Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 «Программная инженерия» –

Системное и прикладное программное обеспечение

**Отчёт**

**По лабораторной работе №3**

**По вычислительной математике**

**Вариант: 3**

Выполнил:

студент 2 курса

Батманов Даниил Евгеньевич

Группа: Р3207

Принял:

Рыбаков Степан Дмитриевич

Отчёт принят «\_\_»\_\_\_\_\_2024 г.

Оценка: \_\_\_\_\_\_\_\_\_\_\_

г. Санкт-Петербург, 2024

**Оглавление**

[Задание 3](#_Toc162833322)

[Рабочие формулы 3](#_Toc162833323)

[Часть I: вычислительная 3](#_Toc162833324)

[Часть II: программная 5](#_Toc162833325)

[Вывод 9](#_Toc162833326)

# 

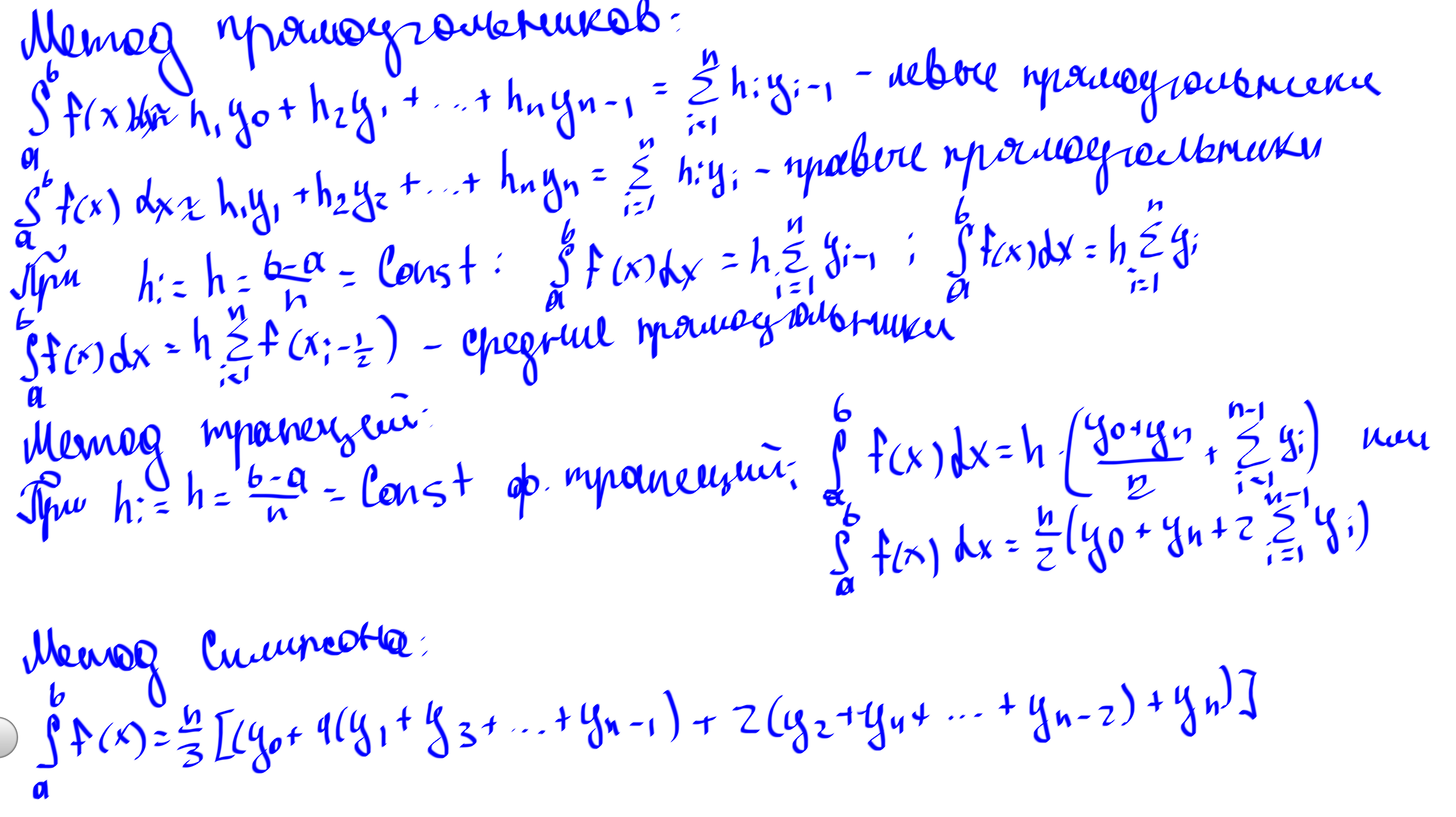
# 

# 

# Задание

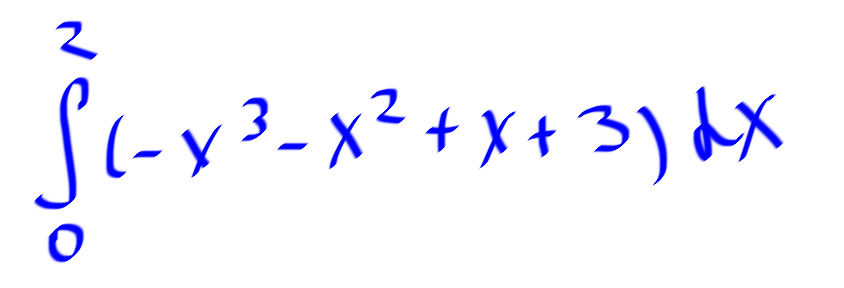
Цель работы: найти приближённое значение определённого интеграла с требуемой точностью различными численными методами.

# Рабочие формулы

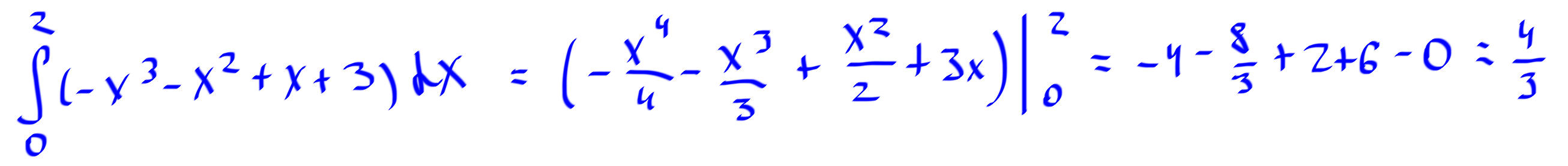


# Часть I: вычислительная

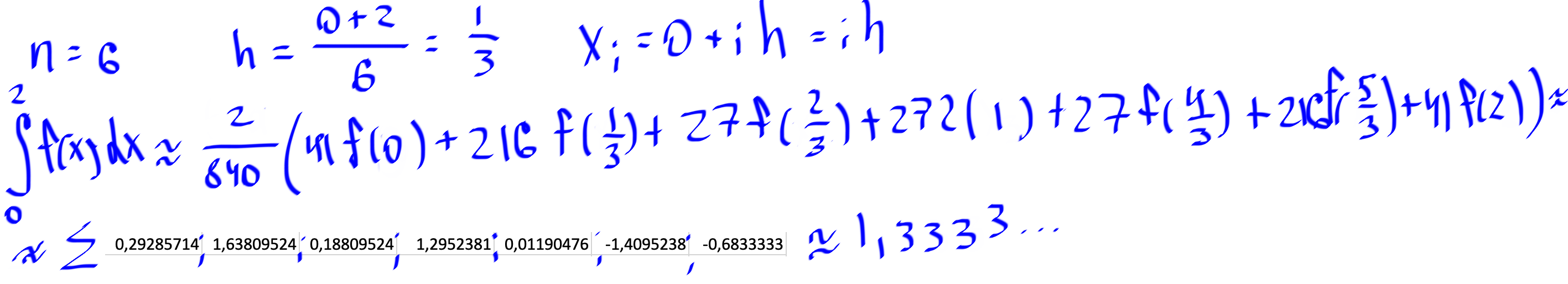
***Исходный интеграл:***



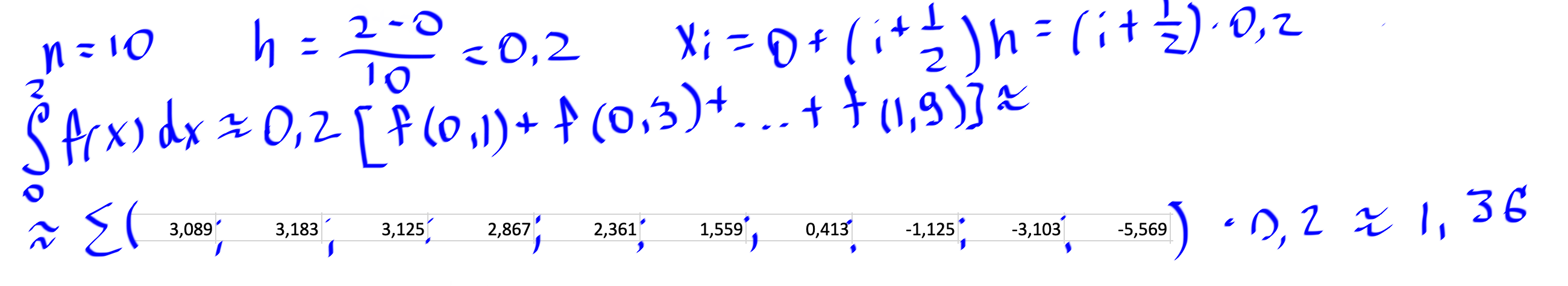
*Точное вычисление интеграла:*



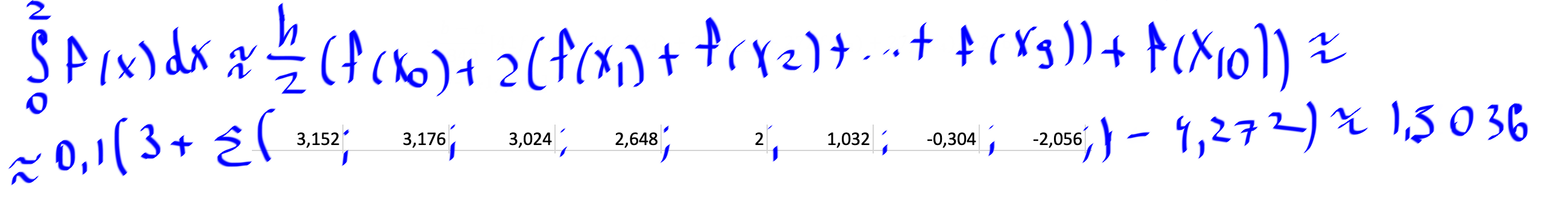
*Решение интеграла по формуле Ньютона-Котеса при n = 6:*



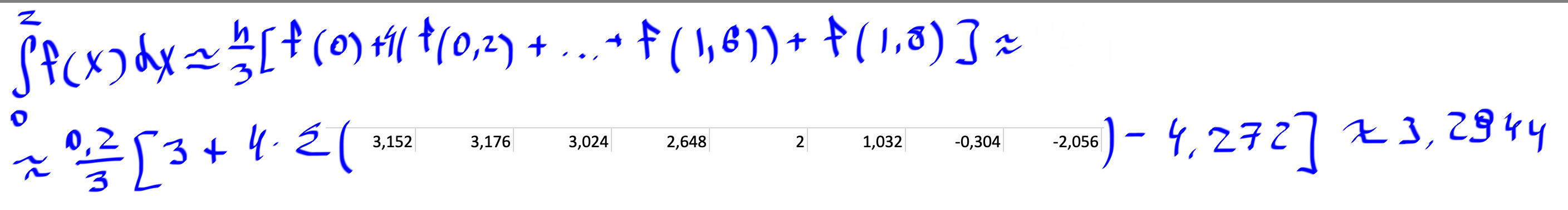
*Решение интеграла методом средних прямоугольников при n = 10:*



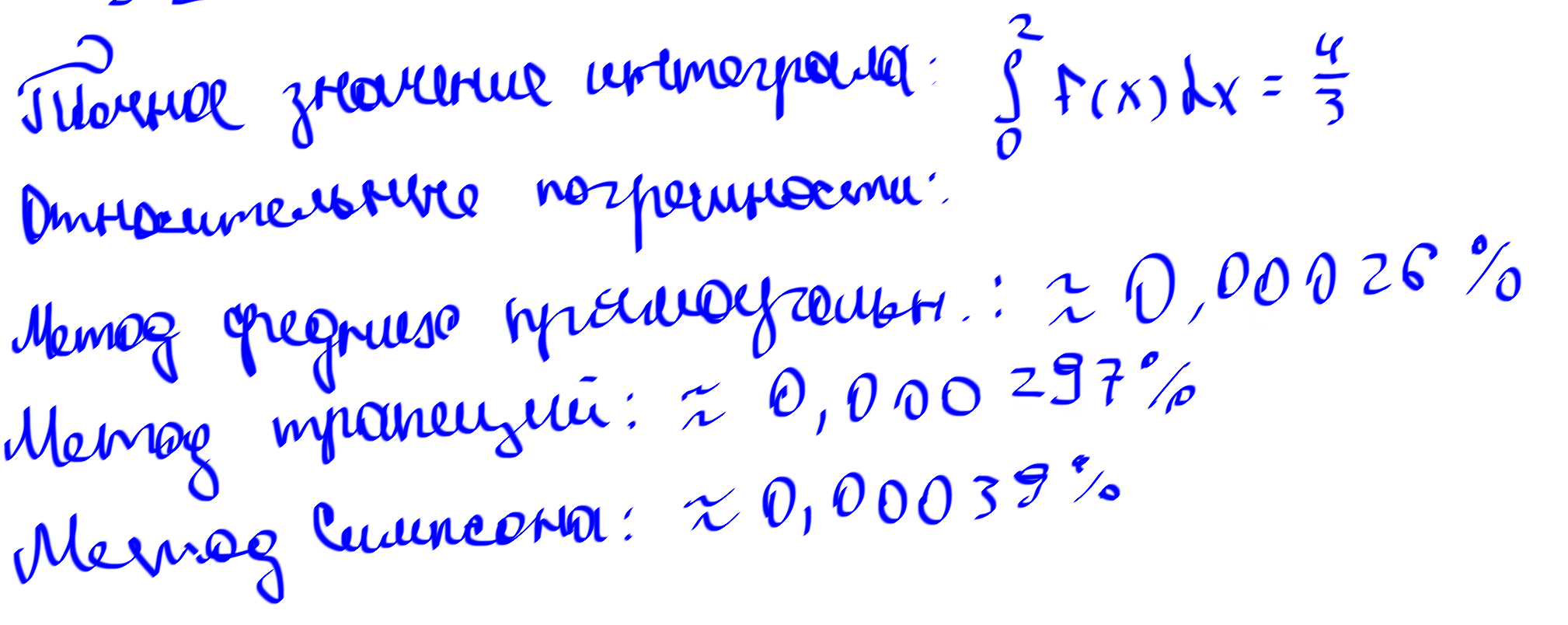
*Решение интеграла методом трапеций при n = 10:*



*Решение интеграла методом Симпсона при n = 10:*



*Сравнение результатов и расчёт погрешностей:*

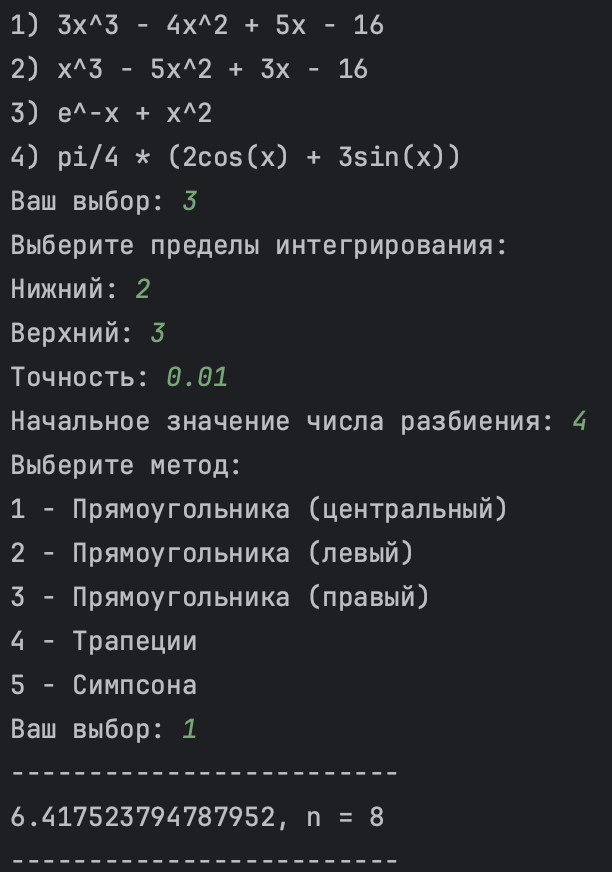


# Часть II: программная

Исходный код:

import math  
  
  
def separate():  
 print("-------------------------")  
  
def get\_function(func\_num):  
 if func\_num == 1:  
 return lambda x: 3 \* x \*\* 3 - 4 \* x \*\* 2 + 5 \* x - 16  
 elif func\_num == 2:  
 return lambda x: x \*\* 3 - 5 \* x \*\* 2 + 3 \* x - 16  
 elif func\_num == 3:  
 return lambda x: math.e \*\* (-x) + x \*\* 2  
 elif func\_num == 4:  
 return lambda x: math.pi / 4 \* (2 \* math.cos(x) + 3 \* math.sin(x))  
  
  
def rectangle\_mid(a, b, e, func\_number, n=4):  
 function = get\_function(func\_number)  
 sum\_i0 = 0  
 a\_new = a  
 h = (b - a) / n  
 for i in range(0, n):  
 sum\_i0 += function(a\_new + h / 2)  
 a\_new += h  
 sum\_i0 = sum\_i0 \* h  
  
 while True:  
 a\_new = a  
 sum\_i1 = 0  
 n \*= 2  
 h = (b - a) / n  
  
 for i in range(0, n):  
 sum\_i1 += function(a\_new + h / 2)  
 a\_new += h  
 sum\_i1 = sum\_i1 \* h  
  
 if (abs(sum\_i1 - sum\_i0) / 3) <= e:  
 print(f"{sum\_i1}, n = {n}")  
 separate()  
 break  
 sum\_i0 = sum\_i1  
  
  
def rectangle\_left(a, b, e, func\_number, n=4):  
 function = get\_function(func\_number)  
 a\_new = a  
 h = (b - a) / n  
 sum\_i0 = 0  
 for i in range(0, n):  
 sum\_i0 += function(a\_new)  
 a\_new += h  
 sum\_i0 = sum\_i0 \* h  
  
 while True:  
 a\_new = a  
 sum\_i1 = 0  
 n \*= 2  
 h = (b - a) / n  
  
 for i in range(0, n):  
 sum\_i1 += function(a\_new)  
 a\_new += h  
 sum\_i1 = sum\_i1 \* h  
  
 if (abs(sum\_i1 - sum\_i0) / 3) <= e:  
 print(f"{sum\_i1}, n = {n}")  
 separate()  
 break  
 sum\_i0 = sum\_i1  
  
  
def rectangle\_right(a, b, e, func\_number, n=4):  
 function = get\_function(func\_number)  
 a\_new = a  
 h = (b - a) / n  
 a\_new += h  
 sum\_i0 = 0  
 for i in range(0, n):  
 sum\_i0 += function(a\_new)  
 a\_new += h  
 sum\_i0 = sum\_i0 \* h  
  
 while True:  
 a\_new = a  
 sum\_i1 = 0  
 n \*= 2  
 h = (b - a) / n  
 a\_new += h  
  
 for i in range(0, n):  
 sum\_i1 += function(a\_new)  
 a\_new += h  
 sum\_i1 = sum\_i1 \* h  
  
 if (abs(sum\_i1 - sum\_i0) / 3) <= e:  
 print(f"{sum\_i1}, n = {n}")  
 separate()  
 break  
 sum\_i0 = sum\_i1  
  
  
def trapezoid(a, b, e, func\_number, n=4):  
 function = get\_function(func\_number)  
 sum\_i0 = 0  
 h = (b - a) / n  
  
 for i in range(1, n):  
 sum\_i0 += function(a + i \* h)  
 sum\_i0 = (function(a) + function(b) + 2 \* sum\_i0) \* h / 2  
  
 while True:  
 sum\_i1 = 0  
 n \*= 2  
 h = (b - a) / n  
  
 for i in range(1, n):  
 sum\_i1 += function(a + i \* h)  
 sum\_i1 = (function(a) + function(b) + 2 \* sum\_i1) \* h / 2  
  
 if abs(sum\_i1 - sum\_i0) / 3 <= e:  
 print(f"{sum\_i1}, n = {n}")  
 separate()  
 break  
 sum\_i0 = sum\_i1  
  
  
def simpson(a, b, e, func\_number, n=4):  
 print(a, b, e, func\_number, n)  
 function = get\_function(func\_number)  
 sum\_i0 = 0  
 h = (b - a) / n  
  
 for i in range(1, n, 2):  
 sum\_i0 += 4 \* function(a + i \* h)  
 for i in range(2, n - 1, 2):  
 sum\_i0 += 2 \* function(a + i \* h)  
  
 sum\_i0 = (function(a) + function(b) + sum\_i0) \* h / 3  
  
 while True:  
 sum\_i1 = 0  
 n \*= 2  
 h = (b - a) / n  
  
 for i in range(1, n, 2):  
 sum\_i1 += 4 \* function(a + i \* h)  
 for i in range(2, n - 1, 2):  
 sum\_i1 += 2 \* function(a + i \* h)  
  
 sum\_i1 = (function(a) + function(b) + sum\_i1) \* h / 3  
  
 if abs(sum\_i1 - sum\_i0) / 15 <= e:  
 print(f"{sum\_i1}, n = {n}")  
 separate()  
 break  
 sum\_i0 = sum\_i1  
  
  
print(f'1) 3x^3 - 4x^2 + 5x - 16')  
print(f'2) x^3 - 5x^2 + 3x - 16')  
print(f'3) e^-x + x^2')  
print(f'4) pi/4 \* (2cos(x) + 3sin(x))')  
  
choice = int(input("Ваш выбор: "))  
print("Выберите пределы интегрирования: ")  
input\_a = float(input("Нижний: "))  
input\_b = float(input("Верхний: "))  
epsilon = float(input("Точность: "))  
nn = int(input("Начальное значение числа разбиения: "))  
while True:  
 method\_number = int(input("Выберите метод:\n1 - Прямоугольника (центральный)\n2 - Прямоугольника (левый)\n3 - "  
 "Прямоугольника (правый)\n4 - Tрапеции\n5 - Симпсона\nВаш выбор: "))  
 separate()  
 if method\_number == 1:  
 rectangle\_mid(input\_a, input\_b, epsilon, choice, nn)  
 elif method\_number == 2:  
 rectangle\_left(input\_a, input\_b, epsilon, choice, nn)  
 elif method\_number == 3:  
 rectangle\_right(input\_a, input\_b, epsilon, choice, nn)  
 elif method\_number == 4:  
 trapezoid(input\_a, input\_b, epsilon, choice, nn)  
 elif method\_number == 5:  
 simpson(input\_a, input\_b, epsilon, choice, nn)  
 else:  
 print("Введите корректный номер метода!")  
 continue  
 break

Вывод программы:



# Вывод

В ходе выполнения данной лабораторной работы я изучил численные методы решения определённых интегралов и реализовал программный продукт, который позволяет решать интегралы численным методом.