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: Хамзина Р.Р.

Группа: ИУ7-43Б

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.

1 Исходные данные

1. ε(τ)

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

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

θ, φ - углы сферических координат.

2. N, M - число узлов сетки по каждому направлению.

   Для исследований N = 5, M = 3...7; N = 3...7, M = 5; N = M = 5.

3. τ- фиксированный параметр.

   Для исследования τ= 1; τ= 0.05...10.

2 Код программы

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


@dataclass
class Limits:
    """
        Пределы операций.
    """

    start = 0
    end = pi / 2

    t_start = 0.05
    t_end = 10


def explore_t(func, n, m):
    """
        Исследование ε(τ).
    """

    x, y  = [], []
    t_step = 0.05

    for t in arange(Limits.t_start, Limits.t_end + t_step, t_step):
        x.append(t)
        y.append(func(t))

    plt.plot(x, y, label = "N = {0}, M = {1}, Simpson-Gauss".format(n, m))
    plt.legend(fontsize = 20)
    plt.ylabel("∫∫", fontsize = 20)
    plt.xlabel("τ", fontsize = 20)
    plt.tick_params(labelsize = 20)
    plt.show()


def simpson(func, a, b, n):
    """
        Интегрирование при помощи
        формулы Симпсона.
    """

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

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

    result *= (h / 3)

    return result


def change_x(a, b, t):
    """
        Изменение x для применения
        квадратурной формулы Гаусса.
    """

    return (b + a) / 2 + (b - a) * t / 2


def change_func(func, x):
    """
        Изменение функции.
    """

    return lambda y: func(x, y)
```

```python
def gauss(func, a, b, m):
    """
        Интегрирование при помощи
        формулы Гаусса.
    """

    arguments, factors = leggauss(m)
    result = 0

    for i in range(m):
        now_x = change_x(a, b, arguments[i])
        result += (b - a) / 2 * factors[i] * func(now_x)

    return result


def get_integrable_func(t):
    """
        Интегрируемая функция.
    """

    inter_func = lambda x, y: 2 * cos(x) / \
                 (1 - (sin(x) ** 2) * (cos(y) ** 2))
    integrable_func = lambda x, y: (4 / pi) * \
                      (1 - exp(-t * inter_func(x, y))) * \
                      cos(x) * sin(x)

    return integrable_func


def get_double_integral(func, n, m):
    """
        Вычисление двукратного интеграла.
    """

    gauss_inter = lambda x: gauss(change_func(func, x), Limits.start, Limits.end, m)
    return simpson(gauss_inter, Limits.start, Limits.end, n)


def data_input():
    """
        Ввод n, m, t.
    """

    n = int(input("\nВведите N: "))
    m = int(input("\nВведите M: "))

    t = float(input("\nВведите τ: "))

    return n, m, t


def explore_count_nodes():
    """
       Исследование влияния количества узлов
       по каждому направлению.
    """

    t = 1
    n = 5

    for m in range(3, 8):
        double_integral = lambda x: get_double_integral(get_integrable_func(x), n, m)
        print("\nN = {0}, M = {1}, Двукратный интеграл = {2}".format(n, m, double_inte
gral(t)))

        x, y  = [], []
        t_step = 0.05

        for t in arange(Limits.t_start, Limits.t_end + t_step, t_step):
            x.append(t)
            y.append(double_integral(t))

        plt.plot(x, y, label = "N = {0}, M = {1}, Simpson-Gauss".format(n, m))

    plt.legend(fontsize = 20)
    plt.ylabel("∫∫", fontsize = 20)
    plt.xlabel("τ", fontsize = 20)
    plt.tick_params(labelsize = 20)
    plt.show()
```

```
    m = 5

    for n in range(3, 8):
        double_integral = lambda x: get_double_integral(get_integrable_func(x), n, m)
        print("\nN = {0}, M = {1}, Двукратный интеграл = {2}".format(n, m, double_inte
gral(t)))

        x, y  = [], []
        t_step = 0.05

        for t in arange(Limits.t_start, Limits.t_end + t_step, t_step):
            x.append(t)
            y.append(double_integral(t))

        plt.plot(x, y, label = "N = {0}, M = {1}, Simpson-Gauss".format(n, m))

    plt.legend(fontsize = 20)
    plt.ylabel("∫∫", fontsize = 20)
    plt.xlabel("τ", fontsize = 20)
    plt.tick_params(labelsize = 20)
    plt.show()


if __name__ == "__main__":
    """
        Ввод данных.
        Печать результата.
        Исследования.
    """

    n, m, t = data_input()

    double_integral = lambda x: get_double_integral(get_integrable_func(x), n, m)

    print("\nДвукратный интеграл = ", double_integral(t))

    explore_t(double_integral, n, m)
    explore_count_nodes()
```

3 Результаты работы

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

Одно из свойств полиномов Лежандра состоит в том, что на отрезке [-1;1] полином Лежандра n-ой степени $P_n(x)$ имеет n - различных, несовпадающих, не кратных, действительных корней. Поэтому мы можем найти корни полинома Лежандра любой степени. Для этого:

1) Делим заданный интервал следующим образом:

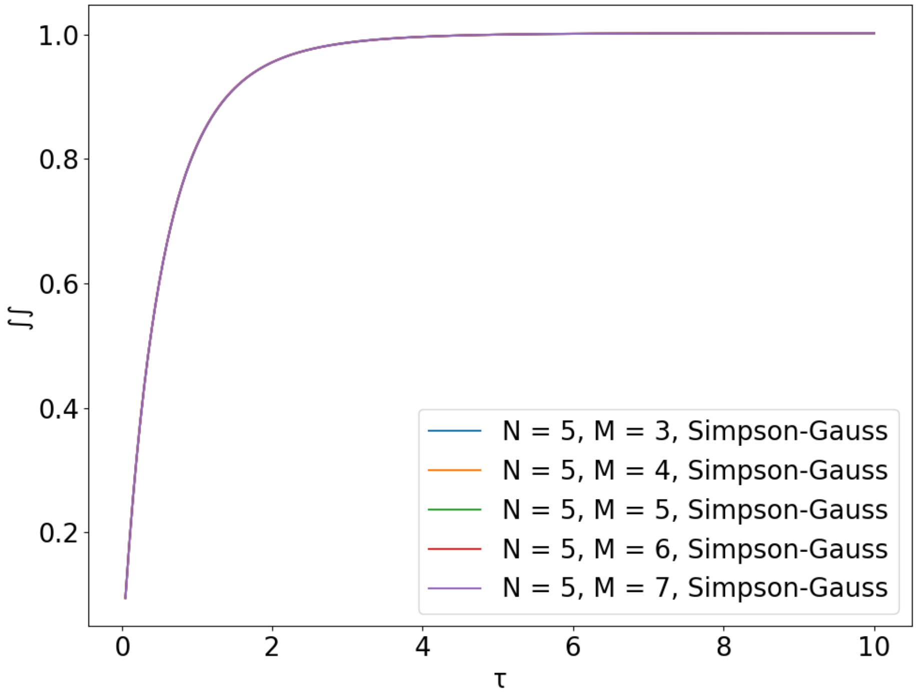
Число отрезков, удовлетворяющих такому условию: произведение значений в границах отрезка должно быть меньше 0 (это означает, что функция имеет разные знаки на концах отрезка), - должно равняться n - степени полинома.

2) Методом половинного деления находим корни:

Разделим отрезок [a; b] пополам, т. е. найдем x = (a + b) / 2 и вычислим значение функции F(x) в этой точке. Если окажется, что F(x) = 0, то x - корень уравнения. Если F(x) ≠ 0, то выбираем ту половину отрезка [a; x] или [x; b], на концах которой функция F(x) имеет противоположные знаки. При повторном делении производятся те же самые операции: новый отрезок [a; b] делится пополам, вычисляется значение функции в точке деления F(x) и определяется отрезок, содержащий истинный корень.

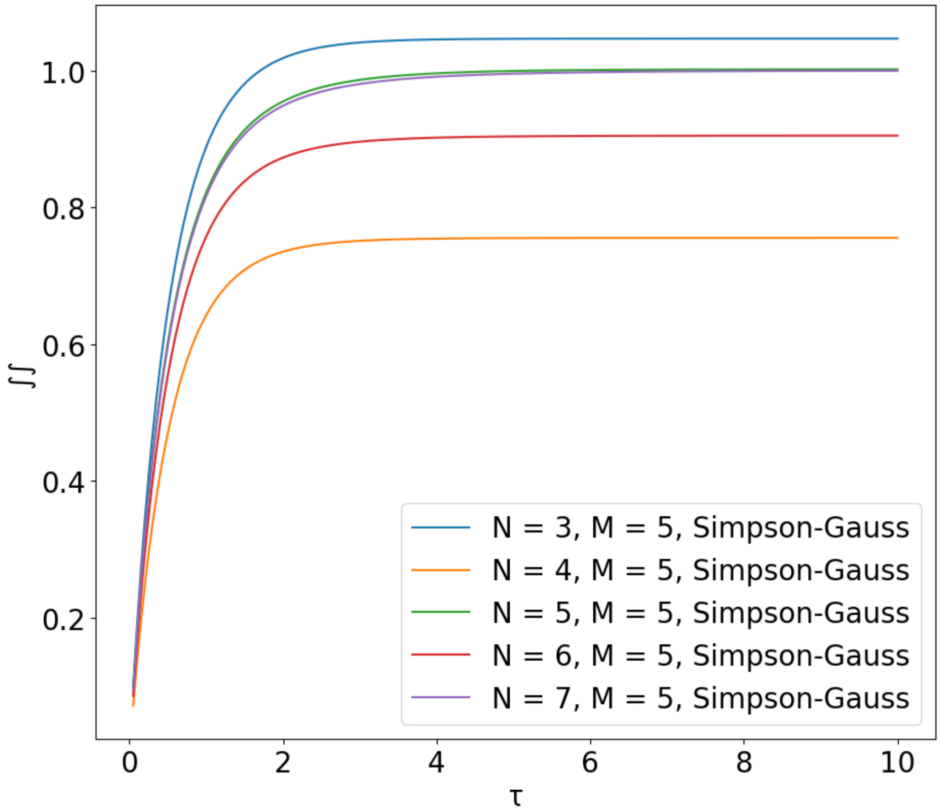
2. Исследование влияния количества выбираемых узлов сетки по каждому направлению на точность расчетов.

N = 5, M = 3...7:



При изменении числа узлов “внутреннего” метода интегрирования результаты совпадают.

N = 3...7, M = 5:



При изменении числа узлов “внешнего” метода интегрирования графики отклоняются друг от друга.

Вывод: “внешний” метод больше влияет на точность результатов.

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

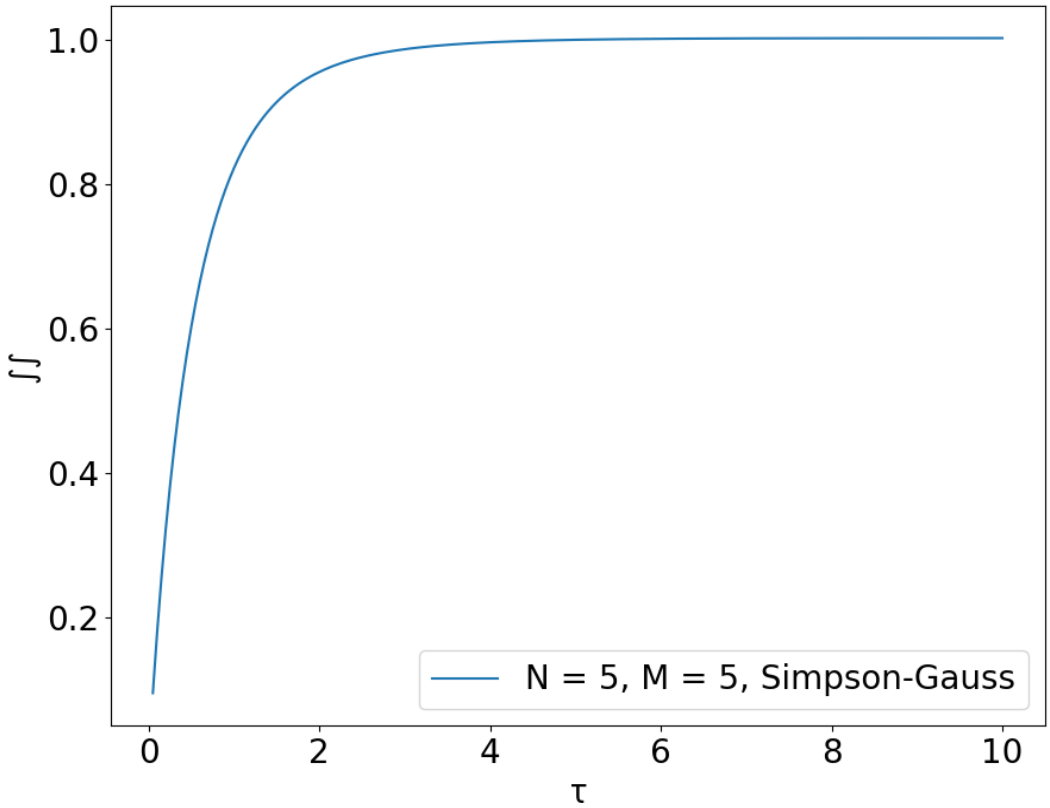


График построен при N = M = 5.

4 Вопросы при защите лабораторной работы

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

Если подынтегральная функция не имеет соответствующих производных, то указанный теоретический порядок точности не достигается. Так, если на отрезке интегрирования не существуют 3-я и 4-я производные, то порядок точности формулы Симпсона будет только 2-ой, $O(n^2)$.

2. Построить формулу Гаусса численного интегрирования при одном узле.

При построении формулы Гаусса необходимо использовать полиномы Лежандра:

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

1. Для n = 1, $P_1(t) = ((1\ d) / (2 * 1!\ dt)\ (t^2 - 1) = t$, корень $t_1 = 0$.
2. Получается система :

   $A_1 = 2$

   $A_1 t_1 = 0$.
3. $\int_a^b f(x)dx = (b-a)/2 * A_1 f(x_1)$, $x_1 = (b - a)/2 * t_1 + (b + a)/2$. Тогда

   результат равен (b - a) f((b+a) / 2).

Ответ: (b - a) f((b+a) / 2).

3. Построить формулу Гаусса численного интегрирования при двух узлах.

При построении формулы Гаусса необходимо использовать полиномы Лежандра:

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

1. Для n = 2, $P_1(t) = ((1\ d^2) / (4 * 2!\ dt^2)\ (t^2 - 1)^2 = ½\ (3t^2 - 1)$.

   Корни: $t_1 = -\sqrt{1/3}$, $t_2 = \sqrt{1/3}$.
2. Получается система :

   $A_1 + A_2 = 2$

   $A_1 t_1 + A_2 t_2 = 0$

   В результате: $A_1 = A_2 = 1$.
4. $\int_a^b f(x)dx = (b-a)/2 * (A_1 f(x_1) + A_2 f(x_2))$, $x_i = (b - a)/2 * t_i + (b + a)/2$. Тогда результат равен $(b - a)/2 * (f((b+a) / 2 - (b- a) / 2\sqrt{3}) + f((b+a) / 2 + (b- a) / 2\sqrt{3}))$.

Ответ: $(b - a)/2 * (f((b+a)/2 - (b- a)/2\sqrt{3}) + f((b+a)/2 + (b- a)/2\sqrt{3}))$.

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.

Пусть областью интегрирования является прямоугольник со сторонами (b-a) и (d-c). Разобьем каждую сторону данного прямоугольника на N и M интервалов, т.е. введем на каждой границе области сетку с N+1 и M+1 узлами. Соответствующие шаги сетки $h_x = (b - a)/N$, $h_y = (d - c)/M$.

Тогда:

$$\int_c^d \int_a^b f(x, y)dxdy = \int_a^b F(x)dx = h_x(½\, F(x_0) + F(x_1) + ½\, F(x_2)) =$$

$$= h_x h_y(½\, (½\, f(x_0;y_0) + f(x_0;y_1) + ½\, f(x_0;y_2)) + ½\, f(x_1;y_0) + f(x_1;y_1) + ½\, f(x_1;y_2) + ½\, (½\, f(x_2;y_0) + f(x_2;y_1) + ½\, f(x_0;y_0))).$$