**Министерство образования и науки Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего профессионального образования**

**«Московский государственный технический университет имени Н.Э. Баумана»**

**(МГТУ им. Н.Э.Баумана)**

**Мытищинский филиал**

**ФАКУЛЬТЕТ КОСМИЧЕСКИЙ**

**КАФЕДРА К-1 САУ**

**ЛАБОРАТОРНАЯ РАБОТА №8**

**ПО ДИСЦИПЛИНЕ**

**“Численные методы”**

**НА ТЕМУ:**

**«Приближенное решение определённых интегралов»**

***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***

**Студент К1-61Б**  **01.04.24 Тимофеев К. А.**

(Группа) (Подпись, дата) (ФИО)

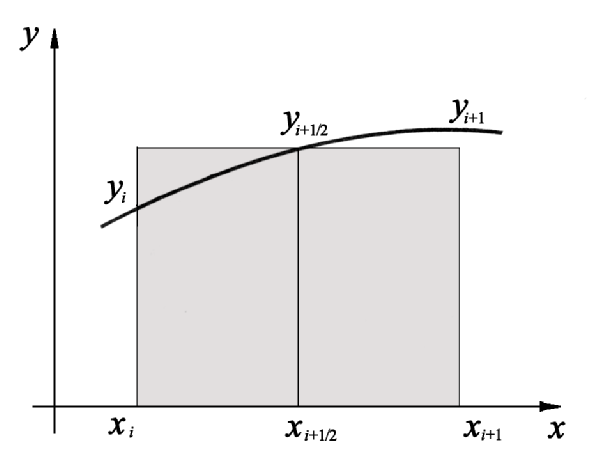
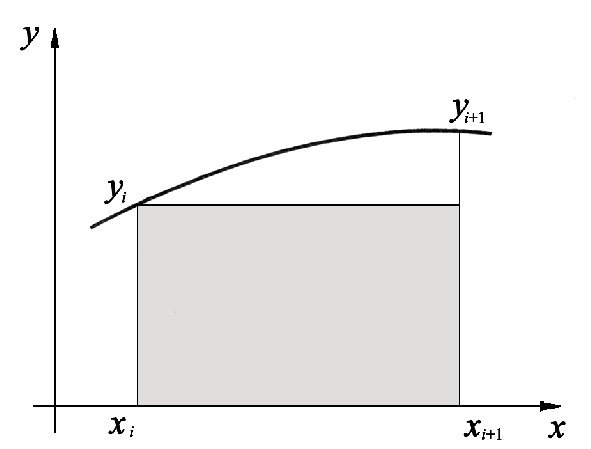
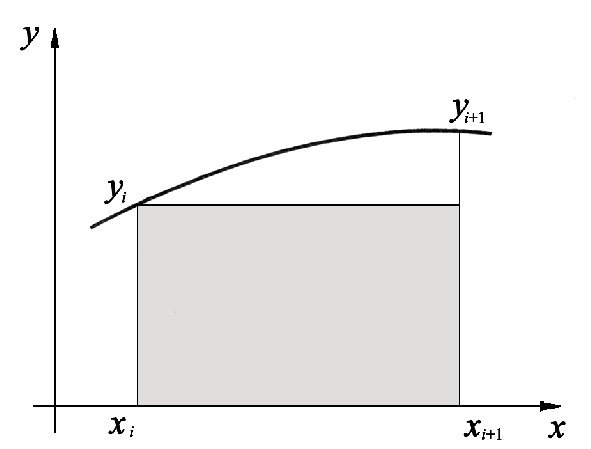
**Руководитель Чернова Т.В**

(Подпись, дата) (ФИО)

*2024 г.*

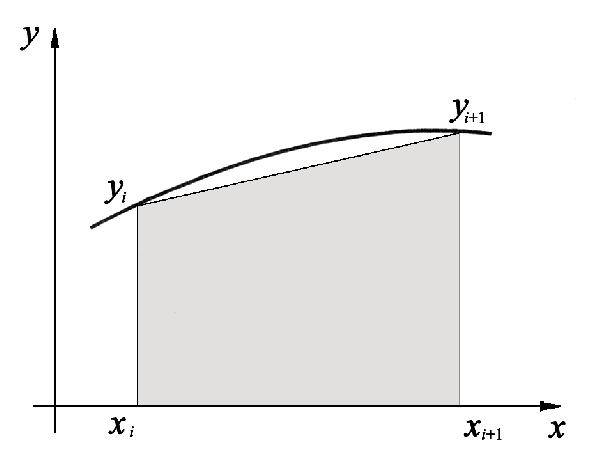
# Теория

## Методы прямоугольников



# 

## Метод трапеций





## Метод Симпсона



# Задачи

1. Вычислить определённый интеграл  левой, правой и центральной формулой прямоугольников.
2. Обеспечить точность вышеуказанного решения в 0,001.
3. Вычислить определённый интеграл  формулой трапеций.
4. Вычислить определённый интеграл  формулой трапеций.
5. Вычислить определённый интеграл  формулой Симпсона.
6. Вычислить определённый интеграл  формулой Симпсона.
7. Составить сравнительную таблицу результатов решения, полученных в 1,3,5.
8. Составить сравнительную таблицу результатов решения, полученных в 2,4,6.

# Ход работы

Напишем функции для решения интегралов указанными методами.

# ПРЯМОУГОЛЬНИКИ:

def rectanglesLeft(function, start, end, n):

    step = abs(end-start)/n

    return step\*sum([function(start + step\*i) for i in range(0, int(abs(end-start)/step))])

def rectanglesRight(function, start, end, n):

    step = abs(end-start)/n

    return step\*sum([function(end - step\*i) for i in range(0, int(abs(end-start)/step))])

def rectanglesCenter(function, start, end, n):

    step = abs(end-start)/n

return step\*sum([(function(start + (i-1)\*step) + function(start + i\*step))/2 for i in range(1, int(abs(end-start)/step)+1)])

# ТРАПЕЦИИ:

def trapeze(function, start, end, n):

    step = abs(end-start)/n

    Y = [function(start+i\*step) for i in range(0, n+1)]

return step\*(sum(Y[1:len(Y)-1:1]) + 0.5\*(Y[0]+Y[len(Y)-1]))

# СИМПСОН:

def simpson(function, start, end, n):

    step = abs(end-start)/n

    Y = [function(start+i\*step) for i in range(0, n+1)]

    return (step/3)\*(Y[0] + Y[len(Y)-1] + 2\*sum(Y[2:len(Y)-1:2]) + 4\*sum(Y[1:len(Y)-1:2]))

# Вспомогательная функция поиска решения по заданной точности epsilon:

def Accuracy( method, function, start, end, n, epsilon):

    In = method(function, start, end, n)

    I2n = method(function, start, end, 2\*n)

    if(abs(In-I2n)<=epsilon): return n, In

    else: return Accuracy( method, function, start, end, 2\*n, epsilon)

#Начальные условия:

def f(x):

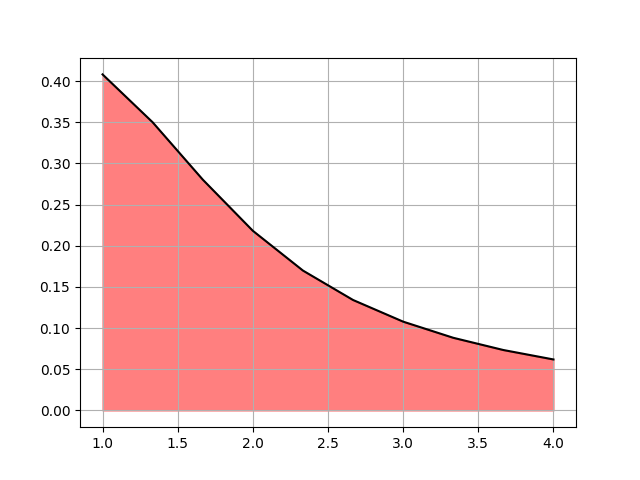
    return round(1 / np.sqrt(x\*\*4+5), 5)

def variantF(x):

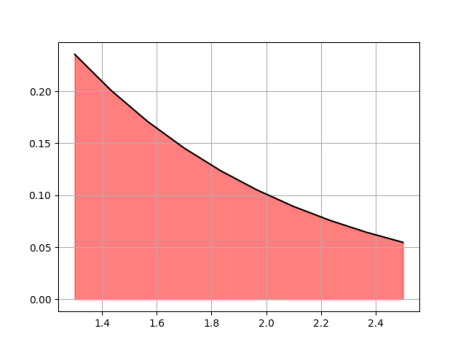
    return round(np.exp(-x) / np.sqrt(0.2\*x\*\*2 + 1), 5)

a = 1; b = 4; n = 10; eps = 0.001

Укажем графически искомые площади под графиками.









Задача 1 и 2

nLeft, resultLeft = Accuracy(rectanglesLeft, f, a, b, n, eps)

nCenter, resultCenter = Accuracy(rectanglesCenter, f, a, b, n, eps)

nRight, resultRight = Accuracy(rectanglesRight, f, a, b, n, eps)

print(f"#----------------------------- TASK[1&2] Rectangle formulas -----------------------------#")

print(f"#------------f(x) = 1 / sqrt(x^4+5); | a = {a}; b = {b}; n = {n}; ε = {eps}; -----------#")

print(tab.tabulate(pd.DataFrame({

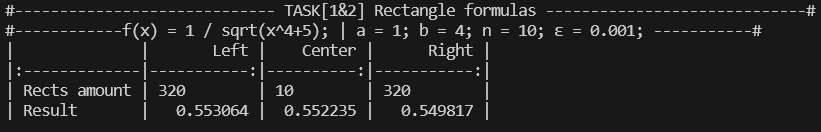
    "": ["Rects amount", "Result"],

    "Left" : [nLeft, resultLeft],

    "Center" : [nCenter, resultCenter],

    "Right" : [nRight, resultRight]

}), tablefmt='pipe', headers='keys', showindex=False))



## Задача 3

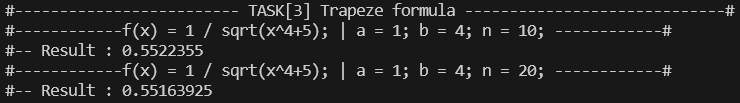
print(f"#------------------------- TASK[3] Trapeze formula -----------------------------#")

print(f"#------------f(x) = 1 / sqrt(x^4+5); | a = {a}; b = {b}; n = {n}; ------------#")

print(f"#-- Result : {trapeze(f, a, b, n)}")

print(f"#------------f(x) = 1 / sqrt(x^4+5); | a = {a}; b = {b}; n = {2\*n}; ------------#")

print(f"#-- Result : {trapeze(f, a, b, 2\*n)}")



## Задача 4

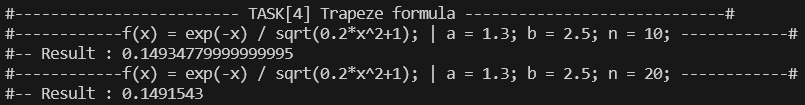
print(f"#------------------------- TASK[4] Trapeze formula -----------------------------#")

print(f"#------------f(x) = exp(-x) / sqrt(0.2\*x^2+1); | a = {1.3}; b = {2.5}; n = {n}; ------------#")

print(f"#-- Result : {trapeze(variantF, 1.3, 2.5, n)}")

print(f"#------------f(x) = exp(-x) / sqrt(0.2\*x^2+1); | a = {1.3}; b = {2.5}; n = {2\*n}; ------------#")

print(f"#-- Result : {trapeze(variantF, 1.3, 2.5, 2\*n)}")



## Задача 5

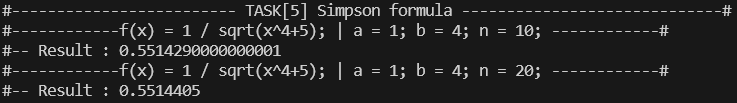
print(f"#------------------------- TASK[5] Simpson formula -----------------------------#")

print(f"#------------f(x) = 1 / sqrt(x^4+5); | a = {a}; b = {b}; n = {n}; ------------#")

print(f"#-- Result : {simpson(f, a, b, n)}")

print(f"#------------f(x) = 1 / sqrt(x^4+5); | a = {a}; b = {b}; n = {2\*n}; ------------#")

print(f"#-- Result : {simpson(f, a, b, 2\*n)}")



## Задача 6

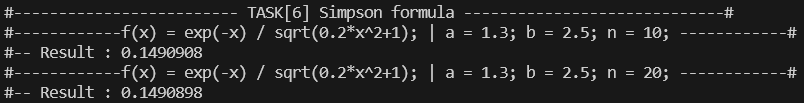
print(f"#------------------------- TASK[6] Simpson formula -----------------------------#")

print(f"#------------f(x) = exp(-x) / sqrt(0.2\*x^2+1); | a = {1.3}; b = {2.5}; n = {n}; ------------#")

print(f"#-- Result : {simpson(variantF, 1.3, 2.5, n)}")

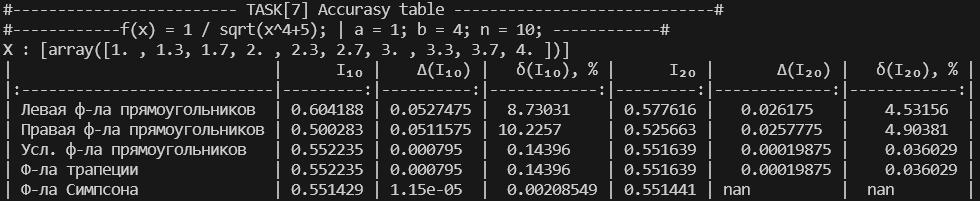
print(f"#------------f(x) = exp(-x) / sqrt(0.2\*x^2+1); | a = {1.3}; b = {2.5}; n = {2\*n}; ------------#")

print(f"#-- Result : {simpson(variantF, 1.3, 2.5, 2\*n)}")



Программная реализация следующих задач громоздка, хотя не имеет практической ценности. Пропустим её.

## Задача 7



## Задача 8

