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 О.И. Лихогуб

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

[1 Постановка задачи 3](#_Toc45296401)

[2 Метод вычисления интеграла 4](#_Toc45296402)

[3 Расчетные формулы для вычисления значений 5](#_Toc45296403)

[4 Аналитическое вычисление интеграла 6](#_Toc45296404)

[5 Описание работы программы 11](#_Toc45296405)

[6 Графики решений 12](#_Toc45296406)

[Список используемых источников 14](#_Toc45296407)

[Приложение. Код программы 15](#_Toc45296408)

# **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.

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

где

*a* = 0,

*b =* 1.

Рассмотрим вычисление неопределенного интеграла:

Введем замену:

Применим замену и вынесем из-под корня :

Применим к интегралу следующую формулу:

тогда уравнение примет вид:

Вынесем из-под знака интеграла как постоянный множитель:

Запишем согласно следующей формуле и преобразуем:

тогда уравнение примет вид:

Запишем интеграл, применив следующее равенство:

тогда получим, что:

Используя формулу понижения степени секанса:

вычислим первообразные секансов пятой и третьей степени:

Вычислим первообразную секанса первой степени, произведя следующую обратную замену:

получим, что:

Отметим, что:

занесем это выражение под знак дифференциала и вычислим интеграл:

Таким образом

Запишем исходное уравнение с учётом полученных равенств:

Использую следующие равенства:

вернемся к замене:

Упростим:

Вернёмся к вычислению определённого интеграла, получаем:

Особый случай:

 При x = 0, получим деление на ноль, для того, чтобы избавиться от него, вручную вычислим неопределённый интеграл, подставив значение x = 0:

По формуле Ньютона-Лейбница:

# **5 Описание работы программы**

Программа написана на высокоуровневом языке программирования Python. Она вычисляет значение определенного интеграла в заданных промежутках и находит значение максимальной невязки при заданных значениях.

Для начала работы импортируем математические функции из библиотеки math и импортируем модуль pyplot из библиотеки matplotlib, который используем для построения графиков.

Задаем подынтегральное выражение и аналитически вычисленную первообразную как функции, запрашиваем данные о промежутках интегрирования у пользователя.

Генерируем списки значений переменной X и переменной T с заданным шагом в заданном интервале.

Для каждого значения x из заданного интервала вычисляем определенный интеграл методом трапеций и аналитически (для случая, когда x = 0, то подставим значение, вычисленное вручную).

Вычисляем максимальное значение невязки между решением методом трапеций и аналитическим решением и выводим полученное значение невязки в консоль.

Генерируем приближенные решения интеграла для разбиений N=5, 10, 20, 50 и для N, введенного пользователем и выводим график полученных решений интеграла в одной системе координат.

# **6 Графики решений**

Результаты вычислений, полученные с помощью программы, написанной на языке высокого уровня, представлены в таблице 1, а так же на рисунке 2 и рисунке 3.

Таблица 1. Максимальная невязка при соответствующих разбиениях интервала интегрирования

|  |  |
| --- | --- |
| Значение N | Максимальная невязка |
| 5 | 0.016387567365723443 |
| 10 | 0.004098816821936135 |
| 20 | 0.0010248244247618166 |
| 50 | 0.0001639772927318761 |



Рисунок 2 – Графики приближенного и точного решений в одной системе координат.



Рисунок 3 – Графики приближенного и точного решений в одной системе координат. Крупно показан участок графика при x от 0 до 0.5.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. 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.
2. Гэддис Т. Начинаем программировать на Python.  – 4-е изд.: Пер. с англ. – СПб.: БХВ-Петербург, 2019. – 768 с.
3. Помощь по Microsoft Office: Microsoft Office Word 2019 (русский). [Электронный ресурс] – URL: <https://support.office.com/ru-ru> (19.04.2020)

Приложение

Код программы

from math import log, sqrt

from matplotlib import pyplot as plt

f = lambda t, x: t\*\*2\*sqrt(x\*\*2 + t\*\*2)

F = lambda x: x\*sqrt(1+x\*\*2)\*(2+x\*\*2)/8/abs(x) -x\*\*4\*log(abs((1+sqrt(1+x\*\*2))/x))/8+x\*\*4\*log(1)/8

diff = lambda a: abs(a[1] - a[0])

a = float(input("Введите a: "))

b = float(input("Введите b: "))

c = float(input("Введите c: "))

d = float(input("Введите d: "))

N = int(input("Введите N: "))

X = [c + i \* (d-c) / 20 for i in range(21)]

h = (b-a) / N

T = [a + i \* h for i in range(N + 1)]

Iapproximate = [h \* sum((f(T[i - 1], x) + f(T[i], x)) / 2 for i in range(1, N + 1)) for x in X]

Iaccurate = [F(x) if x != 0 else 0.25 for x in X]

maximum\_error = max(list(map(diff, list(zip(Iapproximate, Iaccurate)))))

print("Максимальная невязка при N={}: ".format(str(N)), maximum\_error)

plt.title("График значений интеграла при различных N")

plt.ylabel("Значение определенного интеграла")

plt.xlabel("Значение X")

plt.grid(True)

for N in set([5, 10, 20, 50, N]):

 h = (b-a) / N

 T = [a + i \* h for i in range(N + 1)]

 Iapproximate = [h \* sum([(f(T[i - 1], x) + f(T[i], x)) / 2 for i in range(1, N + 1)]) for x in X]

 plt.plot(X, Iapproximate, label="N = {}".format(N))

plt.plot(X, Iaccurate, color="black", alpha=0.5, label="Точное")

plt.legend()

plt.show()