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 Пересторонин Павел

Группа ИУ7-43Б

Оценка (баллы) ____________________

Преподаватель Градов В. М.

Москва.
2020 г.

Техническое задание

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

$$\varepsilon(\tau)=\frac{4}{\pi}\int_{0}^{\pi/2} d\phi \int_{0}^{\pi/2} [1-\exp(-\tau \frac{l}{R})] \cos\theta \, \sin\theta \, d\theta \, d\phi \, ,$$

где $\frac{l}{R}=\frac{2\cos\theta}{1-\sin^2\theta \cos^2\phi}$, θ, ϕ - углы сферических координат.

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

Исходный код алгоритмов.

```python
# a = start, b = end
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)


#converts to args func to one arg func
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


# func = f(x, y) - function which we integrate
# limits = [[a, b], [c, d]] - limits for x and y
# num_of_nodes = [N, M] - number of nodes for x and y
# integrators = [func1, func2] - functions, which we use to integrate: f(func, a, b, num_of_nodes)
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])
```

Интерфейс работы.

```
Enter N: 5
Enter M: 5
Enter param (tao): 1
Enter external method (0 - Gauss; 1 - Simpson): 0
Enter internal method (0 - Gauss; 1 - Simpson): 1
Result with your parameter: 0.8143810072256764
qt5ct: using qt5ct plugin
End? (0 - No, 1 - Yes):
```

Сначала программа спрашивает параметры сетки (Кол-во узлов N, M)
Затем указывается значение параметра тао.
Потом указываются методы, которые требуется использовать в разных направлениях, так как одно интегрирование в одном направлении используется внутри интегрирования в другом направлении, я называю их "внутреннее" и "внешнее":

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

где

$$F(x) = \int_c^d f(x, y) dy \, .$$

В данном случае интеграл I мы ищем как интеграл от функции одного переменного F(x), но функция F для какого-то определенного значения x есть интеграл от функции одного переменного f(x, y) (потому что x в данном случае есть константа). Таким образом для I используется "внешнее" интегрирование, а для каждого значения F(x) = $\int f(x, y) dy$ - "внутреннее".

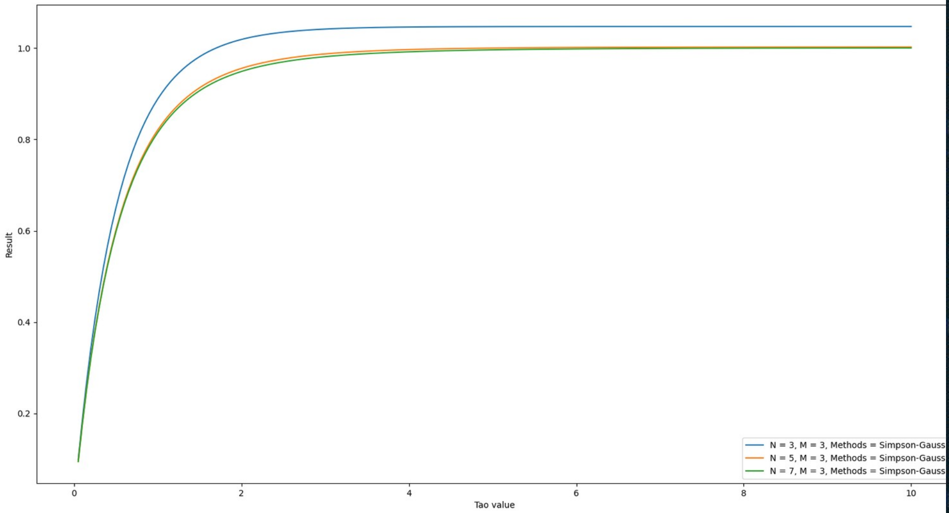
Результаты.

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

В математике доказывается, что производная четной функции есть нечетная функция и наоборот, производная нечетной функции есть функция четная. При составлении функции полинома Лежандра берется n-ная производная от четной функции, что в результате дает либо четную, либо нечетную функцию. Это дает нам следущее соотношение:
f(x) = 0 => f(-x) = 0 (из свойств четных и нечетных функций)
Нам известно, что 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] (довольно очевидно, что если полином нечетной степени, то будет еще корень 0).
В моем случае поиск корней осуществлялся методом 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 (ищем его методом половинного деления). Если один из концов отрезка = 0 - мы нашли корень. Таким образом находятся все корни полинома.

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

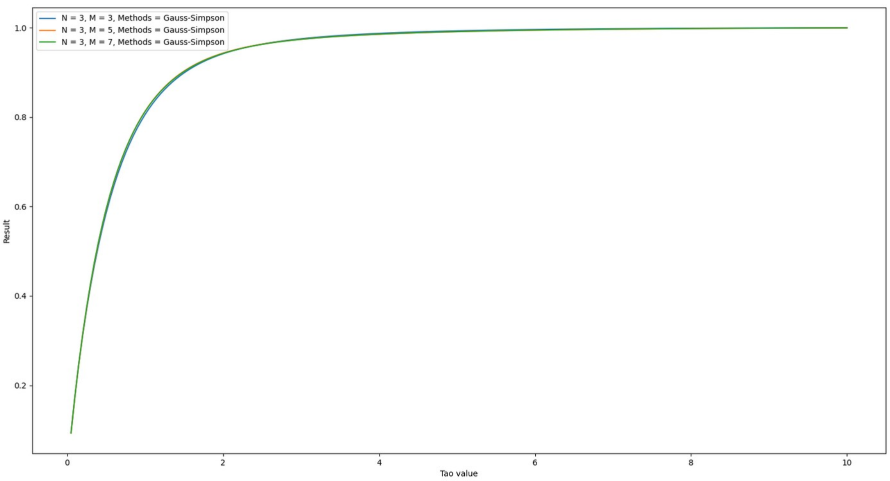
Для начала исследование для внешнего направления и метода Симпсона:




```
Enter N: 3
Enter M: 3
Enter param (tao): 1
Enter external method (0 - Gauss; 1 - Simpson): 1
Enter internal method (0 - Gauss; 1 - Simpson): 0
Result with your parameter: 0.8886948534129957
qt5ct: using qt5ct plugin
End? (0 - No, 1 - Yes): 0
Enter N: 5
Enter M: 3
Enter param (tao): 1
Enter external method (0 - Gauss; 1 - Simpson): 1
Enter internal method (0 - Gauss; 1 - Simpson): 0
Result with your parameter: 0.8208551739111585
End? (0 - No, 1 - Yes): 0
Enter N: 7
Enter M: 3
Enter param (tao): 1
Enter external method (0 - Gauss; 1 - Simpson): 1
Enter internal method (0 - Gauss; 1 - Simpson): 0
Result with your parameter: 0.8152779841066683
```

Видно, что при кол-ве узлов = 3 график отклоняется от графиков с кол-вом узлов 5 и 7 (график для кол-ва узлов 17 так же совпал с 5 и 7, однако здесь не приведен; цвета и функции можно определить в правом нижнем углу). Сравнивая с программой расчета интегралов (Вольфрам, справа можно посмотреть значения: правильное = 0.814...), можно убедиться, что при большом узлов результат точнее, при этом при кол-ве узлов 5+ результаты относительно близки друг к другу (кол-во узлов для внутреннего направления оставалось постоянным).

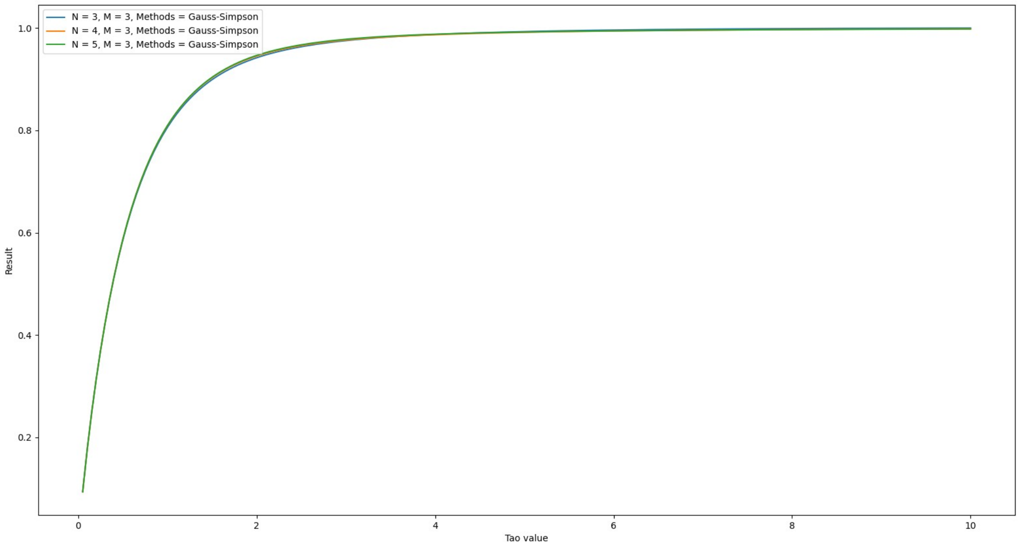
Исследование для внутреннего направления и метода Симпсона:




```
Enter N: 3
Enter M: 3
Enter param (tao): 1
Enter external method (0 - Gauss; 1 - Simpson): 0
Enter internal method (0 - Gauss; 1 - Simpson): 1
Result with your parameter: 0.8057719561370532
qt5ct: using qt5ct plugin
End? (0 - No, 1 - Yes): 0
Enter N: 3
Enter M: 7
Enter param (tao): 1
Enter external method (0 - Gauss; 1 - Simpson): 0
Enter internal method (0 - Gauss; 1 - Simpson): 1
Result with your parameter: 0.8126077758469115
End? (0 - No, 1 - Yes): 0
Enter N: 3
Enter M: 17
Enter param (tao): 1
Enter external method (0 - Gauss; 1 - Simpson): 0
Enter internal method (0 - Gauss; 1 - Simpson): 1
Result with your parameter: 0.8122308865577255
End? (0 - No, 1 - Yes): 1
```

Результаты практически совпадают. Это говорит нам о том, что большее влияние оказывает точность внешнего интегрирования.

Исследования для внешнего направления и метода Гаусса:



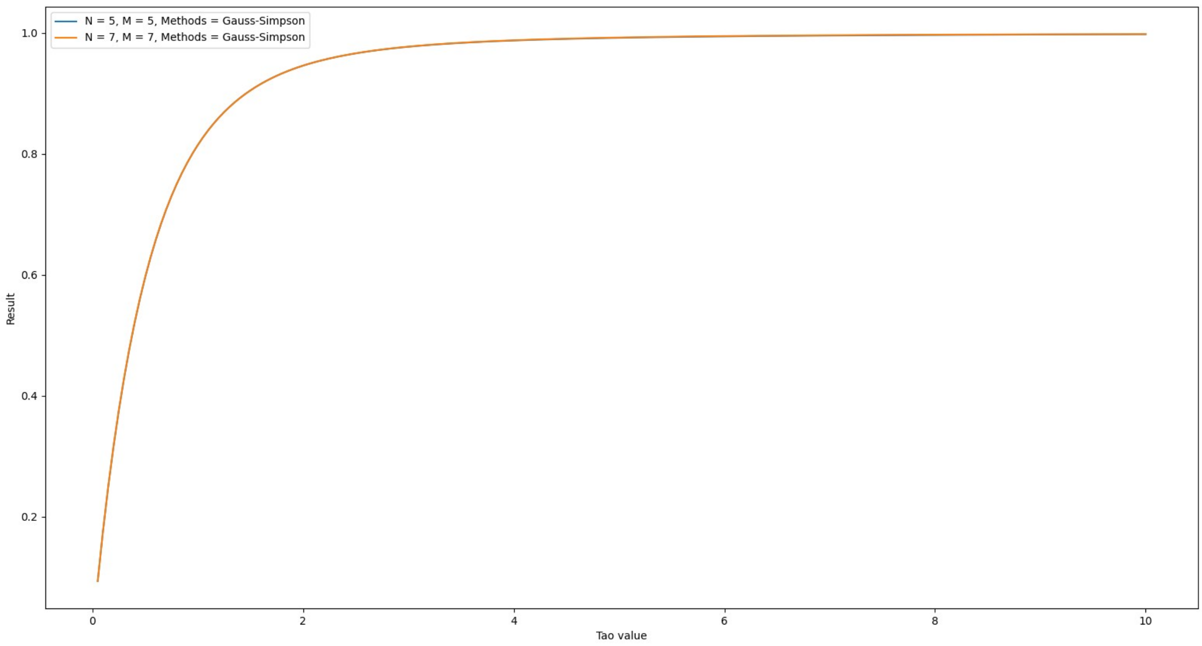

```
Enter N: 3
Enter M: 3
Enter param (tao): 1
Enter external method (0 - Gauss; 1 - Simpson): 0
Enter internal method (0 - Gauss; 1 - Simpson): 1
Result with your parameter: 0.8057719561370532
qt5ct: using qt5ct plugin
End? (0 - No, 1 - Yes): 0
Enter N: 4
Enter M: 3
Enter param (tao): 1
Enter external method (0 - Gauss; 1 - Simpson): 0
Enter internal method (0 - Gauss; 1 - Simpson): 1
Result with your parameter: 0.8091466137324328
End? (0 - No, 1 - Yes): 0
Enter N: 5
Enter M: 3
Enter param (tao): 1
Enter external method (0 - Gauss; 1 - Simpson): 0
Enter internal method (0 - Gauss; 1 - Simpson): 1
Result with your parameter: 0.8094601626173514
End? (0 - No, 1 - Yes): 1
```

Результаты близки к действительному даже при малом кол-ве узлов.
Видно, что Гаусс дает более точные результаты, чем Симпсон на внешнем направлении (на 3 и на 5 результаты несильно отличаются и близки к действительному, в то время как Симпсон на 3 узлах заметно отклонялся)

Очевидно, что на внутреннем направлении Гаусса большую часть в погрешность будет вносить Симпсон на внешнем направлении.

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

Такие графики зависимости построены выше, но здесь приведу еще парочку для кол-ва узлов по направлениям (5; 5) и (7; 7).



Графики наложились друг на друга (близки по значениям), а результат очень близок к действительному:

```
Enter N: 5
Enter M: 5
Enter param (tao): 1
Enter external method (0 - Gauss; 1 - Simpson): 0
Enter internal method (0 - Gauss; 1 - Simpson): 1
Result with your parameter: 0.8143810072256764
qt5ct: using qt5ct plugin
End? (0 - No, 1 - Yes): 0
Enter N: 7
Enter M: 7
Enter param (tao): 1
Enter external method (0 - Gauss; 1 - Simpson): 0
Enter internal method (0 - Gauss; 1 - Simpson): 1
Result with your parameter: 0.814383069793779
End? (0 - No, 1 - Yes): 1
```

Ответы на вопросы к защите.

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

Если подинтегральная 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, $O(h^2)$.

2. Построить формулу Гаусса численного интегрирования при одном узле.

№2

$$\sum_{i=1}^{n} A_i = 2 \Rightarrow A_1 = 2$$

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

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

$$x = \frac{b+a}{2} + \frac{b-a}{2}t \quad \text{(аргумент для } [a, b]\text{)}$$

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

3. Построить формулу Гаусса численного интегрирования при двух узлах.

№3

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

$$\begin{cases} A_1 + A_2 = 2 \\ A_1 \underbrace{t_1}_{=-\sqrt{\frac{1}{3}}} + A_2 \underbrace{t_2}_{=\sqrt{\frac{1}{3}}} = 0 \end{cases} \Rightarrow A_1 = A_2 = 0$$

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

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

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.

№4

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

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

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

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

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