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: Фролов Е.А.

Группа: ________

ИУ7-45Б **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 двукратного интеграла при фиксированном значении параметра τ.
Применить метод последовательного интегрирования. По одному направлению использовать формулу Гаусса, а по другому - формулу Симпсона.

Код программы

main.py

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


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


def simpson(func, a, b, nodes):
    if (nodes < 3 or nodes & 1 == 0):
        return

    h = (b - a) / (nodes - 1)
    x = a
    result = 0

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

    return result * (h / 3)


def arg_conversion(func2, value):  # преобразуем функцию args в одну функцию arg
    return lambda y: func2(value, y)


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

```python
def gauss(func, a, b, nodes):
    args, factor = leggauss(nodes)
    result = 0

    for i in range(nodes):
        result += (b - a) / 2 * factor[i] * func(t_to_x(args[i], a, b))

    return result


def integrate(func, limits, nodes, integrators):
    inside = lambda x: integrators[1](arg_conversion(func, x), limits[1][0], limits[1][1], nodes[1])
    return integrators[0](inside, limits[0][0], limits[0][1], nodes[0])


def tao_graph(integrate_func, arr_params, label):
    x = list()
    y = list()
    for i in arange(arr_params[0], arr_params[1] + arr_params[2], arr_params[2]):
        x.append(i)
        y.append(integrate_func(i))
    plt.plot(x, y, label=label)


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

    return result
```

```
finish = False
while not finish:
    print("Введите")
    N = int(input("N: "))
    M = int(input("M: "))
    param = float(input("Параметр (tao): "))
    mode = bool(int(input("Внешний метод (0 - Гаусс; 1 - Симпсон): ")))

    if mode:
        func1 = simpson
    else:
        func1 = gauss
    mode = bool(int(input("Внутренний метод (0 - Гаусс; 1 - Симпсон): ")))
    if mode:
        func2 = simpson
    else:
        func2 = gauss

    integr_param = lambda tao: integrate(bass_func(tao), [[0, pi / 2], [0, pi / 2]], [N, M], [func1, func2])
    print('Результат с вашим параметром:', integr_param(param))
    try:
        tao_graph(integr_param, [0.05, 10, 0.05], generate_label(N, M, func1, func2))
    except ValueError:
        print("Симпсоном: аргумент должен быть > 2 и не (3, 5...);")
    except ZeroDivisionError:
        print("Нельзя использовать 2 Симпсона, в знаменателе ноль")
    finish = bool(int(input("Конец? (0 - Да, 1 - Нет): ")))

plt.legend()
plt.ylabel("Результат")
plt.xlabel("Значение tao")
plt.show()
```

Пример работы программы

```
C:\Users\gimna\AppData\Local\Microsoft\WindowsApps\python3.7.exe C:/Users/gimna/Desktop/BMSTU/VychAlg/lab_05/main.py
Введите
N: 3
M: 3
Параметр (tao): 1
Внешний метод (0 - Гаусс; 1 - Симпсон): 0
Внутренний метод (0 - Гаусс; 1 - Симпсон): 1
Результат с вашим параметром: 0.8057719561370532
Конец? (0 - Да, 1 - Нет):
```

Вводим количество узлов N, M. Потом значение tao и указываем методы, которые требуется использовать в разных направлениях

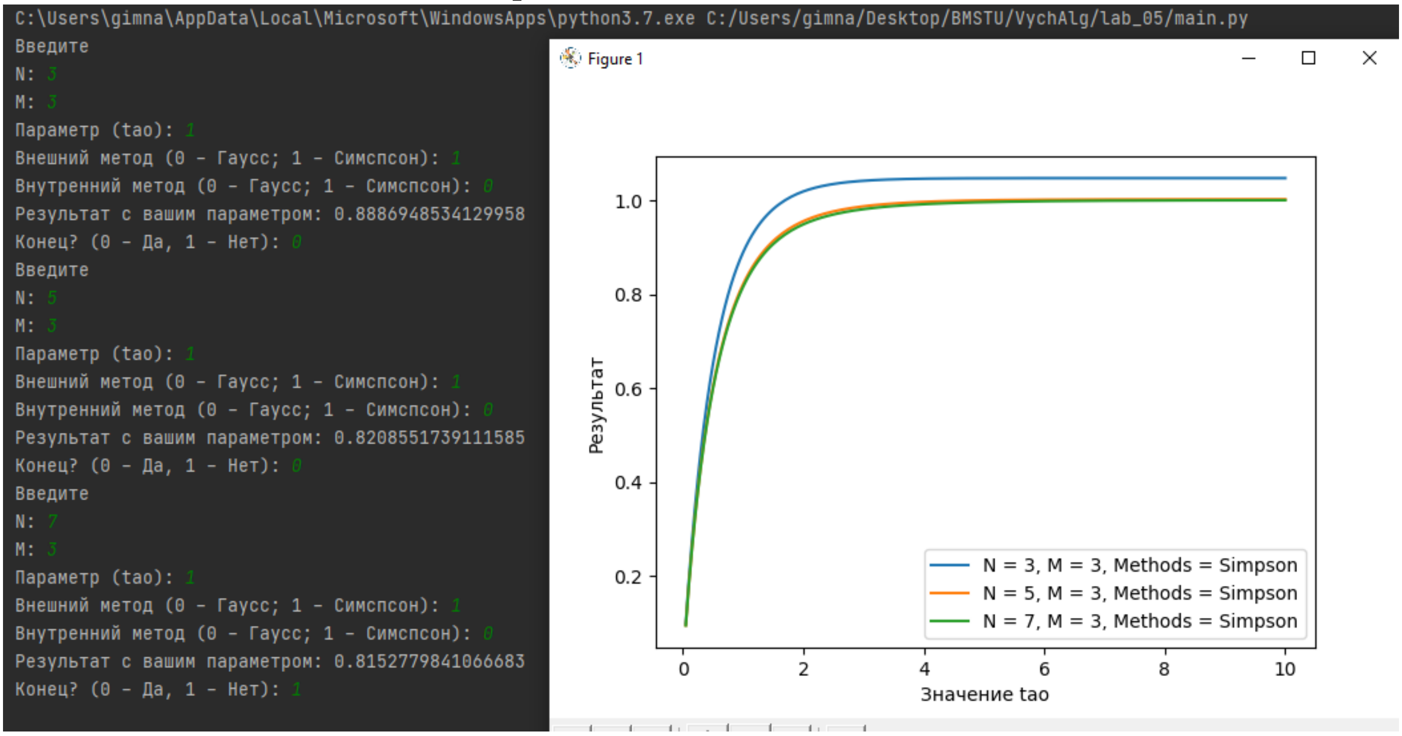
Результаты

1. Описать алгоритм вычисления n корней полинома Лежандра n-ой степени $P_n(x)$ при реализации формулы Гаусса.

Производная четной функции есть нечетная функция и наоборот, производная нечетной функции есть функция четная. Когда мы составляем функцию Лежандра, берется n-ая производная от четной функции и в результате получаем либо четную, либо нечетную функцию. Из свойств четных и нечетных функций имеем соотношение
f(x) = 0 => f(-x) = 0.
У полинома Лежандра n-ной степени - n действительных различных корней на отрезке [-1; 1], и из каждого корня x => корень -x. Тогда достаточно искать корни только на отрезке (0; 1].
В программе используется метод половинного деления на маленьких интервалах. Весь интервал разбивается на большое количество одинаковых отрезков и каждый из них проверяется. Если отрезок имеет разные знаки на концах - он содержит корень. Если один из концов отрезка = 0 - мы нашли корень. Так мы находим все корни полинома.

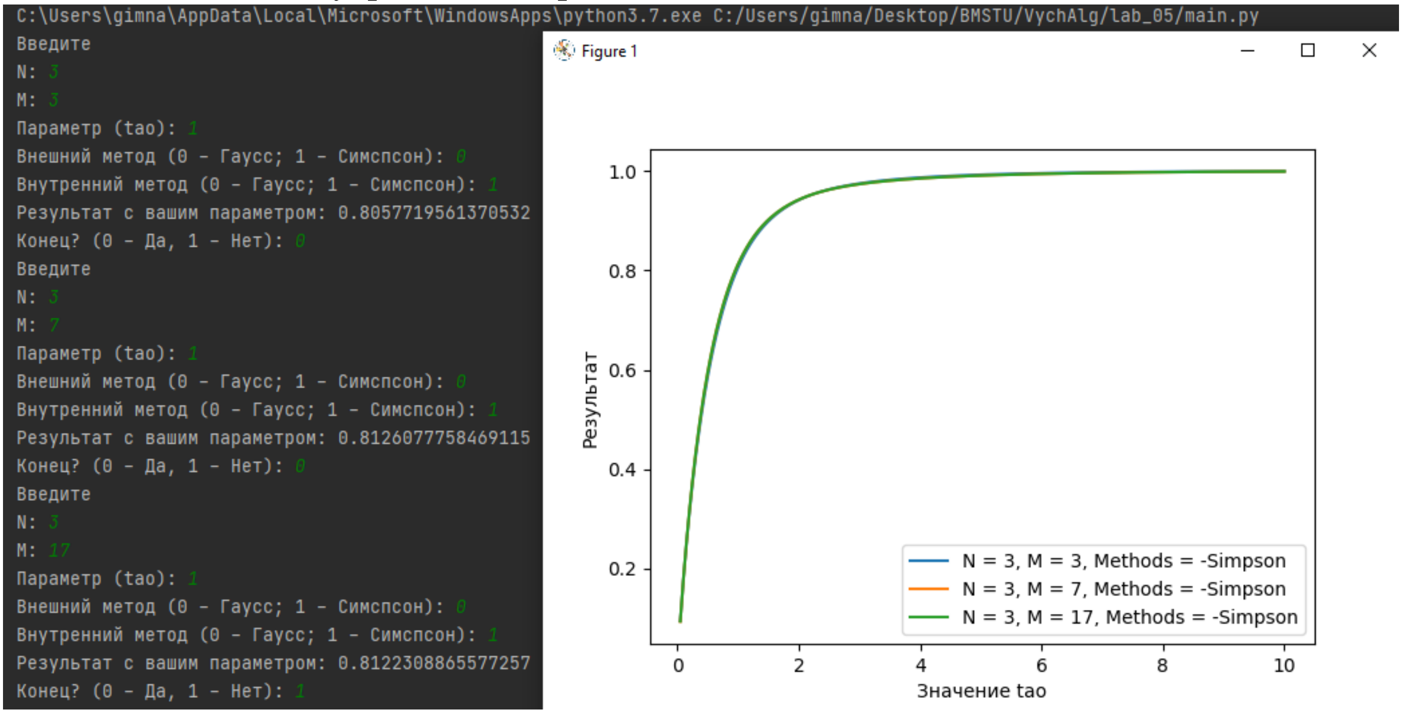
2. Исследовать влияние количества выбираемых узлов сетки по каждому направлению на точность расчетов.

Исследование для внешнего направления и метода Симпсона:



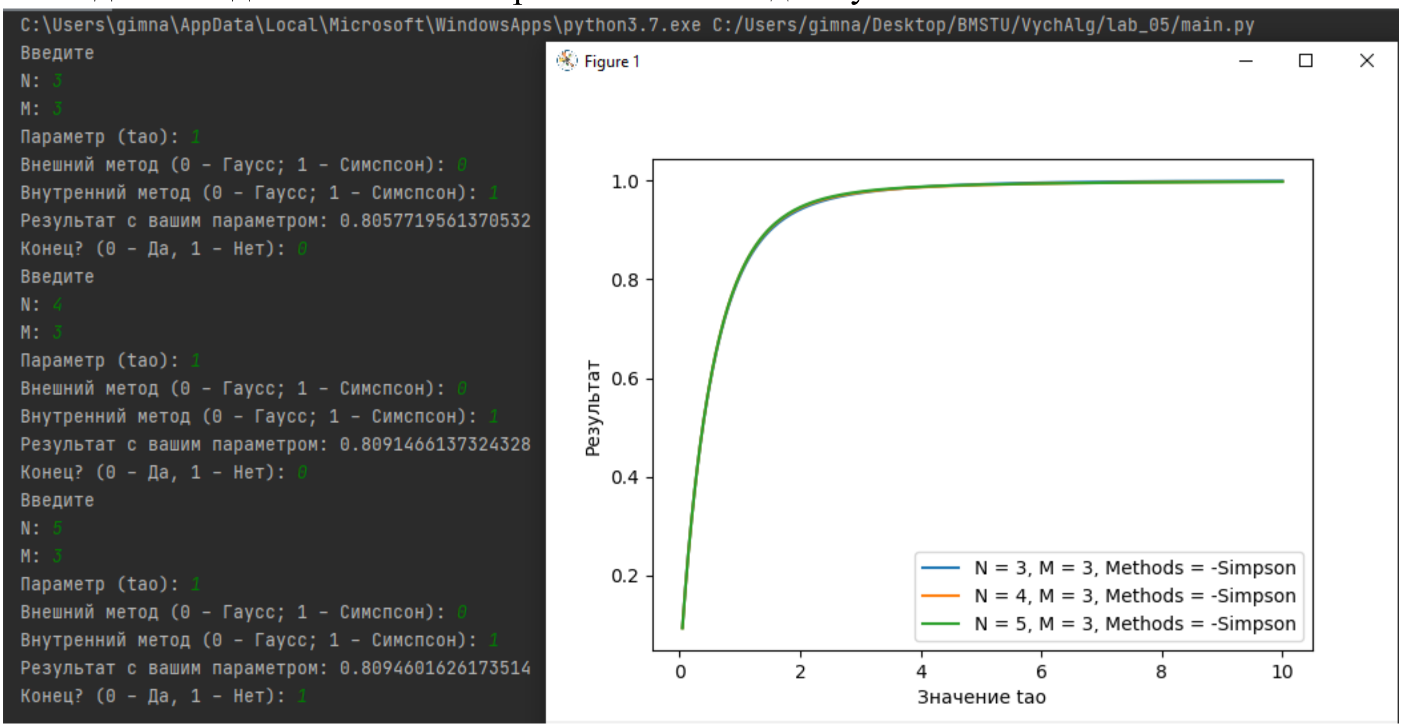
При количестве узлов = 3, график отклоняется от графиков с количеством узлов 5 и 7. Я сравнил с программой расчета интегралов в Вольфрам, значение правильное = 0.814
При большом количестве узлов результат точнее, при этом когда узлов 5+, результаты относительно близки друг к другу.

Исследование для внутреннего направления и метода Симпсона:



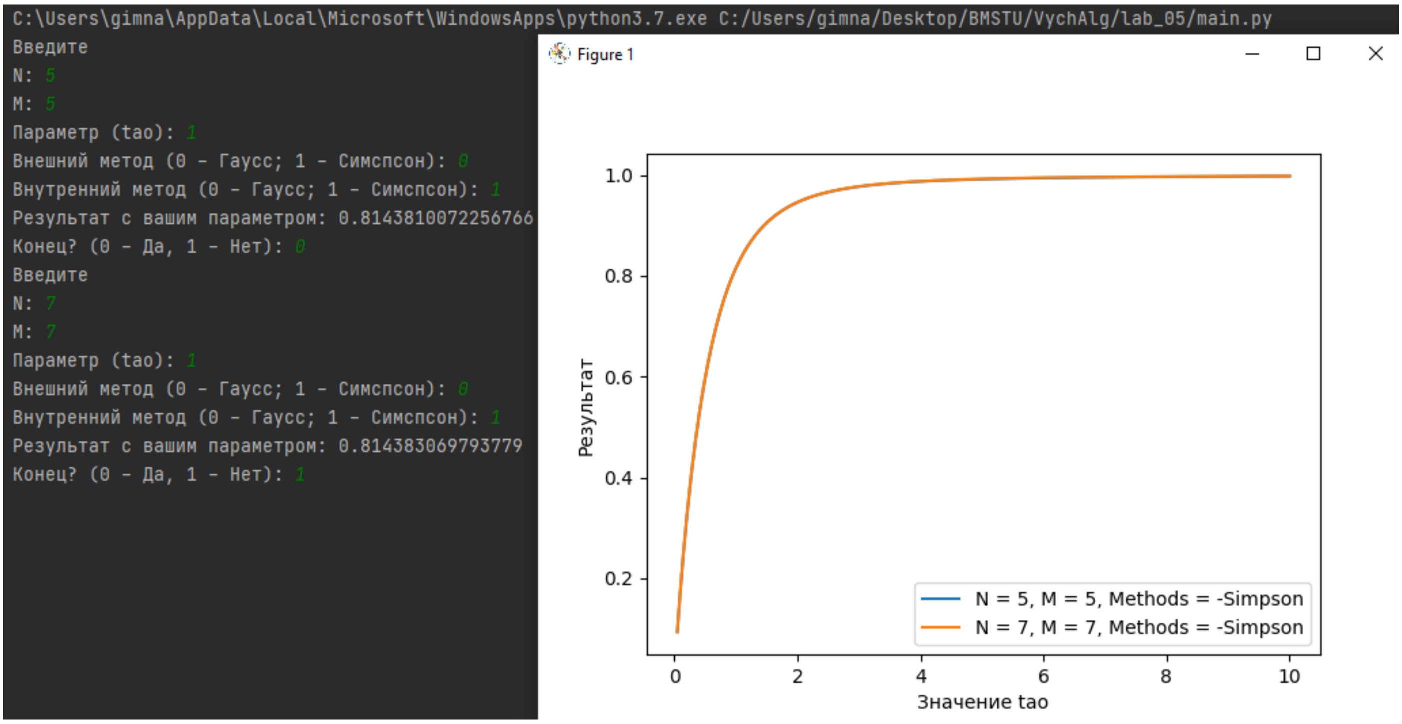
Результаты совпадают, следовательно, делаем вывод, что большее влияние оказывает точность внешнего интегрирования.

Исследования для внешнего направления и метода Гаусса:



Результаты близки к действительному при малом количестве узлов. Гаусс дает более точные результаты, чем Симпсон на внешнем направлении

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



Графики наложились друг на друга (близки по значениям), результат близок к действительному.

Вопросы при защите лабораторной работы.

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

Теоретический порядок квадратурных формул численного интегрирования не достигается в ситуациях, когда подынтегральная функция не имеет соответствующих производных. Порядок точности равен номеру последней существующей производной.

2. Построить формулу Гаусса численного интегрирования при одном узле.

$$\sum_{i=1}^{n} A_i = 2 \quad P_1(x) = x \Rightarrow x = 0$$

Полученная формула:

$$\int_a^b f(x)dx = \frac{b-a}{2} 2f(\frac{b+a}{2} + \frac{b-a}{2} \cdot 0) = (b-a)f(\frac{b+a}{2})$$

3. Построить формулу Гаусса численного интегрирования при двух узлах.

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

$$\begin{cases} A_1 + A_2 = 2 \\ -\frac{1}{\sqrt{3}}A_1 + \frac{1}{\sqrt{3}}A_2 = 0 \end{cases} \Rightarrow A_2 = A_1 = 1$$

$$\int_{-1}^{1} f(f)df = f(-\frac{1}{\sqrt{3}}) + f(\frac{1}{\sqrt{3}})$$

Полученная формула:

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

4. Получить обобщенную кубатурную формулу, на основе методе трапеций, с тремя узлами на каждом направлении.

Полученная формула:

$\int_c^d\int_a^b f(x, y)dxdy = \int_a^b dx\int_c^d f(x, y)dy = \int_a^b F(x)dx = \ h_x(0.5F0 + F1 + 0.5F2) = \ h_y h_x(0.5(0.5f(x_0, y_0) + f(x_0, y_1) + 0.5f(x_0, y_2)) + 0.5f(x_1, y_0) + f(x_1, y_1) + 0.5f(x_1, y_2) + 0.5(0.5f(x_2, y_0) + f(x_2, y_1) + 0.5f(x_2, y_2)))$

при $h_x = (b - a) / 2$, $h_y = (d - c) / 2$