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

«Построение и программная реализация алгоритма численного интегрирования»

Студент Маслова Марина Дмитриевна
фамилия, имя, отчество

Группа ИУ7-43Б

Оценка (баллы) _____

Преподаватель Градов Владимир Михайлович
фамилия, имя, отчество

2021 г.

Оглавление

Задание **3**

Описание алгоритма **4**

Код программы **6**

Результат работы **10**

Контрольные вопросы **12**

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

Задание

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

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

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.

Описание алгоритма

В программе используется метод последовательного интегрирования, то есть сначала заданная функция интегрируется по переменной θ, потом по переменной φ. Так как вычисляются определенные интегралы, после первого преобразования получается функция двух переменных (φ, τ), после второго преобразования получается функция от одной переменной τ, вычислением которой при заданном пользователем параметре получается результат.

Так как интегрирование по каждой переменной производится независимо, используются различные методы интегрирования по одной переменной. В данном случае для интегрирования по θ применяется метод Гаусса, а по переменной φ — метод Симпсона.

В методе Гаусса интеграл вычисляется по квадратурной формуле:

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

где t_i — нули полинома Лежандра $P_n(t)$;

A_i определяются из системы:

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

Для подсчета интеграла на произвольном промежутке $[a,\ b]$ производится преобразование переменной:

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

и получается интеграл:

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

В методе Симпсона используется квадратурная формула:

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

Код программы

Код программы представлен на листингах 1-3.

Листинг 1. integration.py

```
"""
    Модуль численного интегрирования
"""

from numpy.polynomial.legendre import leggauss


def simpson(func, left, right, num):
    """
        Интегрирование методом Симпсона
    """
    step = (right - left) / num

    result = 0

    x = left

    while x < right:
        xNext = x + step
        result += func(x) + 4 * func(x + step / 2)  + func(xNext)
        x = xNext

    return step / 6 * result


def toX(t, left, right):
    """
        Преобразование переменной
    """
    return (right + left) / 2 + (right - left) / 2 * t


def gauss(func, left, right, num):
    """
        Интегрирование методом Гаусса
    """
    nodes, coefs = leggauss(num)

    result = 0

    for i in range(num):
        result += coefs[i] * func(toX(nodes[i], left, right))

    return (right - left) / 2 * result


def toTetaFunc(func, tau, phi):
    """
        Получение функции для интегрирования по teta
    """
    return lambda teta : func(tau, phi, teta)


def toPhiFunc(func, tau):
```

```
    """
        Получение функции для интегрирования по tau
    """
    return lambda phi : func(tau, phi)


def getTauFunc(treeArgsFunc, tetaConf, phiConf):
    """
        Последовательное интегрирование
    """
    twoArgsFunc = (
        lambda tau, phi :
        gauss(
            toTetaFunc(treeArgsFunc, tau, phi),
            tetaConf[0][0],
            tetaConf[0][1],
            tetaConf[1]
        )
    )

    oneArgFunc = (
        lambda tau :
        simpson(
            toPhiFunc(twoArgsFunc, tau),
            phiConf[0][0],
            phiConf[0][1],
            phiConf[1]
        )
    )

    return oneArgFunc
```

Листинг 2. graphics.py

```
"""
    Модуль отображения графиков
"""

import matplotlib.pyplot as plt


def getPointFunc(func, xRange):
    """
        Формирование списков точек функции
        в заданном диапазоне
    """
    step = (xRange[1] - xRange[0]) / 100

    xData = []
    yData = []

    xCur = xRange[0]

    while xCur < xRange[1] + step:
        xData.append(xCur)
        yData.append(func(xCur))
        xCur += step

    return xData, yData
```

```
def getGraph(func, xRange, N, M):
    """
        Добавление графика на изображение
    """
    xList, yList = getPointFunc(func, xRange)

    plt.plot(xList, yList, label="при N = {:d}, M = {:d}".format(N, M))


def show():
    """
        Отображение графиков
    """
    plt.gcf().canvas.set_window_title("Численное интегрирование")
    plt.legend()
    plt.title("ε(τ)")
    plt.show()
```

Листинг 3. main.py

```
"""
    Модуль для запуска программы
    ЛАБОРАТОРНАЯ РАБОТА №5
    ПОСТРОЕНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ
    АЛГОРИТМОВ ЧИСЛЕННОГО ИНТЕГРИРОВАНИЯ
"""

from math import sin, cos, exp, pi
import matplotlib.pyplot as plt

import integration
import graphics

subFunc = (lambda phi, teta :
           2 * cos(teta) / (1 - sin(teta) ** 2 * cos(phi) ** 2))
fullFunc = (lambda tau, phi, teta :
            4 / pi * (1 - exp(-tau * subFunc(phi, teta)))
            * cos(teta) * sin(teta))


if __name__ == "__main__":
    notEnd = True

    while notEnd:
        try:
            N = int(input("Введите количество узлов по θ: "))
            if N < 1:
                raise TypeError

            M = int(input("Введите количество узлов по φ: "))
            if M < 1:
                raise TypeError

            tau = float(input("Введите параметр τ: "))

        except ValueError:
            print("\nНечисловые данные недопустимы!")

        except TypeError:
            print("\nКоличество узлов -- натуральное число!")
```

```
        else:
            tetaRange = [0, pi / 2]
            phiRange = [0, pi / 2]
            tauFunc = integration.getTauFunc(fullFunc, (tetaRange, N),
(phiRange, M))
            print("Вычисленное значение интеграла:", tauFunc(tau))

            graphics.getGraph(tauFunc, [0.05, 10], N, M)

            notEnd = input("\nДля выхода нажмите Enter"
                           + "\nДля продолжения введите что-либо\n")

    graphics.show()
```

Результат работы

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

Полином Лежандра имеет n различных действительных корней, которые расположены на интервале [-1, 1]. В зависимости от степени n полином Лежандра является либо четной, либо нечетной функцией, то есть из того, что x— корень, следует, что $-x$ — тоже корень. Таким образом, при поиске корней можно исследовать только промежуток [0, 1].

Конкретно каждый корень можно искать методом половинного деления, проходя по промежутку [0, 1] с маленьким шагом. На каждом шаге проверять знаки функции на концах интервала, если они различны, то на интервале есть корень, который и находится методом половинного деления, если функция с заданной точностью в одном из концов равна 0, то корень найден без дополнительных действий, если знаки одинаковы — корней на заданном интервале нет, так как случай кратных корней не учитывается в силу свойств полинома Лежандра.

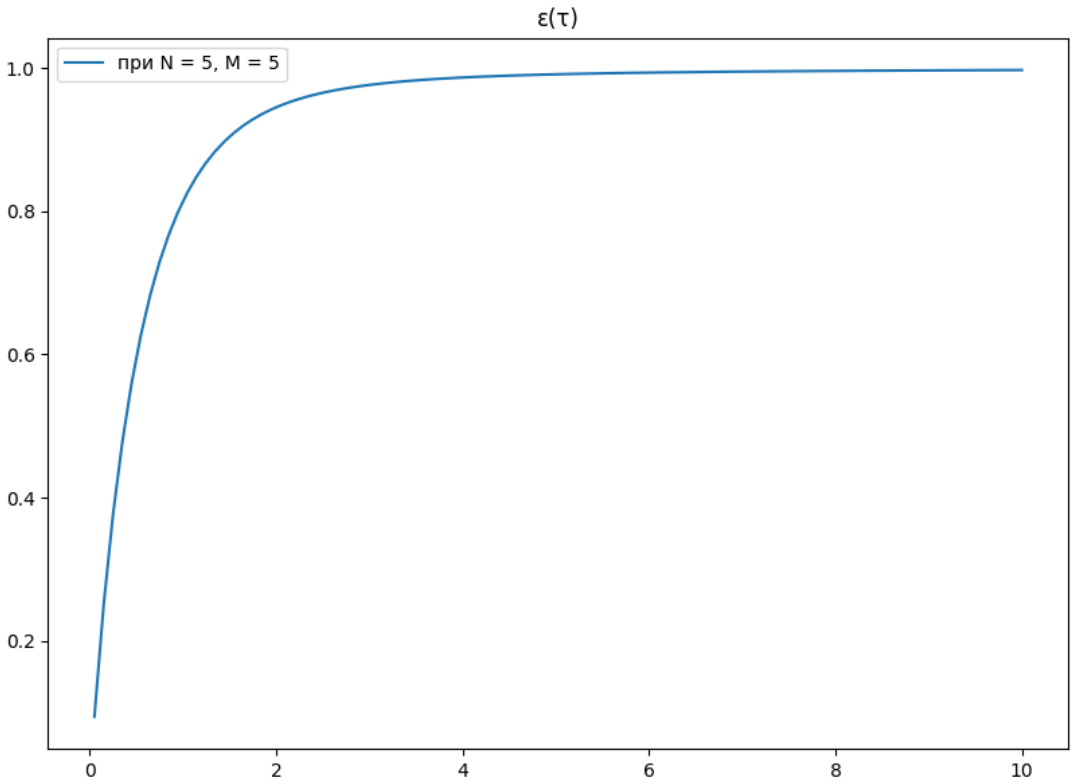
2. Исследовать влияние количества выбираемых узлов сетки по каждому направлению на точность расчетов.

Для исследования с помощью программы получим значения интеграла при значении аргумента $\tau = 1$:

N\M	1	2	3	4	5
1	1.333	1.333	1.333	1.333	1.333
2	0.761	0.770	0.768	0.768	0.768
3	0.806	0.813	0.813	0.812	0.812
4	0.809	0.814	0.814	0.814	0.814
5	0.809	0.814	0.814	0.814	0.814

По таблице делаем вывод, что при одном узле вычисления крайне неточны. При этом увеличение количества узлов в методе Гаусса более значительно увеличивает точность, нежели увеличение количества узлов в методе Симпсона, которое может приводить и к понижению точности $(N = 3)$

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



Как и следовало ожидать, при увеличении τ увеличивается ε, при этом $\varepsilon < 1$, что соответствует физическому смыслу данной величины.

Контрольные вопросы

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

Асимптотическая погрешность формул выражается через интеграл производной функции определенного порядка, при этом если подынтегральная функция не имеет соответствующих производных, то указанный теоретический порядок квадратурной формулы численного интегрирования не достигается. Например, если на отрезке интегрирования не существуют 3-я и 4-я производные, то порядок точность формулы Симпсона 2-ым ($O(h^2)$), а не 4-ым ($O(h^4)$).

2. Построить формулу Гаусса численного интегрирования при одном узле.

$$\int_{-1}^{1} f(t)dt = A_1 f(t_1)$$

$$A_1 = 2$$

$$P_1(x) = x \Rightarrow t_1 = 0$$

$$\int_{-1}^{1} f(t)dt = A_1 f(t_1) = 2\,f(0)$$

Для аргумента в интервале $[a,\ b]$:

$$x_1 = \frac{b+a}{2} + \frac{b-a}{2} t_1 = \frac{b+a}{2}$$

$$\int_{a}^{b} f(x)dx = \frac{b-a}{2}(A_1 f(x_1)) = \frac{b-a}{2} \cdot 2\,f(\frac{b+a}{2}) = (b-a)\,f(\frac{b+a}{2})$$

3. Построить формулу Гаусса численного интегрирования при двух узлах.

$$\int_{-1}^{1} f(t)dt = A_1 f(t_1) + A_2 f(t_2)$$

$$\begin{cases} A_1 + A_2 = 2 \\ A_1 t_1 + A_2 t_2 = 0 \end{cases}$$

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

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

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

Для аргумента в интервале $[a,\ b]$:

$$x_1 = \frac{b+a}{2} + \frac{b-a}{2}t_1 = \frac{b+a}{2} - \frac{b-a}{2}\frac{1}{\sqrt{3}}$$

$$x_2 = \frac{b+a}{2} + \frac{b-a}{2}t_2 = \frac{b+a}{2} + \frac{b-a}{2}\frac{1}{\sqrt{3}}$$

$$\int_a^b f(x)dx = \frac{b-a}{2}(A_1 f(x_1) + A_2 f(x_2)) = \frac{b-a}{2}(f(\frac{b+a}{2} - \frac{b-a}{2}\frac{1}{\sqrt{3}}) + f(\frac{b+a}{2} + \frac{b-a}{2}\frac{1}{\sqrt{3}}))$$

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.

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

$$= h_x h_y[\frac{1}{2}(\frac{1}{2}f(x_0,y_0) + f(x_0,y_1) + \frac{1}{2}f(x_0,y_2)) + \frac{1}{2}f(x_1,y_0) + f(x_1,y_1) + \frac{1}{2}f(x_1,y_2) +$$

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

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