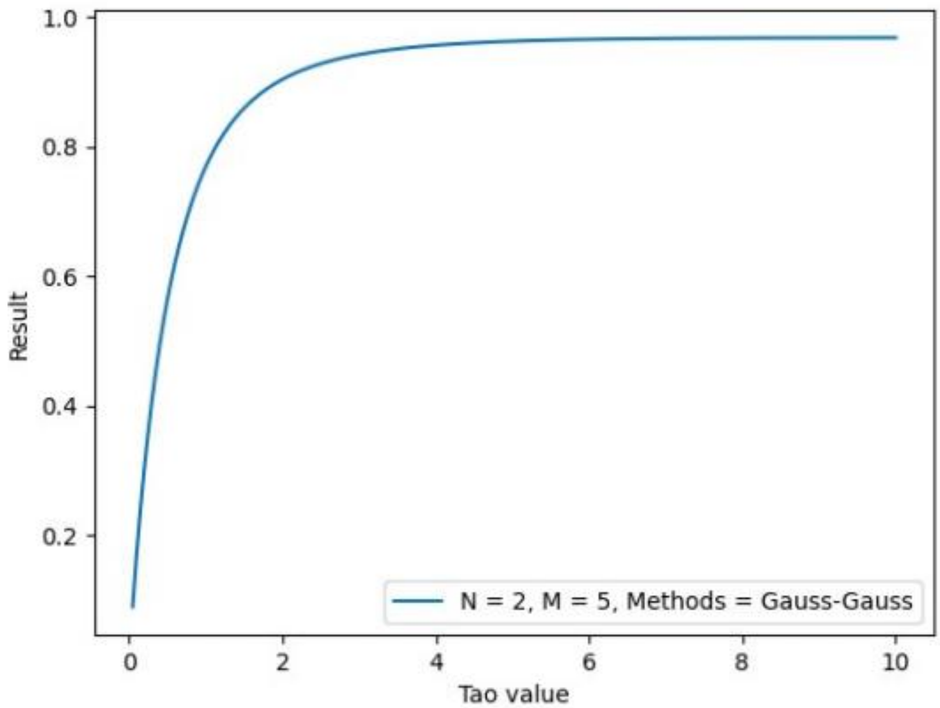
2. Исследовать влияние количества выбираемых узлов сетки по каждому направлению на точность расчетов.

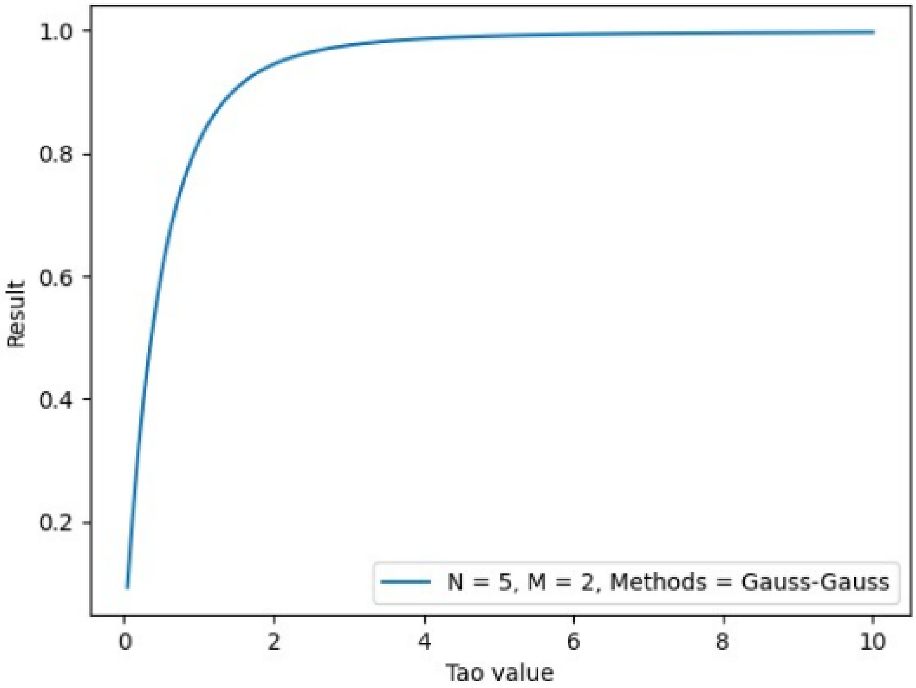
Задаем некоторе фиксированное значение для n = 3 при этом значение m не константное, оно увеличивается во время эксперимента. Значения полученные по таких условиях имеют большую погрешность, в сравнение с результатами полученныее при n = 5, m = 5. Так получается , когда неизменяемое остается значение m, а n увеличивается, но в этом случае результаты уже ближе к полученному результату при одинаковых узлах

3. Построить график зависимости ε(τ) в диапазоне изменения τ =0.05-10. Указать при каком количестве узлов получены результаты.




1.0
0.8
0.6
0.4
0.2
Result
N = 5, M = 5, Methods = Simpson-Gauss
0
2
4
6
8
10
Tao value


1.0
0.8
0.6
0.4
0.2
Result
N = 2, M = 5, Methods = Gauss-Gauss
0
2
4
6
8
10
Tao value



Ответы на контрольные вопросы:

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

- Когда подынтегральнаяфункция не имеет соответствующих производных. Например, если на отрезке интегрирования не существует 3-я и 4-я производные, то порядок тончности формула Симпсона будет только 2- ой(O(h2)).

2. Построить формулу Гаусса численного интегрирования при одном узле.

$$\sum_{i=1}^{n} A_i = 2 \Rightarrow A_i = 2$$

$$P_1(x) = x \Rightarrow x_1 = 0$$

$$\int_{-1}^{1} f(t)\,df = A_1 f(x_1) = 2f(0)$$

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

$$\int_{a}^{b} f(x)\,dx = \frac{b-a}{2} \int_{-1}^{1} f\left(\frac{b+a}{2} + \frac{b-a}{2} t\right) dt$$

$$= \frac{b-a}{2} \cdot 2 f\left(\frac{b+a}{2}\right)$$

3. Построить формулу Гаусса численного интегрирования при двух узлах.

$$P_2(x) = \frac{1}{2}\left(3x^2 - 1\right) \Rightarrow x = \pm\sqrt{\frac{1}{3}}$$

$$\left.\begin{cases} A_1 + A_2 = 2 \\ \underset{-\sqrt{\frac{1}{3}}}{A_1 t_1} + \underset{\sqrt{\frac{1}{3}}}{A_2 t_2} = 0 \end{cases}\right\} \Rightarrow A_1 = A_2 = 0$$

$$\int_{-1}^{1} f(t)\,dt = f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right)$$

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

4. Получить обобщенную кубатурную формулу, аналогичную (6.6) из лекции №6, для вычисления двойного интеграла методом последовательного интегрирования на основе формулы трапеций с тремя узлами по каждому направлению.

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

$$= h_x\left(\frac{1}{2}F(x_0) + F(x_1) + \frac{1}{2}F(x_2)\right) =$$

$$= h_x h_y\Big[\frac{1}{2}\Big(\frac{1}{2}f(x_0, y_0) + f(x_0; y_1) + \frac{1}{2}f(x_0; y_2)\Big) +$$

$$+ \frac{1}{2}f(x_1; y_0) + f(x_1, y_1) + \frac{1}{2}f(x_1, y_2) + \frac{1}{2}\Big(\frac{1}{2}f(x_2; y_0) +$$

$$+ f(x_2; y_1) + \frac{1}{2}f(x_2, y_2)\Big)\Big].$$

Код программы:

```
from numpy.polynomial.legendre import leggauss
from numpy import arange
from math import pi, cos, sin, exp
import matplotlib.pyplot as plt

def main_function(param):
    subfunc = lambda x, y: 2 * cos(x) / (1 - (sin(x) ** 2) * (cos(y) ** 2))
    func = lambda x, y: (4 / pi) * (1 - exp(-param * subfunc(x, y))) * cos(x) * sin(x)
    return func


def simpson(func, a, b, num_of_nodes):
    if (num_of_nodes < 3 or num_of_nodes & 1 == 0):
        raise ValueError

    h = (b - a) / (num_of_nodes - 1)
    x = a
    res = 0

    for _ in range((num_of_nodes - 1) // 2):
        res += func(x) + 4 * func(x + h) + func(x + 2 * h)
        x += 2 * h

    return res * (h / 3)


def func_2_to_1(func2, value):
    return lambda y: func2(value, y)


def t_to_x(t, a, b):
    return (b + a) / 2 + (b - a) * t / 2


def gauss(func, a, b, num_of_nodes):
    args, coeffs = leggauss(num_of_nodes)
    res = 0

    for i in range(num_of_nodes):
        res += (b - a) / 2 * coeffs[i] * func(t_to_x(args[i], a, b))

    return res

def integrate2(func, limits, num_of_nodes, integrators):
    inner = lambda x: integrators[1](func_2_to_1(func, x), limits[1][0], limits[1][1],
```

```
num_of_nodes[1])
    return integrators[0](inner, limits[0][0], limits[0][1], num_of_nodes[0])

def tao_graph(integrate_func, ar_params, label):
    X = list()
    Y = list()
    for t in arange(ar_params[0], ar_params[1] + ar_params[2], ar_params[2]):
        X.append(t)
        Y.append(integrate_func(t))
    plt.plot(X, Y, label=label)

def generate_label(n, m, func1, func2):
    res = "N = " + str(n) + ", M = " + str(m) + ", Methods = "
    res += "Simpson" if func1 == simpson else "Gauss"
    res += "-Simpson" if func2 == simpson else "-Gauss"
    return res


end = False
while not end:
    N = int(input("Enter N: "))
    M = int(input("Enter M: "))
    param = float(input("Enter param (tao): "))
    mode = bool(int(input("Enter external method (0 - Gauss; 1 - Simpson): ")))
    func1 = simpson if mode else gauss
    mode = bool(int(input("Enter internal method (0 - Gauss; 1 - Simpson): ")))
    func2 = simpson if mode else gauss

    param_integrate = lambda tao: integrate2(main_function(tao), [[0, pi / 2], [0, pi / 2]], [N, M], [func1, func2])
    print("Result with your parameter:", param_integrate(param))
    try:
        tao_graph(param_integrate, [0.05, 10, 0.05], generate_label(N, M, func1, func2))
    except ValueError:
        print("Be careful with simpson: argument should be > 2 and not even (3, 5...);")
    except ZeroDivisionError:
        print("Can't use 2 Simpsons, zero in denominator")
    end = bool(int(input("End? (0 - No, 1 - Yes): ")))

plt.legend()
plt.ylabel("Result")
plt.xlabel("Tao value")
plt.show()
```