**Белорусский государственный университет**

**Факультет прикладной математики и информатики**

Лабораторная работа №1

по Методам численного анализа

Задание №И-17

Выполнил :

Студент 3 курса 8 группы ФПМИ ТП

Бусыгин Владимир Дмитриевич

Преподаватель :

Радкевич Елена Владимировна

2021

**Описание правила Рунге**

В случае с КФ можно указать такое число разбиений N, при котором погрешность КФ будет . Идея в том, чтобы, по нескольким приближененным значениям искомой величины, вычислить значения главных частей разложения остатка КФ.

- приближенное значение

Вычислим значение интеграла при

- считается, что интеграл вычислим с заданной точности если выполняется неравенство

При проведении расчетов обычно вычисляют интеграл по шагам :

**Описание метода построения КФ левых прямоугольников**

- общая КФ

При , тогда

- простейшая КФ левых прямоугольников

Построим составную КФ левых прямоугольников :

Разобьем [a,b] на N частей , , N - количество разбиений [a,b], N+1 -точна.

Тогда составная КФ левых прямоугольников :

**Описание метода построения КФ трапеций**

- общая КФ

Построим составную формулу :

Из формулы остатка мы можем получить шаг h, для достижения необходимой точности, который требуется по заданию:

**Описание метода построения КФ средних прямоугольников**

- общая КФ

Т.к - середина [a,b], то кратность этого узла при интерполировании = 2, Тогда остаток соответственно формулы :

Построим составную формулу :

Из формулы остатка мы можем получить шаг h, для достижения необходимой точности, который требуется по заданию:

**Описание метода построения КФ НАСТ Гаусса**

- общая КФ

- формула Гаусса

Системой многочленов ортогональных по весу на [-1, 1]

будут являться корни многочлена Лежандра определяемые из соотношения для каждого n.

- коэффициенты в формуле Гаусса

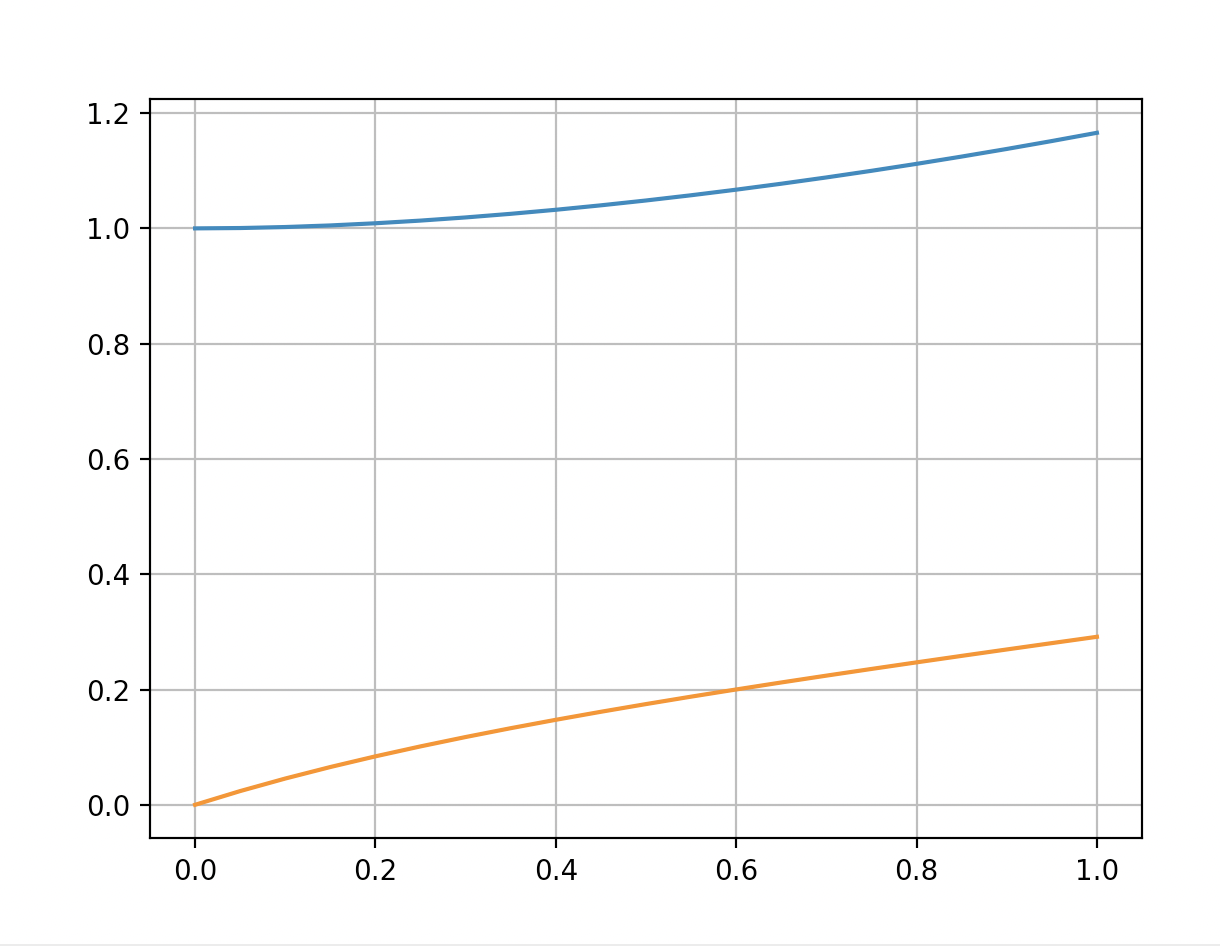
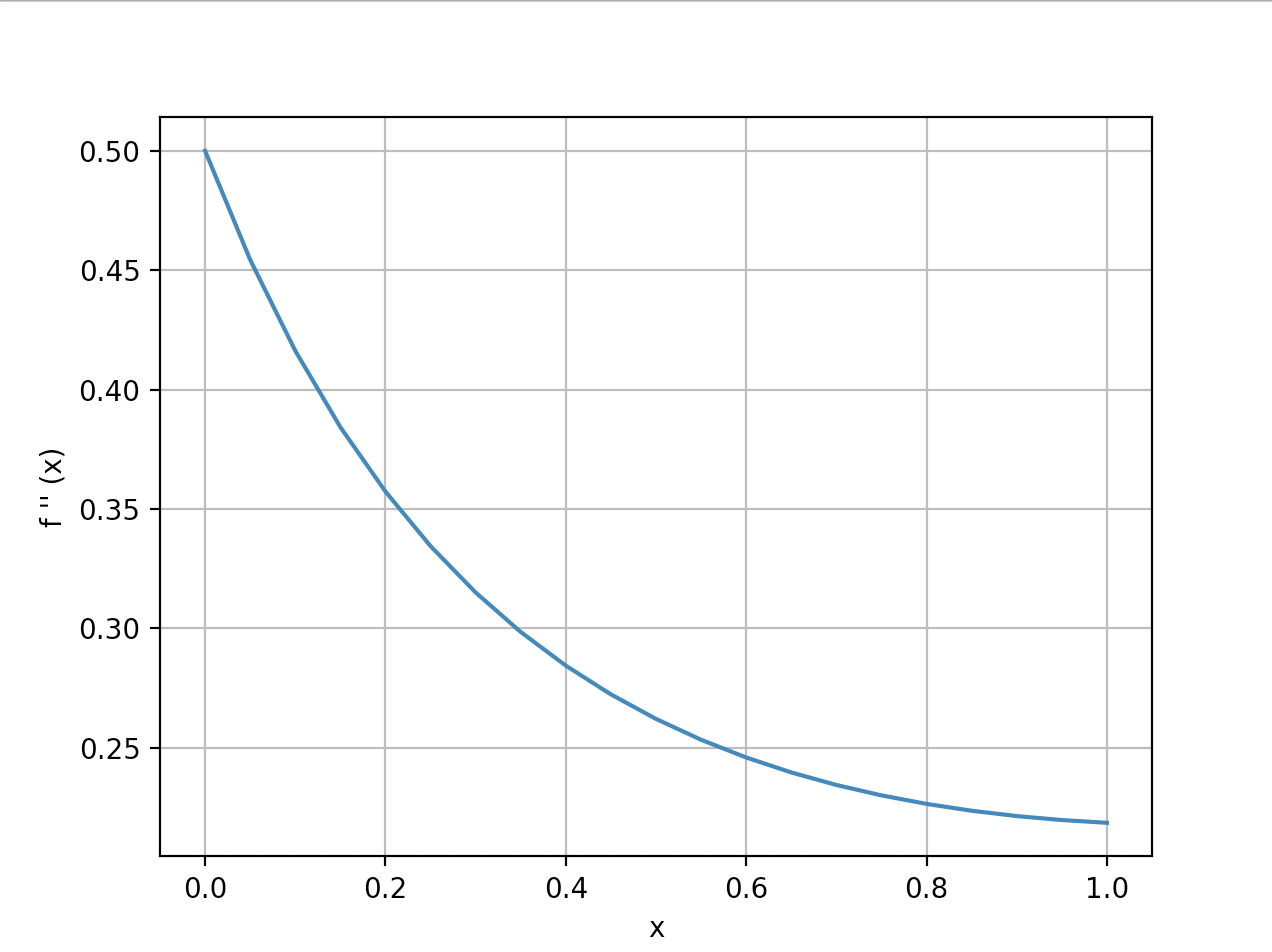
Поскольку исходный интеграл имеет произвольные пределы интегрирования, то для применения формулы Гаусса необходимо пересчитать узлы :

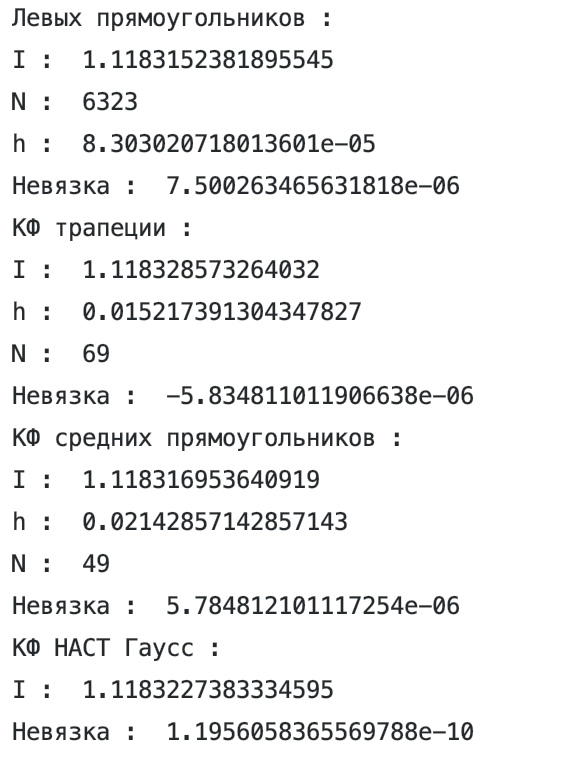
Формула для нахождения остаточного члена :

**Анализ полученных результатов**

**Входные данные :**

**Выходные данные :**

****

****

Для КФ НАСТ Гаусса расчет остаточного члена проводился в wolfram :

**Сравнение методов**

По заданию требовалось получить шаги h, для которых построенные КФ имели бы заданную точность. Посмотрев на величину невязки каждого метода , можно увидеть, что она удовлетворяет требуемому условию : , следовательно КФ применены правильно. Также, можно сравнить шаги и количество разбиений исходного отрезка интегрирования. По результатам можно увидеть, что для КФ левых прямоугольников требуется больше разбиений исходного отрезка интегрирования и как следствие более меньший шаг по сравнению с КФ трапеции и средних прямоугольников. Это связано непосредственно со структурой метода построения КФ, а конкретнее с алгебраической степенью точности, для КФ трапеций и средних прямоугольников АСТ = 1, а у левых прямоугольников на порядок ниже АСТ = 0 и как следствие, для достижения необходимой точности требуется больше разбиений исходного отрезка интегрирования.

В случае с КФ НАСТ Гаусса, мы можем взглянуть на величину невязки и величину остаточного члена, она меньше по сравнению с другими методами. Это связано с тем, что если КФ с

𝑛 узлами является интерполяционной, то ее алгебраическая степень точности не меньше 𝑛 − 1, при любых заданных узлах построение интерполяционной КФ осуществляется за счет выбора ее 𝑛 коэффициентов. За счет же выбора 𝑛 узлов интерполяционной КФ можно добиться того, чтобы она имела возможно более высокую алгебраическую степень точности, а именно 2𝑛 − 1. КФ с 𝑛 узлами, алгебраическая степень точности которой равна 2𝑛−1 и есть КФ НАСТ Гаусса. Следовательно полученная КФ имеет более высокую АСТ по сравнению с КФ трапеций, средних прямоугольников, левых прямоугольников и как следствие более точный результат, что мы и можем наблюдать по величине невязки и остаточного члена.

**Код программы (Python)**

**Задание 1 :**

import matplotlib.pyplot as plt

from numpy import arange, exp

f = lambda x: exp(x/2) / (x + 1)\*\*(1/2)

I\_precise = 1.11832273845302

df = lambda x: exp(x/2) / (2\*(x + 1)\*\*(1/2)) - exp(x/2) / (2\*(x+1)\*\*(3/2))

a = 0

b = 1.05

epsilon = 10\*\*-5

def graphics() -> None:

plt.plot([x for x in arange(a, b, 0.05)], [f(x) for x in arange(a, b, 0.05)],

[x for x in arange(a, b, 0.05)], [df(x) for x in arange(a, b, 0.05)])

plt.grid()

plt.show()

# **TODO :**

# **1)Правило Рунге, сост квадр формула левых прямоугольников, шаг разб h?**

def I(h: float) -> float:

sum = 0

N = int ((b - a) / h)

for k in range(0, N):

sum += f(a + k\*h)

return sum\*h

def R(h1: float, h2: float) -> float:

return h1\*\*2 \* (I(h2) - I(h1)) / (h1\*\*2 - h2\*\*2)

def Runge\_h() -> tuple:

h1 = b

h2 = b/2

N = 2

while abs(R(h1, h2)) > epsilon:

N += 1

h1 = b / N

h2 = h1 / 2

return N, h2

def print\_result() -> None:

N, h = Runge\_h()

print(**'Левых прямоугольников :** \n**I : '**, I(h), **'**\n**N : '**, N, **'**\n**h : '**, h, **'**\n**Невязка : '**, I\_precise - I(h))

**Задание 2 - 3 :**

import matplotlib.pyplot as plt

from numpy import arange, exp

f = lambda x: exp(x/2) / (x + 1)\*\*(1/2)

I\_precise = 1.11832273845302

df = lambda x: exp(x/2) / (2\*(x + 1)\*\*(1/2)) - exp(x/2) / (2\*(x+1)\*\*(3/2))

a = 0

b = 1.05

epsilon = 10\*\*-5

# **TODO : Определить шаги h для заданной точности**

# **2.a) Составная формула трапеций**

# **2.б) Составная формула средних прямоугольников**

ddf = lambda x: (1 - 2/(x + 1) + 3/((x + 1)\*\*2))\*exp(x/2) / (4\*((x + 1)\*\*(1/2)))

M = ddf(a)

def graphic\_ddf() -> None:

plt.plot([x for x in arange(a, b, 0.05)], [ddf(x) for x in arange(a, b, 0.05)])

plt.xlabel(**'x'**)

plt.ylabel(**'f** \'\' **(x)'**)

plt.grid()

plt.show()

def trapeze() -> tuple:

N = int(((((b - a)\*\*3)\*M / (12\*epsilon))\*\*(1/2)))

h = (b - a) / N

return N, h

def I\_trapeze() -> float:

result = (f(a) + f(b)) / 2

N, h = trapeze()

for k in range(1, N):

result += f(a + k\*h)

return result \* h

def middle\_rectangles() -> tuple:

N = int(((((b - a) \*\* 3) \* M) / (24 \* epsilon)) \*\* (1 / 2))

h = (b - a) / N

return N, h

def I\_middle\_rectangles() -> float:

N, h = middle\_rectangles()

result = 0

for k in range(0, N):

result += f(a + (k + 1/2)\*h)

return result \* h

def print\_result() -> None:

N\_t, h\_t = trapeze()

I\_t = I\_trapeze()

N\_m\_r, h\_m\_r = middle\_rectangles()

I\_m\_r = I\_middle\_rectangles()

print(**'КФ трапеции :** \n**I : '**, I\_t, **'**\n**h : '**, h\_t, **'**\n**N : '**, N\_t, **'**\n**Невязка : '**, I\_precise - I\_t)

print(**'КФ средних прямоугольников :** \n**I : '**, I\_m\_r, **'**\n**h : '**, h\_m\_r, **'**\n**N : '**, N\_m\_r, **'**\n**Невязка : '**, I\_precise - I\_m\_r)

**Задание 4 :**

from numpy import exp

from scipy.special import legendre, roots\_legendre

f = lambda x: exp(x/2) / (x + 1)\*\*(1/2)

I\_precise = 1.11832273845302

df = lambda x: exp(x/2) / (2\*(x + 1)\*\*(1/2)) - exp(x/2) / (2\*(x+1)\*\*(3/2))

a = 0

b = 1.05

epsilon = 10\*\*-5

n = 5

x\_new = lambda x\_old: (b - a)\*x\_old/2 + (a+b)/2

# **TODO: Гаусс-НАСТ**

# **1.Применить формулу Гаусса НАСТ**

# **2.Оценить погрешность Rn**

legendre\_x = roots\_legendre(n=n + 1, mu=False)[0] # корни Pn+1(x)

dif\_poly\_legendre\_coefficients = legendre(n=n + 1).deriv() #коэф dPn+1(x):

def A(k: int) -> float:

return 2 / ( (1 - legendre\_x[k]\*\*2)\*(dif\_poly\_legendre\_coefficients(legendre\_x[k])\*\*2) )

def I\_Gauss() -> float:

result = 0

for i in range(0, n + 1):

result += A(i)\*f(x\_new(legendre\_x[i]))

return result\*(b-a)/2

def print\_result() -> None:

print(**'КФ НАСТ Гаусс :** \n**I : '**, I\_Gauss(), **'**\n**Невязка : '**, I\_precise - I\_Gauss())