|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ДИСЦИПЛИНА «Вычислительные алгоритмы»\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Лабораторная работа № 5**

|  |  |
| --- | --- |
| **Тема «**Построение и программная реализация алгоритмов численного интегрирования»  **Студент**  Кузин А.А.  **Группа** ИУ7 – 42 Б  **Оценка (баллы) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  **Преподаватель** Градов В.М. |  |

Москва.

2020 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 

,

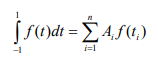
где ,

- углы сферических координат.

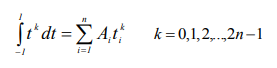
Применить метод последовательного интегрирования. По одному направлению использовать формулу Гаусса, а по другому - формулу Симпсона.

**Описание алгоритма:**

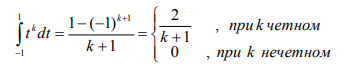
Пусть интеграл вычисляется на стандартном интервале [1; 1]. Задача состоит в том, чтобы подобрать точки t1, t2, ..., tn и коэффициенты A1, A2, ..., An так, чтобы квадратурная формула

****

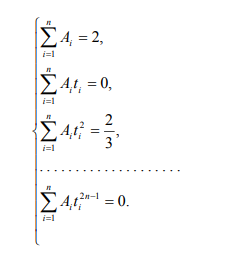
Была точной для всех полиномов наивысшей возможной степени. Установлено, что эта наивысшая степень равна N = 2n − 1 Согласно вышенаписанной формуле:

****

Принимая во внимание, что:

****

Таким образом, коэффициенты Ai и узлы ti находятся из системы 2n уравнений

****

Система нелинейная и найти решения довольно трудно. Для облегчения задачи воспользуемся полиномом Лежандра степени n и найдём его нули. Нули полинома Лежандра буду являться узлами формулы Гаусса. Проще всего найти нули полинома Лежандра, учитывая рекуррентное соотношение

****

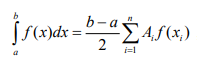
Далее системе решается методом Гаусса, откуда находятся коэффициенты Ai, после чего подставляются в формулу:

****

При вычислении интеграла на произвольном интервале [a; b], т.е.  для применения квадратурной формулы Гаусса необходимо выполнить преобразование переменной

****

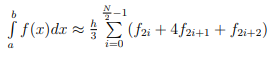
Таким образом, получим:



Где



Квадратурная формула Симпсона :



**Kод:**

def f(fi, teta, tau):

lr = (2 \* cos(teta) / (1 - sin(teta) \*\* 2 - cos(fi) \*\* 2))

return (1 - exp(-tau \* lr)) \* cos(teta) \* sin(teta)

def F(fi, tau, m, h\_y):

res = 0

for i in range(int(m / 2 - 1)):

res += f(fi, 2\*i\*h\_y, tau) + 4 \* f(fi, (2\*i + 1) \* h\_y, tau) + f(fi, (2\*i+2)\*h\_y, tau)

return h\_y / 3 \* res

def roots\_legendre(n):

roots = []

roots\_inter = []

h = 2 / n

a = -1

b = a + h

legendre = form\_legendre(n)

while len(roots\_inter) != n:

roots\_inter = []

while b <= 1:

if legendre.get(a) \* legendre.get(b) < 0:

roots\_inter.append([a, b])

a = b

b += h

h /= 2

a = -1

b = a + h

for i in roots\_inter:

roots.append(bisection(i[0], i[1], 0.000001, legendre.get))

return roots

def matrix\_normalize\_rows(matrix, x\_vars, cur):

i = cur

while i < x\_vars:

normalize = max(matrix[i])

if fabs(normalize) < 1e-6:

i+= 1

continue

j = cur

while j < x\_vars + 1:

matrix[i][j] /= normalize

j += 1

i += 1

i = cur + 1

while i < x\_vars + 1:

if matrix[i][cur] < 1e-6:

i += 1

continue

j = cur

while j < x\_vars + 1:

matrix[i][j] -= matrix[cur][j]

j += 1

i += 1

def matrix\_solve(matrix, x\_vars):

x = [0 for i in range(x\_vars)]

i = x\_vars - 1

while i >= x\_vars - 1:

sigma = 0

j = x\_vars - 1

while j > i:

sigma += matrix[i][j] \* x[j]

j -= 1

if fabs(matrix[i][j]) <= 1e-6:

x[i] = 0

else:

x[i] = (matrix[i][x\_vars] - sigma) / matrix[i][i]

i -= 1

return x

tau = 0.5

n = int(input("Введите n: "))

m = int(input("Введите m: "))

b = pi/2

a = 0

h\_y = (b - a) / n

t = []

y = []

roots = roots\_legendre(n)

matrix = [[pow(roots[j], i) for j in range(n)] for i in range(n)]

for i in range(n):

matrix[i].append((1 - pow(-1, i + 1)) / (i + 1))

for i in range(n):

matrix\_normalize\_rows(matrix, n, i)

A = matrix\_solve(matrix, n)

while tau <= 10:

t.append(tau)

res = 0

for i in range(n):

res += A[i] \* F((b + a) / 2 + (b - a) / 2 \* roots[i], tau, m, h\_y)

res \*= (b - a) / 2

res \*= 4 / pi

y.append(res)

tau += 0.05

**Полученные результаты:**

**Описать алгоритм вычисления n корней полинома Лежандра n-ой степени Pn(x) при реализации формулы Гаусса***:*

Для нахождения корней полинома Лежандра я применил метод половинного деления, опираясь на реккурентное свойство полинома Лежандра:

****

Базой рекурсии в таком случае является:

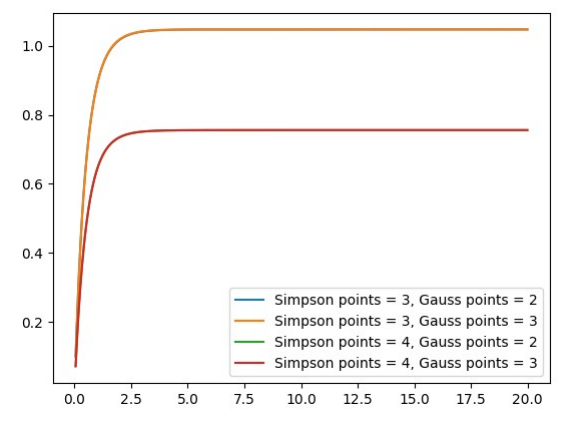
P0(x) = 1

P1(x) = x

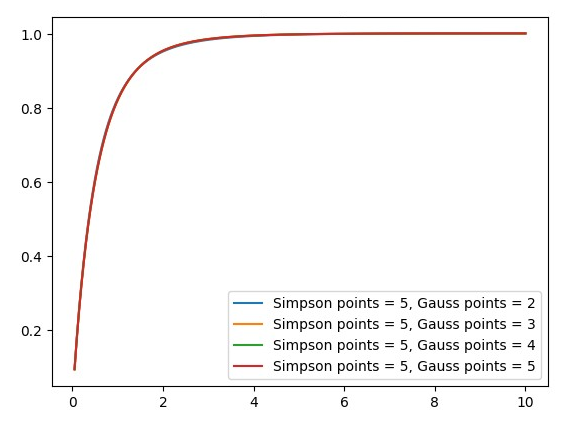
Очевидно, что для применения половинного деления необходимо сначала найти такой отрезок, чтобы на его концах знаки функции были различны (либо чтобы произведение обращалось в нуль), т.е. Pn(x) ∗ Pn(x + step) <= 0. Далее с этими исходными данными применяется метод половинного деления.

**Исследовать влияние количества выбираемых узлов сетки по каждому направлению на точность расчетов:**

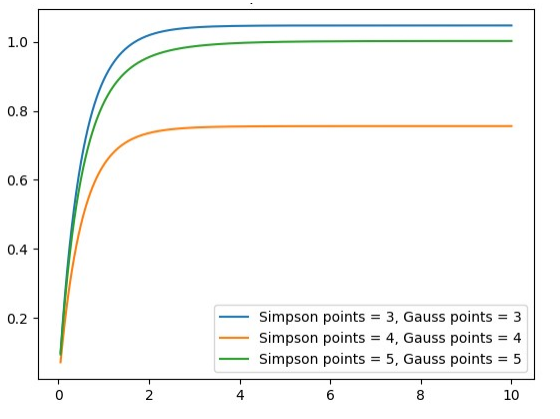
1. Рассмотрим случай для метода Симпсона с меньшим количеством узлов 3 и 4, больший вклад вносит метод являющийся внешним.



1. При задании для формулы Симпсона 5 узлов направление метода Гаусса при разных количествах узлов вносит один и тот же вклад, графики накладываются.

****

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

****

Таким образом, можно сделать вывод, что количество узлов напрямую влияет на точность результатов. Наиболее оптимальное решение, содержащее физический смысл получено с помощью сетки (5, 5).

**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, порядок точности будет равен номеру последней существующей проводной.

**2) Построить формулу Гаусса численного интегрирования при одном узле.**

n = 1

Корень полинома Лежандра первой степени: x = 0

Решив систему, получим: A1 = 2

В итоге

**3) Построить формулу Гаусса численного интегрирования при двух узлах.**

n = 2

Корни полинома Лежандра второй степени: x = ±

Решив систему, получим: A1 = A2 = 1

В итоге

**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.**

Где и