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 **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 Программная инженерия**

О Т Ч Е Т

по лабораторной работе № 5

Название: **Построение и программная реализация алгоритмов численного интегрирования.**

Дисциплина: Вычислительные алгоритмы

Студент	ИУ7-46Б (Группа)	(Подпись, дата)	Нгуен Ань Тхы (И.О. Фамилия)
Преподаватель		(Подпись, дата)	Градов В.М. (И.О. Фамилия)

Москва, 2020

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

I. Исходные данные.

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.

II. Результат работы программы.

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

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

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

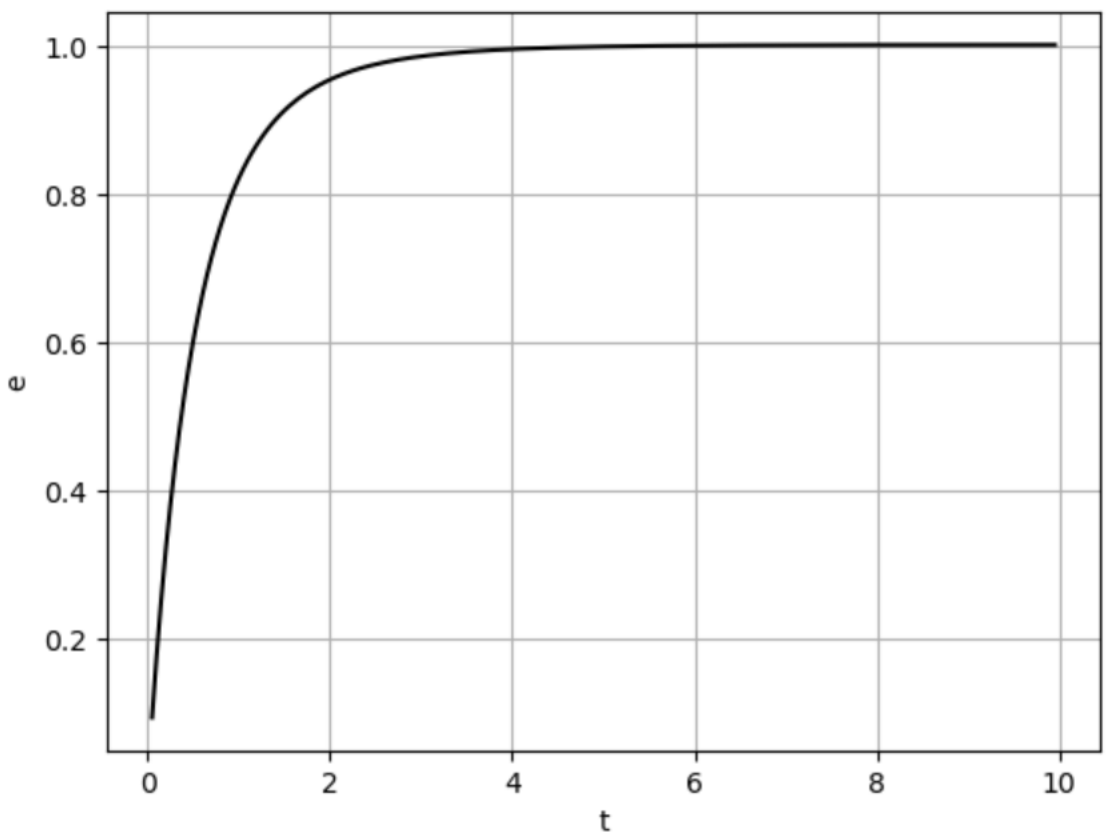
III. Пример выполнения прграммы:

```
Input N: 3
Input M: 1
Input t: 1
result:  0.8883007737422615
>>>
====== RESTART: D:\2019_2020_2_
Input N: 3
Input M: 2
Input t: 1
result:  0.8890043702642584
>>>
====== RESTART: D:\2019_2020_2_
Input N: 3
Input M: 3
Input t: 1
result:  0.8886948534129958
>>>
====== RESTART: D:\2019_2020_2_
Input N: 3
Input M: 4
Input t: 1
result:  0.8888050660353766
>>>
```

```
Input N: 3
Input M: 5
Input t: 1
result:  0.8887830243938746
>>>
====== RESTART: D:\2019_2020_2_
Input N: 5
Input M: 5
Input t: 1
result:  0.8220079359661656
>>>
====== RESTART: D:\2019_2020_2_
Input N: 7
Input M: 5
Input t: 1
result:  0.8166642919024298
>>>
```

При увеличении М результат почти совпадает, что говорит нам о том, что больше влияние оказывает точность внешнего интегрирования.

график зависимости $\varepsilon(\tau)$ **в диапазоне изменения** τ **=0.05-10:**



N = N = 5. Как и было сказано в физическом содержание задачи степень черноты не может быть больше 1.

IV. Описание алгоритма:

1. Квадратная формула Гаусса:

Пусть интервал вычисляется на стандартном интервале [-1;1]. Квадратная формула:

$\int_{-1}^{1} f(t)dt = \sum_{i=1}^{n} A_i f(t_i)$ (5.1)

Запишем полином в виде $f(t) = \sum_{k=0}^{2n-1} a_k t^k$

$\int_{-1}^{1} t^k dt = \sum_{i=1}^{n} A_i {t_i}^k$ k = 0, 1, 2…, 2n – 1 (5.2)

Итак, согласнр (5.2) коэффициенты A_i и узлы t_i находятся из системы 2n уравнений

$$\begin{cases} \sum_{i=1}^{n} A_i = 2 \\ \sum_{i=1}^{n} A_i t_i = 0 \\ \sum_{i=1}^{n} A_i {t_i}^2 = \frac{2}{3} \\ \dots \\ \sum_{i=1}^{n} A_i {t_i}^{2n-1} = 0 \end{cases} \quad (5.3)$$

Система (5.30 нелинейная, и ее решение найти довольно трудно. Рассмотрим полиномы Лежандра:

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

Полиномы Лежандра обладают рядом полезных свойств:

1) $P_n(1) = 1$, $P_n(-1) = (-1)^n$, n = 0, 1, 2…

2) $\int_{-1}^{1} P_n(x) P_m(x) dz = \delta_{mn} N_m,\ N_m = \frac{2}{2m+1}$

3) Полином Лежандра $P_n(x)$ имеет n различных и действительных корней, расположенных на интервале [-1, 1] (5.4)

4) Справедливо рекуррентное соотношение

$P_m(x) = \frac{1}{m} [(2m-1) x P_{m-1}(x) - (m-1) P_{m-2}(x)]$ (5.5)

При вычислении интеграла на произвольном интервале [a;b] т.е $\int_a^b f(x)dx$ для применения квадратурной формулы Гаусса необходимо выпольнить преобразование переменной, получим:

$\int_a^b f(x)dx = \frac{b-a}{2} \sum_{i=1}^{n} A_i f(x_i)$ (5.6) где: $x_i = \frac{b+a}{2} + \frac{b-a}{2} t_i$, i = 1, 2,…n (5.7)

2. Другие формулы численного интегрирования:

F $= \int_a^b f(x)dx$ (5.8)

Построена формула Симпсона:

$\int_a^b f(x)dx = \frac{b-a}{6}[f(x_0) + 4f(x_1) + f(x_2)]$ (5.13)

Рассмотрим интеграл по прямоугольной области:

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

По каждой координате введем сетку узлов. Каждый однократный интеграл вычисляют по квадратным формулам. Конечная формула:

$I = \int_c^d \int_a^b f(x,y)dxdy = \sum_{i=1}^{n} \sum_{j=1}^{m} A_i B_{ij} f(x_i, y_j)$,

где: A_i, B_{ij} – известные постоянные

V. Ответ на вопросы защита:

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

Если подынтегральная функция не имеет соответствующих произвольных. К примеру, если на отрезке интегрирования не существует 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.

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

3. Построить формулу Гаусса численного интегрирования при двух узлах.

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

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.

$h_x = \frac{b-a}{2}$ $h_y = \frac{d-c}{2}$

$\int_c^d \int_a^b f(x,y)dxdy = \int_a^b dx \int_c^d f(x,y)dy = \int_a^b F(x)dx = h_x\left(\frac{1}{2}F_0 + F_1 + \frac{1}{2}F_2\right)$

Где:$F_i(x) = \int_c^d f(x_i, y)dy = h_y\left(\frac{1}{2}(x_i, y_0) + (x_i, y_1) + \frac{1}{2}(x_i, y_2)\right)$

$$\Rightarrow \int_c^d \int_a^b f(x,y)dxdy = h_x h_y \left(\frac{1}{4}(x_0,y_0) + \frac{1}{2}(x_0,y_1) + \frac{1}{4}(x_0,y_2) + \frac{1}{4}(x_1,y_0) + \frac{1}{2}(x_1,y_1) + \frac{1}{4}(x_1,y_2) + \frac{1}{4}(x_2,y_0) + \frac{1}{2}(x_2,y_1) + \frac{1}{4}(x_2,y_2)\right)$$

VI. Код программы:

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

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

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

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

def function(parameter):
    return lambda x, y: (4 / pi) * (1 - exp(-parameter * 2 * cos(x) / (1 - (sin(x) ** 2) * (cos(y) **
2)))) * cos(x) * sin(x)

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

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

    return res

def simpson(func, a, b, amounts):
    if (amounts < 3 or amounts & 1 == 0):
        raise ValueError
    h = (b - a) / (amounts - 1)
    x = a
    res = 0

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

    return res * (h / 3)

def result(func, n, m, tao):
    return simpson(lambda x: gauss(converts(func, x), LIMITS[1][0], LIMITS[1][1], m),
LIMITS[0][0], LIMITS[0][1], n)

def main():
    N = int(input("Input N: "))
    M = int(input("Input M: "))
    tao = float(input("Input t: "))
```

```
    print("result: ", result(function(tao), N, M, tao))

def graph():
    plt.grid(True)
    plt.xlabel("t")
    plt.ylabel("e")
    N = 5
    M = 5
    T = np.arange(0.05, 10, 0.05)
    E = [result(function(tao), N, M, tao) for tao in T]
    plt.plot(T, E, "k")
    plt.show()

main()
graph()
```