Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

Лабораторная работа № 3

Вычислительная математика

Вариант №9

Студент

Карагодина Ксения Андреевна

группа Р3215

Преподаватель

Малышева Татьяна Алексеевна

Санкт - Петербург 2025 год

**Вычислительная реализация задачи**

|  |
| --- |
|  |

1. Вычислим интеграл точно, с помощью формулы Ньютона-Лейбница
2. Метод Ньютона – Котеса при 𝑛 = 6.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | *0* | *1* | *2* | *3* | *4* | *5* | *6* |
|  | *1* |  |  |  |  |  | *2* |
|  | *-5* |  |  |  |  |  | *5* |

*Погрешность = 0*

1. Метод средних прямоугольников

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* |
|  | *1* | *1.1* | *1.2* | *1.3* |  |  | *1.6* | *1.7* | *1.8* | *1.9* | *2* |
|  | *-5* |  |  |  |  | *-1.5* | *-0.488* | *0.656* | *1.944* | *3.388* | *5* |
|  |  | *1.05* | *1.15* | *1.25* | *1.35* | *1.45* | *1.55* | *1.65* | *1.75* | *1.85* | *1.95* |
|  |  | *-4.74225* | *-4.17575* | *-3.53125* | *-2.79675* | *-1.96025* | *-1.00975* | *0.06675* | *1.28125* | *2.64575* | *4.17225* |

*Погрешность в вычислении интеграла составляет :*

1. Метод трапеций

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* |
|  | *1* | *1.1* | *1.2* | *1.3* |  |  | *1.6* | *1.7* | *1.8* | *1.9* | *2* |
|  | *-5* |  |  |  |  | *-1.5* | *-0.488* | *0.656* | *1.944* | *3.388* | *5* |

*Погрешность в вычислении интеграла составляет :*

1. Метод Симпсона

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* |
|  | *1* | *1.1* | *1.2* | *1.3* |  |  | *1.6* | *1.7* | *1.8* | *1.9* | *2* |
|  | *-5* |  |  |  |  | *-1.5* | *-0.488* | *0.656* | *1.944* | *3.388* | *5* |

*Погрешность в вычислении интеграла составляет :*

**Вычислительная реализация задачи**

*def left\_rectangles(f, a, b, n):  
 h = (b - a) / n  
 integral = 0.0  
 for i in range(n):  
 x = a + i \* h  
 integral += f(x)  
 return integral \* h  
  
  
def right\_rectangles(f, a, b, n):  
 h = (b - a) / n  
 integral = 0.0  
 for i in range(1, n + 1):  
 x = a + i \* h  
 integral += f(x)  
 return integral \* h  
  
  
def middle\_rectangles(f, a, b, n):  
 h = (b - a) / n  
 integral = 0.0  
 for i in range(n):  
 x = a + (i + 0.5) \* h  
 integral += f(x)  
 return integral \* h  
  
  
def trapezoid(f, a, b, n):  
 h = (b - a) / n  
 integral = (f(a) + f(b)) / 2  
 for i in range(1, n):  
 x = a + i \* h # Текущая точка на оси X  
 integral += f(x)  
 return integral \* h # Умножаем сумму на шаг (получаем площадь всех трапеций)  
  
  
def simpson(f, a, b, n):  
 if n % 2 != 0:  
 n += 1 # Метод Симпсона требует четное число интервалов  
 h = (b - a) / n  
 integral = f(a) + f(b)  
 for i in range(1, n):  
 x = a + i \* h  
 if i % 2 == 0: # Если индекс чётный  
 integral += 2 \* f(x) # Учитываем с коэффициентом 2  
 else: # Если индекс нечётный  
 integral += 4 \* f(x) # Учитываем с коэффициентом 4  
 return integral \* h / 3 # Умножаем на (h/3) по формуле Симпсона  
  
  
def calculate\_integral():  
 # Выбор функции  
 print("Выберите функцию для интегрирования:")  
 print("1. x^2")  
 print("2. sin(x)")  
 print("3. e^x")  
 print("4. 1/x")  
 print("5. sqrt(x)")  
  
 choice = int(input("Введите номер функции (1-5): "))  
  
 if choice == 1:  
 f = lambda x: x \*\* 2  
 func\_name = "x^2"  
 a = float(input("Введите нижний предел интегрирования a: "))  
 b = float(input("Введите верхний предел интегрирования b: "))  
 if a >= b:  
 print("Ошибка: верхний предел должен быть больше нижнего. Попробуйте снова.")  
 return  
 elif choice == 2:  
 f = lambda x: math.sin(x)  
 func\_name = "sin(x)"  
 a = float(input("Введите нижний предел интегрирования a: "))  
 b = float(input("Введите верхний предел интегрирования b: "))  
 if a >= b:  
 print("Ошибка: верхний предел должен быть больше нижнего. Попробуйте снова.")  
 return  
 elif choice == 3:  
 f = lambda x: math.exp(x)  
 func\_name = "e^x"  
 a = float(input("Введите нижний предел интегрирования a: "))  
 b = float(input("Введите верхний предел интегрирования b: "))  
 if a >= b:  
 print("Ошибка: верхний предел должен быть больше нижнего. Попробуйте снова.")  
 return  
 elif choice == 4:  
 f = lambda x: 1 / x  
 func\_name = "1/x"  
 a = float(input("Введите нижний предел интегрирования a: "))  
 b = float(input("Введите верхний предел интегрирования b: "))  
 if (a <= 0 or b <= 0):  
 print("Ошибка: для функции 1/x пределы должны быть положительными. Попробуйте снова.")  
 return  
 elif (a >= b):  
 print("Ошибка: верхний предел должен быть больше нижнего. Попробуйте снова.")  
 return  
 elif choice == 5:  
 f = lambda x: math.sqrt(x)  
 func\_name = "sqrt(x)"  
 a = float(input("Введите нижний предел интегрирования a: "))  
 b = float(input("Введите верхний предел интегрирования b: "))  
 if (a < 0 or b < 0):  
 print("Ошибка: для функции sqrt(x) пределы не могут быть отрицательными. Попробуйте снова.")  
 return  
 elif a >= b:  
 print("Ошибка: верхний предел должен быть больше нижнего. Попробуйте снова.")  
 return  
 else:  
 print("Неверный выбор функции")  
 return  
  
 # Ввод пределов интегрирования  
  
  
 # Ввод точности  
 epsilon = float(input("Введите требуемую точность (например, 0.001): "))  
  
 # Выбор метода интегрирования  
 print("\nВыберите метод интегрирования:")  
 print("1. Левые прямоугольники")  
 print("2. Правые прямоугольники")  
 print("3. Средние прямоугольники")  
 print("4. Метод трапеций")  
 print("5. Метод Симпсона")  
 method\_choice = int(input("Введите номер метода (1-5): "))  
  
 methods = {  
 1: ("Левые прямоугольники", left\_rectangles),  
 2: ("Правые прямоугольники", right\_rectangles),  
 3: ("Средние прямоугольники", middle\_rectangles),  
 4: ("Метод трапеций", trapezoid),  
 5: ("Метод Симпсона", simpson)  
 }  
  
 if method\_choice not in methods:  
 print("Неверный выбор метода")  
 return  
  
 method\_name, method\_func = methods[method\_choice]  
  
 # Начальное число разбиений  
 n = 4  
  
 print("\nРезультаты вычислений:")  
 print("Функция: ", func\_name)  
 print("Интервал: [", a, ", ", b, "]")  
 print("Метод: ", method\_name)  
 print("Точность: ", epsilon)  
 print("\nИтерационный процесс:")  
  
 prev\_integral = 0  
  
 while True:  
 # Вычисляем интеграл выбранным методом  
 integral = method\_func(f, a, b, n)  
  
 # Выводим информацию о текущей итерации  
 print(f"\nРазбиение n = {n}")  
 print(f"{method\_name}: {integral:.6f}")  
  
 # Проверяем точность  
 if n > 4:  
 error = abs(integral - prev\_integral)  
 print(f"Текущая погрешность: {error:.6f}")  
  
 if error < epsilon:  
 break  
  
 # Увеличиваем число разбиений  
 n \*= 2  
 prev\_integral = integral  
  
 # Вывод итоговых результатов  
 print("\nИтоговые результаты:")  
 print(f"Достигнутая точность: {error:.6f}")  
 print(f"Число разбиений для достижения точности: {n}")  
 print(f"Значение интеграла методом {method\_name}: {integral:.6f}")  
  
 # Вывод таблицы значений для последнего n  
 print("\nТаблица значений для последнего разбиения:")  
 h = (b - a) / n  
 if method\_choice == 3: # Для средних прямоугольников  
 print("i\tx\_i\t\ty\_i\t\tx\_{i-1/2}\t\ty\_{i-1/2}")  
 for i in range(n + 1):  
 x\_i = a + i \* h  
 y\_i = f(x\_i)  
  
 if i > 0:  
 x\_mid = a + (i - 0.5) \* h  
 y\_mid = f(x\_mid)  
 else:  
 x\_mid = ""  
 y\_mid = ""  
  
 print(f"{i}\t{x\_i:.6f}\t{y\_i:.6f}\t{x\_mid if i > 0 else '':<12}\t{y\_mid if i > 0 else '':<12}")  
 elif method\_choice == 5: # Для метода Симпсона  
 print("i\tx\_i\t\ty\_i\tКоэффициент")  
 for i in range(n + 1):  
 x\_i = a + i \* h  
 y\_i = f(x\_i)  
 if i == 0 or i == n:  
 coeff = 1  
 elif i % 2 == 1:  
 coeff = 4  
 else:  
 coeff = 2  
 print(f"{i}\t{x\_i:.6f}\t{y\_i:.6f}\t{coeff}")  
 else: # Для других методов  
 print("i\tx\_i\t\ty\_i")  
 for i in range(n + 1):  
 x\_i = a + i \* h  
 y\_i = f(x\_i)  
 print(f"{i}\t{x\_i:.6f}\t{y\_i:.6f}")  
  
  
# Запуск программы  
import math  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 calculate\_integral()*