МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра математического моделирования**

**ОТЧЕТ О ПРОХОЖДЕНИИ УЧЕБНОЙ ПРАКТИКИ**

(практике по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности)

Выполнил Е.Д. Худышкин

Направление подготовки 02.03.02 Математическое обеспечение и администрирование информационных систем

Курс 1

Руководитель учебной практики

канд. физ.-мат. наук, доцент кафедры

математического моделирования С.Е. Рубцов

Краснодар

2020г.

Содержание

[Содержание 2](#_Toc44765095)

[1 Постановка задачи 3](#_Toc44765096)

[2 Метод вычисления интеграла 4](#_Toc44765097)

[3 Расчетные формулы для вычисления значений 5](#_Toc44765098)

[4 Аналитическое вычисление интеграла 6](#_Toc44765099)

[5 Описание работы программы 8](#_Toc44765100)

[6 Графики решений 10](#_Toc44765101)

[Список используемых источников 12](#_Toc44765102)

[Приложение 13](#_Toc44765103)

# **1 Постановка задачи**

Пусть задана функция

Написать программу на языке высокого уровня (Python) для расчета значений функции на промежутке . Для вычисления интеграла использовать метод трапеций с шагом h ().

Произвести расчеты для различных значений разбиения интервала интегрирования (например, при N = 5, 10, 20, 50).

Аналитически вычислить интеграл. Произвести сравнение точного и приближенного решений: вычислить максимальную невязку (наибольшую по абсолютной величине разность между точным и приближенным решениями для различных значений ().

В одной системе координат построить графики точного и приближенного решений. Для построения графиков использовать графические возможности выбранного языка программирования или Microsoft Excel.

Создать в электронном виде и распечатать отчет по учебной практике. При этом использовать стандарты, принятые для оформления курсовых и дипломных работ.

Отчет должен включать постановку задачи, описание численного метода, используемого для расчета интеграла, выкладки аналитического вычисления интеграла, графики точного и приближенных решений, полученных для различных значений N, вычисленные значения невязок, текст программы.

# **2 Метод вычисления интеграла**

Пусть функция непрерывна на отрезке . Требуется вычислить определенный интеграл , который численно представляет собой площадь криволинейной трапеции.

Разобьем отрезок на N частей точками:

.

По определению определенный интеграл есть предел интегральных сумм при бесконечном уменьшении длины отрезка разбиения, следовательно, любая из интегральных сумм является приближенным значением интеграла.

Определение: Интегральной суммой для функции на отрезке для заданного набора точек , где называется сумма вида

.

Суть метода трапеций заключается в том, что в качестве приближенного значения берется интегральная сумма

Иначе говоря, на каждом элементарном отрезке подынтегральная функция заменяется полиномом нулевой степени. Отсюда следует, что алгебраический порядок точности данного численного метода равен 0.

С геометрической точки зрения для неотрицательной функции на отрезке точное значение определенного интеграла представляет собой площадь криволинейной трапеции, а приближенное значение по методу трапеций – площадь фигуры, образованной прямолинейными трапециями (рисунок 1).

# **3 Расчетные формулы для вычисления значений**

Формула интегрирования для метода трапеций:

.

Абсолютная погрешность метода трапеций вычисляется по формуле:

.

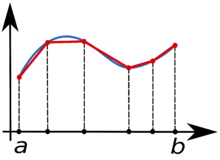


Рисунок 1 – Графическое представление метода трапеций

# **4 Аналитическое вычисление интеграла**

Вычислить определенный интеграл

(1)

где

*a* – 1

*b –* 2

Рассмотрим вычисление неопределенного интеграла:

Разложим дробь на простейшие:

Применим линейность:

Теперь вычисляем отдельные интегралы:

1)

Производим подстановку:

2) , это табличный интеграл:

3) , это табличный интеграл:

Подставим уже вычисленные интегралы:

Теперь вернёмся к вычислению определённого интеграла, получаем:

# **5 Описание работы программы**

Программа написана на высокоуровневом языке программирования Python, версии 3.8.1.

Для корректной работы программы импортируем сторонние библиотеки:

Numpy — фундаментальный пакет для научных вычислений на Python

Matplotlib — это комплексная библиотека для создания статических, анимированных и интерактивных визуализаций в Python, из неё нам понадобится только функционал для построения графика (pylab).

Далее идёт блок с объявлением и вводом необходимых переменных:

a — Нижний предел интегрирования.

b — Верхний предел интегрирования.

c — Верхний предел для X.

d — Нижний предел для X.

N — Кол-во разбиений.

h — шаг, с которым мы идём по промежутку (вычисляется по формуле h = (b-a)/N).

k — кол-во сравниваемых точек, по которым строятся наши графики.

В переменную f записываем лямбда-функцию, в которой объявляем наше подынтегральное выражение, с которым будет происходить работа программы.

В двух следующих функциях описываются два метода вычисления интеграла:

1) Функция trapez – вычисляет приближённое значение интеграла методом трапеций.

2) Функция exactIntegral – вычисляет точное значение интеграла.

В следующем блоке происходит вычисление точек, которые будут использованы для построения графика функции, по формуле

вычисляем и заполняем список со значениями точек x и преобразуем его в массив из библиотеки numpy, для удобства вычислений.

После вычисления точек X, вычисляем значение точек Y, двумя приведенными выше способами.

Далее вычисляем значение максимальной невязки (размер ошибки), путём сравнения точек Y, которые были вычислены разными способами и выводим её значение в консоль.

С помощью библиотеки Matplotlib, строим графики точного и приближённого значений и выводим в отдельном окне.

# **6 Графики решений**

На рисунках представлен график решений для N=5, N=10, N=20, N=50.

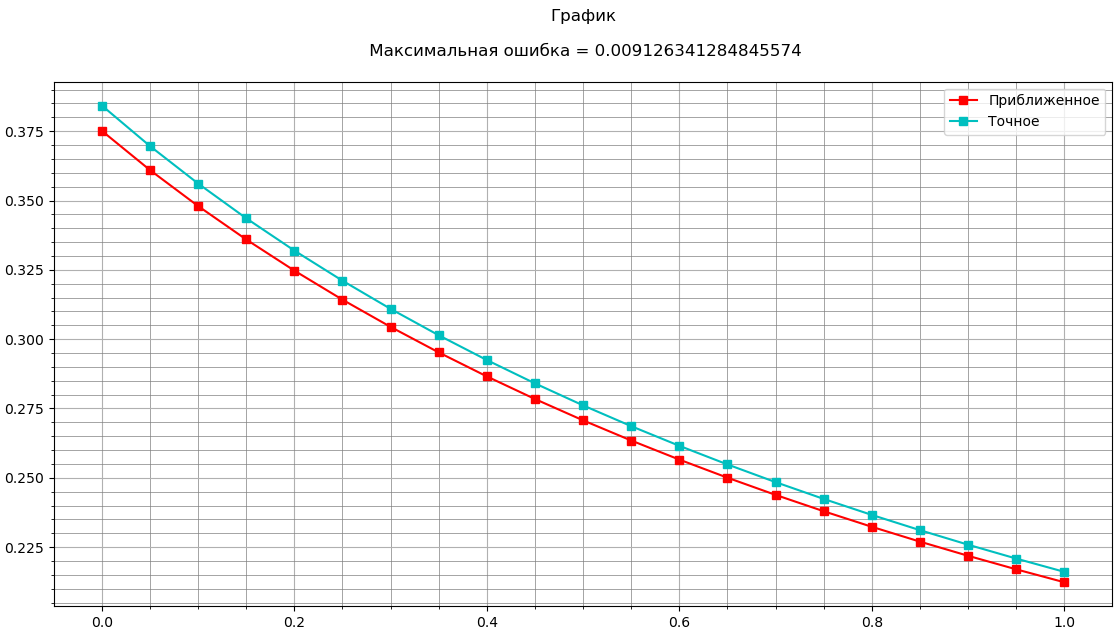


Рисунок 2 – Графики для N = 5

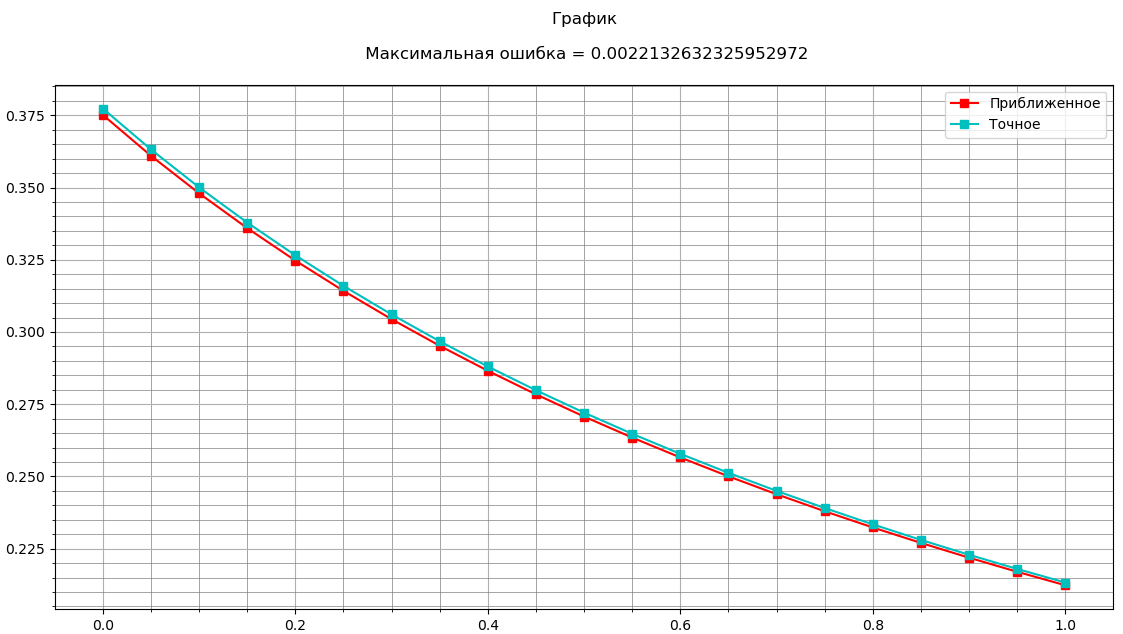
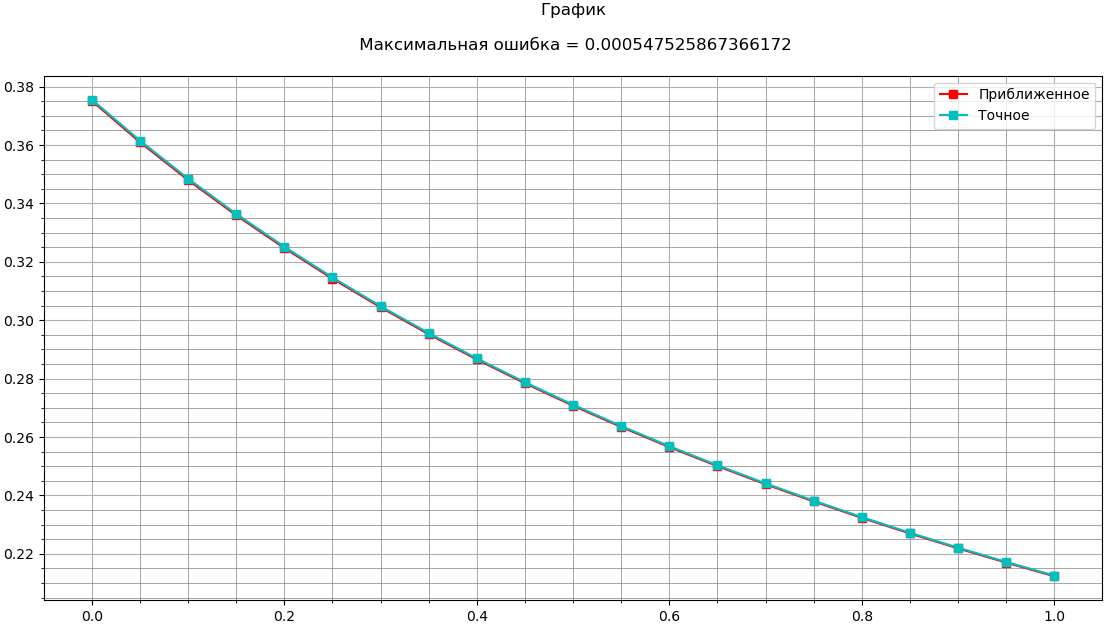


Рисунок 3 – Графики для N = 10

 Рисунок 4 – Графики для N = 20

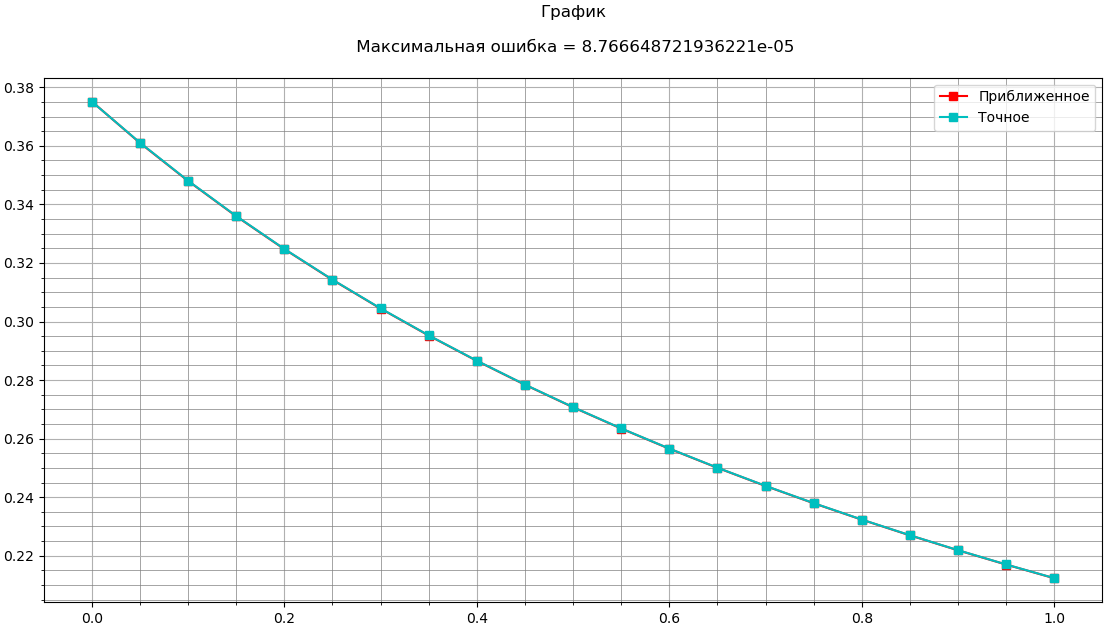


Рисунок 5 – Графики для N = 50

# Список используемых источников

1. Доля П.Г. Введение в научный Python.: учебное пособие / Харьков: Издательство Харьковского Национального Университета, 2016. — 265 с.

2. Помощь по Microsoft Office: Microsoft Office Word 2019 (русский). [Электронный ресурс] – URL: https://support.office.com/ru-ru (19.04.2020)

3. Кудрявцев, П.П. Краткий курс математического анализа. Т.1. Дифференциальное и интегральное исчисления функций одной переменной. Ряды: Учебник. / П.П. Кудрявцев – М.: ФИЗМАТЛИТ, 2015. – 444 с.

Приложение

Код программы

1. **import** numpy as np
2. **import** matplotlib.pylab as plt
4. # Объявление и ввод переменных:
5. a **=** float(input("Нижний предел интегрирования: "))
6. b **=** float(input("Верхний предел интегрирования: "))
7. c **=** float(input("Верхний предел для X: "))
8. d **=** float(input("Нижний предел для X: "))
9. N **=** int(input("Кол-во разбиений: "))
10. h **=** (b**-**a)**/**N
11. k **=** 21 # кол-во сравниваемых точек
13. # Подынтегральное выражение:
14. f **=** **lambda** x, t: 1**/**((x**+**t)**\***t**\*\***2)
16. # Метод трапеций
17. **def** trapez(x):
18. sumOfInter **=** 0.5 **\*** (f(x, a) **+** f(x, b))
19. **for** i **in** range(1, N):
20. sumOfInter **+=** f(x, a **+** i **\*** h)
21. **return** sumOfInter **\*** h
23. # Точное вычисление
24. **def** exactIntegral(xPoints):
25. **return** np.log(xPoints[i]**+**2)**/**xPoints[i]**\*\***2**-**(np.log(xPoints[i]**+**1)**-**xPoints[i])**/**xPoints[i]**\*\***2**-**1**/**(2**\***xPoints[i])**-**np.log(2)**/**xPoints[i]**\*\***2
27. # Создание точек
28. xPoints **=** [c **+** i**\***(d**-**c)**/**20 **for** i **in** range(k)]
29. xPoints **=** np.array(xPoints)
31. **for** i **in** range (k):
32. **if** xPoints[i] **==** 0:
33. xPoints[i] **=** 0.000001
35. # Точное значение:
36. yPoints **=** []
37. yExactPoints **=** []
38. **for** i **in** range (k):
39. yExactPoints.append(exactIntegral(xPoints))
41. # Приближенное значение:
42. yPoints **=** trapez(xPoints)
44. # Размер ошибки:
45. error **=** 0.0
46. **for** i **in** range(k):
47. **if** abs(yPoints[i] **-** yExactPoints[i]) > error:
48. error **=** abs(yPoints[i] **-** yExactPoints[i])
49. print("Максимальная ошибка = ", error)
51. # Построение графика:
52. plt.title("График\n\n Максимальная ошибка **=** {}\n".format(error))
53. plt.ylabel("Значение определенного интеграла")
54. plt.xlabel("Значение переменной x")
55. plt.plot(xPoints, yExactPoints, "sr-", label**=**"Приближенное")
56. plt.plot(xPoints, yPoints, "sc-" , label**=**"Точное")
57. plt.legend()
58. plt.grid(True)
59. plt.minorticks\_on()
60. plt.grid(which**=**'minor', color**=**'gray', linestyle**=**'-', linewidth**=**0.5)
61. plt.show()