**Национальный исследовательский университет ИТМО**

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Вычислительная математика»

**Отчет**

По лабораторной работе №3

Вариант 9

Выполнил:

*Кочнев Р. Д.*

*P32081*

Преподаватель:

*Рыбаков С. Д.*

Санкт-Петербург, 2023 г.

Цель работы

Цель работы

Найти приближенное значение определенного интеграла с требуемой точностью различными численными методами.

Порядок выполнения

1. Изучить численные методы нахождения интегралов
2. Найти интеграл вручную
3. Найти в excel решение методами Ньютона – Котеса, Средних прямоугольников, трапеций, Симпсона
4. Сделать красиво
5. Программно реализовать методы хорд, Ньютона, простой итерации и метод простой итерации для СЛАУ
6. Проверить правильность решения

Таблица для метода Ньютона – Котеса

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| n | 0 | 1 | 2 | 3 | 4 | 5 |
| c | -0,06597222222 | -0,2604166667 | -0,1736111111 | -0,1736111111 | -0,2604166667 | -0,06597222222 |
| x | 1 | 1,2 | 1,4 | 1,6 | 1,8 | 2 |
| f(x)\*c | 0,3298611111 | 1,00625 | 0,4152777778 | 0,08472222222 | -0,50625 | -0,3298611111 |

Таблица для метода Средних прямоугольников

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| xi | 1 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 2 |
| yi | -5 | -4,468 | -3,864 | -3,176 | -2,392 | -1,5 | -0,488 | 0,656 | 1,944 | 3,388 | 5 |
| xi-1/2 |  | 1,05 | 1,15 | 1,25 | 1,35 | 1,45 | 1,55 | 1,65 | 1,75 | 1,85 | 1,95 |
| yi-1/2 |  | -4,734 | -4,166 | -3,52 | -2,784 | -1,946 | -0,994 | 0,084 | 1,3 | 2,666 | 4,194 |

Таблица для метода трапеций

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| xi | 1 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 2 |
| yi | -5 | -4,468 | -3,864 | -3,176 | -2,392 | -1,5 | -0,488 | 0,656 | 1,944 | 3,388 | 5 |

Таблица для метода Симпсона

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| xi | 1 | 1,1 | 1,2 | 1,3 | 1,4 | 1,5 | 1,6 | 1,7 | 1,8 | 1,9 | 2 |
| yi | -5 | -4,468 | -3,864 | -3,176 | -2,392 | -1,5 | -0,488 | 0,656 | 1,944 | 3,388 | 5 |

Пример выполнения программы

Enter the bounds of the integral: 1 2

Enter the exp:0.001

Choose an integration method:

1: left\_rectangles

2: right\_rectangles

3: middle\_rectangles

4: trapezoidal

5: simpson

Enter the number of your choice: 2

Result: 0.38899942417149685

Number of subdivisions:128

Значение интеграла при обработке разрывов: 0.386294

Число разбиений интервала при обработке разрывов: 8

Код программы

<https://github.com/poma12390/2course/tree/master/witch_math/lab3>

import numpy as np  
  
import math  
  
  
def left\_rectangles(func, a, b, n):  
 h = (b - a) / n  
 return h \* sum(func(a + i \* h) for i in range(n))  
  
  
def right\_rectangles(func, a, b, n):  
 h = (b - a) / n  
 return h \* sum(func(a + (i + 1) \* h) for i in range(n))  
  
  
def middle\_rectangles(func, a, b, n):  
 h = (b - a) / n  
 return h \* sum(func(a + (i + 0.5) \* h) for i in range(n))  
  
  
def trapezoidal(func, a, b, n):  
 h = (b - a) / n  
 return h \* (0.5 \* (func(a) + func(b)) + sum(func(a + i \* h) for i in range(1, n)))  
  
  
def simpson(func, a, b, n):  
 h = (b - a) / (2 \* n)  
 return h / 3 \* (func(a) + func(b) + 4 \* sum(func(a + (2 \* i - 1) \* h) for i in range(1, n + 1)) + 2 \* sum(  
 func(a + 2 \* i \* h) for i in range(1, n)))  
  
  
# Обработка бесконечных разрывов  
def handle\_infinite(f, a, b, tol):  
 if a == float('-inf'):  
 return runge\_rule(simpson, f1, a, b, tol)  
 elif b == float('inf'):  
 return runge\_rule(simpson, f1, a, b, tol)  
 else:  
 return runge\_rule(simpson, f1, a, b, tol)  
  
  
# Проверка на сходимость интегралов 2 рода  
def check\_convergence(f, a, b):  
 if a == float('-inf') and b == float('inf'):  
 return False  
 if a == float('-inf'):  
 if hasattr(f, "converges\_at\_inf"):  
 return f.converges\_at\_inf  
 else:  
 return False  
 if b == float('inf'):  
 if hasattr(f, "converges\_at\_minus\_inf"):  
 return f.converges\_at\_minus\_inf  
 else:  
 return False  
 return True  
  
  
def runge\_rule(method, func, a, b, tol):  
 n = 1  
 prev\_res = method(func, a, b, n)  
 n \*= 2  
 curr\_res = method(func, a, b, n)  
 k = 2  
 while abs(curr\_res - prev\_res) / (2 \*\* k - 1) > tol:  
 k += 1  
 n \*= 2  
 prev\_res = curr\_res  
 curr\_res = method(func, a, b, n)  
  
 return curr\_res, n  
  
  
methods = {  
 "left\_rectangles": left\_rectangles,  
 "right\_rectangles": right\_rectangles,  
 "middle\_rectangles": middle\_rectangles,  
 "trapezoidal": trapezoidal,  
 "simpson": simpson  
}  
  
f1 = lambda x: 2 \* x \*\* 3 - 3 \* x \*\* 2 + 5 \* x - 9  
  
a, b = map(float, input("Enter the bounds of the integral: ").split())  
tol = float(input("Enter the exp:"))  
  
print("Choose an integration method:")  
for i, method in enumerate(methods.keys()):  
 print(f"{i + 1}: {method}")  
choice = int(input("Enter the number of your choice: ")) - 1  
method\_name = list(methods.keys())[choice]  
method = methods[method\_name]  
  
if not check\_convergence(f1, a, b):  
 print("Интеграл не существует")  
 exit(1)  
  
result, n = runge\_rule(method, f1, a, b, tol)  
print(f"Значение интеграла: {result}")  
print(f"Число разбиений интервала:{n}")  
  
result, n = handle\_infinite(f1, a, b, tol)  
print(f"Значение интеграла при обработке разрывов: {result:.6f}")  
print(f"Число разбиений интервала при обработке разрывов: {n}")

;l

Вывод

Познакомился с различными методами нахождения интегралов. Посчитал интеграл этими методами. Написал программу для нахождения этих интегралов.