

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА



Институт радиоэлектроники и информационных технологий

Кафедра «Вычислительные системы и технологии»

ОТЧЁТ

по лабораторной работе №4

" ОПРЕДЕЛЕННЫЙ ИНТЕГРАЛ"

по дисциплине

Вычислительная математика

(наименование дисциплины)

РУКОВОДИТЕЛЬ:

(подпись)

Панкратова А.З.
(фамилия, и.,о.)

СТУДЕНТ:

(подпись)

Халеев А.А.
(фамилия, и.,о.)

21-ВМз-4
(шифр группы)

Работа защищена «__» _____

С оценкой _____

Нижний Новгород

2023

Тема работы:

Численное интегрирование функций методами прямоугольников, трапеций, Симпсона.

Цель работы:

Изучить численные методы и алгоритмы численного интегрирования

Постановка задачи:

Реализовать изученные алгоритмы численного интегрирования и провести сравнение методов.

Вариант №7:

Вычислить интеграл по формулам центральных (средних) прямоугольников, трапеций и формуле Симпсона, при $n=8$ и $n=20$; оценить погрешность результата.

$$\int_{0.8}^{1.6} \frac{\lg(x^2 + 1)}{x} dx$$

Теория:

Пусть на отрезке a, b задана функция $y = f(x)$. Разобьем отрезок на элементарные отрезки (рис. 1).

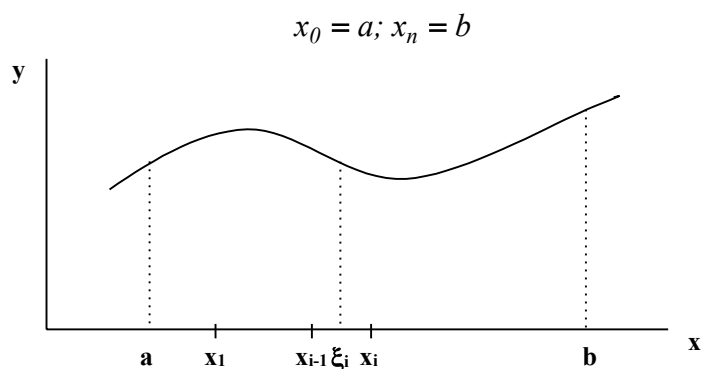


Рис. 1. Разбиение отрезка

На каждом из этих отрезков выберем произвольную точку: $x_{i-1} \leq \varepsilon_i \leq x_i$. Найдем произведение S_i значения функции в точке ε_i на длину элементарного отрезка:

$$S_i = f(\varepsilon_i)(x_i - x_{i-1}) \quad (1)$$

Составим сумму всех таких произведений:

$$S_n = S_1 + S_2 + \dots + S_n = \sum_{i=1}^n f(\varepsilon_i) \Delta x_i \quad (2)$$

S_n – называется интегральной суммой.

Определенным интегралом от функции $f(x)$ на $[a; b]$ называется предел интегральной суммы при неограниченном увеличении числа точек разбиения, или при $\Delta x_i \rightarrow 0$ (максимального из отрезков)

$$\int_a^b f(x) dx = \lim_{\max \Delta x_i \rightarrow 0} \sum_{i=1}^n f(\varepsilon_i) \Delta x_i \quad (3)$$

Геометрический смысл

Выражение (1) при $i=1, 2, \dots, n$ описывает площадь элементарных прямоугольников S_1, S_2, \dots, S_n а выражение (2) интегральной суммы – является суммой всей ступенчатой фигуры (рис. 2).

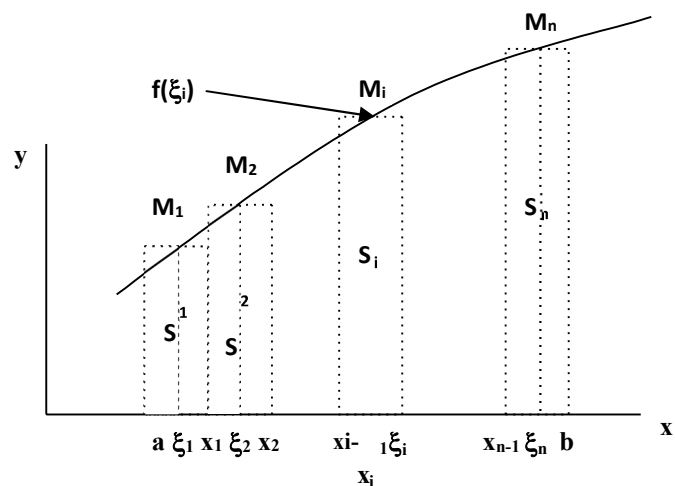


Рис. 2. Геометрический смысл

При неограниченном увеличении числа точек деления или $\Delta x \rightarrow 0$, верхняя граница фигуры (ломаная линия) переходит в кривую $y=f(x)$. Площадь полученной фигуры (криволинейной трапеции) – определенный интеграл.

Во многих случаях, когда подынтегральная функция задана в аналитическом виде, определенный интеграл удастся вычислить непосредственно с помощью определенного (с помощью первообразной) используя формулу Ньютона-Лейбница. Она состоит в том, что определенный интеграл равен приращению первообразной $F(x)$ на отрезке интегрирования $[a;b]$.

$$\int_a^b f(x)dx = F(x) \Big|_a^b = F(b) - F(a) \quad (4)$$

Однако, на практике этой формулой часто нельзя воспользоваться по двум основным причинам:

- вид подынтегральной функции $f(x)$ не допускает непосредственного интегрирования (т.е. первообразную нельзя выразить в элементарной функции);
- значение $f(x)$ задано на фиксированном множестве точек, т.е. в виде таблицы.

Тогда используются методы численного интегрирования. Они основаны на аппроксимации подынтегральной функции (замене ее некоторым более простым выражением).

В дальнейшем будем использовать кусочную (локальную) интерполяцию. Это позволяет приближенно заменить определенный интеграл (3) интегральной суммой (2). В зависимости от способа интерполяции подынтегральные функции различают разные методы численного интегрирования (методы прямоугольников, трапеций, парабол и др.).

Метод центральных прямоугольников

Метод непосредственно использует замену определенного интеграла интегральной суммой. В качестве точек ε_i может выбираться любая точка в промежутке $[x_{i-1}; x_i]$. В зависимости от выбора этой точки различают методы левых, правых и центральных прямоугольников.

Середина интервала, метод центральных прямоугольников:

$$\varepsilon_i = \frac{x_i + x_{i-1}}{2}$$

Обычно, когда рассматривают метод прямоугольников, разбивают $[a, b]$ на n равных отрезков: $\Delta x_i = h = \text{const}$.

Заметим, что в пределах одного шага подынтегральная функция заменяется (аппроксимация) отрезком горизонтальной прямой т.е. первым членом полинома

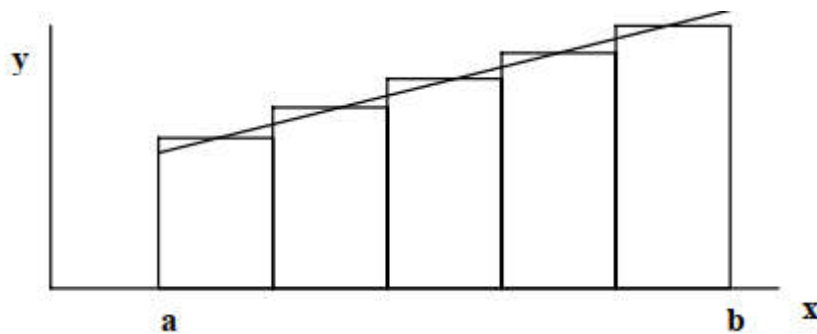
$$y = a_0$$

Коэффициент a_0 ищется из условия прохождения кривой через точки ε_0 . Широко распространенным и более точным является вид формулы прямоугольников, использующий значения функции в средних точках элементарных отрезков, т.е.

$$y(\varepsilon_i) = y_i = y\left(\frac{x_i + x_{i+1}}{2}\right) = y\left(x_{i-\frac{1}{2}}\right)$$

В этом случае формула прямоугольника имеет вид:

$$\int_a^b f(x) dx = h \sum_{i=1}^n y\left(x_{i-\frac{1}{2}}\right), \quad i = 1, 2 \dots n \text{ (для центральных)}$$



Исходные данные:

$$\int_{0.8}^{1.6} \frac{\lg(x^2 + 1)}{x} dx$$

$n1 = 8; n2 = 20$

Решение в Excel:

Вычисление интеграла методом центральных прямоугольников			
Нижний предел	Верхний предел	n	h
0,80	1,60	8,00	0,10
k	x	$\xi_{i-1/2}$	$F_k(x)$
1	0,90	0,85	0,027783439
2	1,00	0,95	0,029402596
3	1,10	1,05	0,030736762
4	1,20	1,15	0,031822237
5	1,30	1,25	0,03269311
6	1,40	1,35	0,033380293
7	1,50	1,45	0,033911158
8	1,60	1,55	0,034309557
		=	0,254039152

Вычисление интеграла методом центральных прямоугольников			
Нижний предел	Верхний предел	n	h
0,80	1,60	20,00	0,04
k	x	$\xi_{i-1/2}$	$F_k(x)$
1	0,84	0,82	0,010894642
2	0,88	0,86	0,011183693
3	0,92	0,9	0,011452381
4	0,96	0,94	0,011701646
5	1,00	0,98	0,011932436
6	1,04	1,02	0,012145699
7	1,08	1,06	0,012342367
8	1,12	1,1	0,012523355
9	1,16	1,14	0,012689554
10	1,20	1,18	0,012841824
11	1,24	1,22	0,01298099
12	1,28	1,26	0,013107846
13	1,32	1,3	0,013223147
14	1,36	1,34	0,013327613
15	1,40	1,38	0,013421925
16	1,44	1,42	0,01350673
17	1,48	1,46	0,013582638
18	1,52	1,5	0,013650223
19	1,56	1,54	0,013710027
20	1,60	1,58	0,01376256
		=	0,253981297

Программа на Python:

```
import math

def func(x: float) -> float:
    return math.log10(x ** 2 + 1) / x # подынтегральная функция

def rectangular_rule(a: float, b: float, n: int) -> float:
    """
    Вычисляет определенный интеграл функции с использованием метода центральных
    прямоугольников.

    Аргументы:
        a (float): Нижний предел интегрирования.
        b (float): Верхний предел интегрирования.
        n (int): Количество интервалов, на которые разбивается область интегрирования.

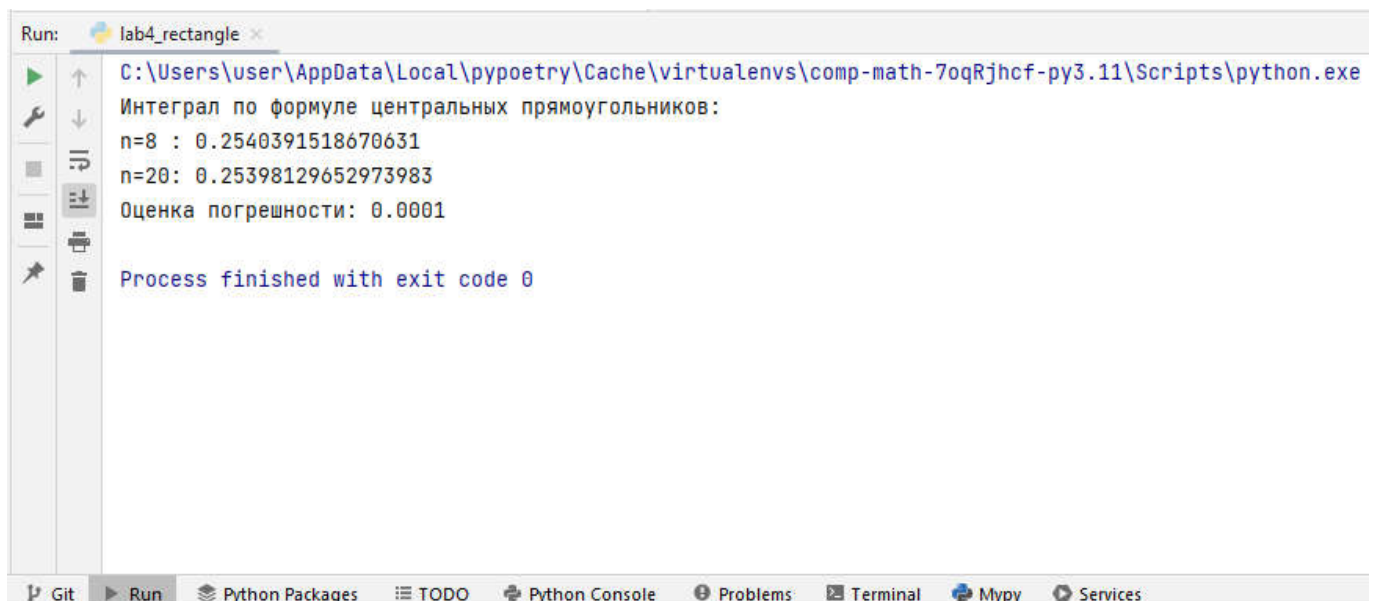
    Возвращает:
        float: Приближенное значение определенного интеграла.
    """
    h = (b - a) / n # шаг
    integral = 0 # аккумулятор значения интеграла

    for i in range(n):
        x = a + h * (i + 0.5) # вычисление значения  $x_{i-1/2}$  с учетом 0-индексированного цикла
        integral += h * func(x) # аккумуляция значения интеграла
    return integral

def main() -> None:
    # Вычисление интегралов
    integral_rectangular_8 = rectangular_rule(a=0.8, b=1.6, n=8)
    integral_rectangular_20 = rectangular_rule(a=0.8, b=1.6, n=20)
    error_rectangular = abs(integral_rectangular_20 - integral_rectangular_8)
    print("Интеграл по формуле центральных прямоугольников:")
    print(f"n=8 : {integral_rectangular_8}\nn=20: {integral_rectangular_20}")
    print(f"Оценка погрешности: {round(error_rectangular, 4)}")

if __name__ == '__main__':
    main()
```

Вывод программы:



```
Run: lab4_rectangle x
C:\Users\user\AppData\Local\py poetry\Cache\virtualenvs\comp-math-7oqRjhcf-py3.11\Scripts\python.exe
Интеграл по формуле центральных прямоугольников:
n=8 : 0.2540391518670631
n=20: 0.25398129652973983
Оценка погрешности: 0.0001
Process finished with exit code 0
```

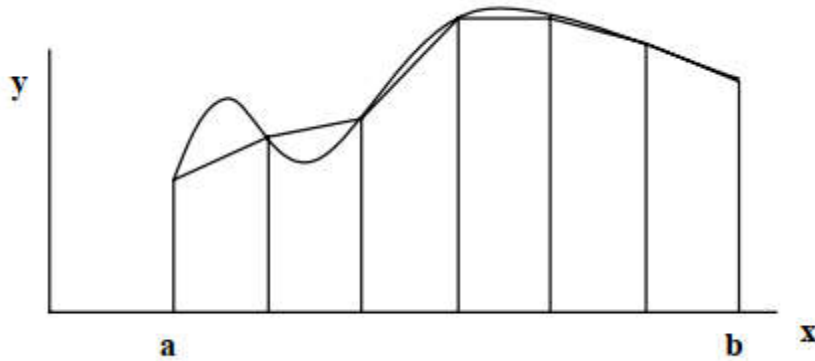

Метод трапеций

Отличается от метода прямоугольников способом аппроксимации отрезка Δx_i . В методе прямоугольников аппроксимация осуществлялась отрезком горизонтальной прямой, а в методе трапеций – прямой общего вида, т.е. полиномом 1-й степени:

$$y = a_0 + a_1 x$$

Коэффициенты вычисляются из условия прохождения этой прямой через две точки (значения подынтегральной функции на краях интервала Δx_i).

График функции $y = f(x)$ представляется в виде ломаной, соединяющей точки (x_i, y_i) :



Площадь всей фигуры складывается из элементарных прямоугольных трапеций. Площадь каждой элементарной трапеции может быть записана в виде:

$$S_i = \frac{y_{i-1} + y_i}{2} \Delta x_i \quad (i = 1, 2, \dots)$$

Площадь всей фигуры:

$$\int_a^b f(x) dx = \frac{1}{2} \sum_{i=1}^n h_i (y_{i-1} + y_i)$$

Если $\Delta x_i = h_i = h = \text{const.}$ (интегрирование с постоянным шагом), то формула трапеций принимает вид:

$$\int_a^b f(x) dx = h \left(\frac{y_0 + y_n}{2} + \sum_{i=1}^n y_i \right)$$

Исходные данные:

$$\int_{0.8}^{1.6} \frac{\lg(x^2 + 1)}{x} dx$$

$$n1 = 8; n2 = 20$$

Решение в Excel:

Вычисление интеграла методом трапеций			
Нижний предел	Верхний предел	n	h
0,80	1,60	8,00	0,10
k	x	Yk(x)	
0	0,80	0,26855481	
1	0,90	0,286309528	
2	1,00	0,301029996	
3	1,10	0,313083885	
4	1,20	0,322824855	
5	1,30	0,330578677	
6	1,40	0,336636936	
7	1,50	0,341255574	
8	1,60	0,344656249	
		=	0,253832498

Вычисление интеграла методом трапеций			
Нижний предел	Верхний предел	n	h
0,80	1,60	20,00	0,04
k	x	Yk(x)	
0	0,80	0,26855481	
1	0,84	0,276044271	
2	0,88	0,283013101	
3	0,92	0,289484556	
4	0,96	0,295482281	
5	1,00	0,301029996	
6	1,04	0,30615123	
7	1,08	0,310869118	
8	1,12	0,315206228	
9	1,16	0,319184442	
10	1,20	0,322824855	
11	1,24	0,326147714	
12	1,28	0,329172374	
13	1,32	0,331917273	
14	1,36	0,334399928	
15	1,40	0,336636936	
16	1,44	0,338643991	
17	1,48	0,340435903	
18	1,52	0,342026627	
19	1,56	0,343429295	
20	1,60	0,344656249	
		=	0,253948226

Программа на Python:

```
import math

def func(x: float) -> float:
    return math.log10(x ** 2 + 1) / x # подынтегральная функция

def trapezoidal_rule(a: float, b: float, n: int) -> float:
    """
    Вычисляет определенный интеграл функции с использованием метода трапеций.

    Аргументы:
        a (float): Нижний предел интегрирования.
        b (float): Верхний предел интегрирования.
        n (int): Количество интервалов, на которые разбивается область интегрирования.

    Возвращает:
        float: Приближенное значение определенного интеграла.
    """
    h = (b - a) / n # шаг
    integral = 0 # аккумулятор значения интеграла


    for i in range(n + 1):
        x = a + i * h # вычисление значения x на основании шага и индекса текущей итерации
        if i == 0 or i == n:
            integral += func(x) / 2 # первое и последнее значение делятся на 2
        else:
            integral += func(x) # остальные значения прибавляются без изменений

    integral *= h # умножение на шаг
    return integral

def main() -> None:
    # Вычисление интегралов
    integral_trapezoidal_8 = trapezoidal_rule(a=0.8, b=1.6, n=8)
    integral_trapezoidal_20 = trapezoidal_rule(a=0.8, b=1.6, n=20)
    error_rectangular = abs(integral_trapezoidal_20 - integral_trapezoidal_8)
    print("Интеграл по формуле трапеций:")
    print(f"n=8 : {integral_trapezoidal_8}\nn=20: {integral_trapezoidal_20}")
    print(f"Оценка погрешности: {round(error_rectangular, 4)}")

if __name__ == '__main__':
    main()
```

Вывод программы:



The screenshot shows a Python IDE window titled "lab4_trapezoidal". The output console displays the following text:

```
Run: C:\Users\user\AppData\Local\pypoetry\Cache\virtualenvs\comp-math-7oqRjhcf-py3.11\Scripts\python.exe
Интеграл по формуле трапеций:
n=8 : 0.25383249805248165
n=20: 0.2539482259745291
Оценка погрешности: 0.0001
Process finished with exit code 0
```

The IDE interface includes a sidebar with icons for Run, Python Packages, TODO, Python Console, Problems, Terminal, Mypy, and Services. The bottom status bar shows the current mode as "Run".

Метод Симпсона

Разобьем отрезок $[a, b]$ на четное число n равных частей с шагом h на каждом отрезке.

Возьмем отрезки, равные двум шагам: $[x_0, x_2]$, $[x_2, x_4]$, ..., $[x_{i-1}, x_{i+1}]$, ..., $[x_{n-2}, x_n]$. На каждом из них подынтегральную функцию заменим интерполяционным полиномом второй степени:

$$f(x) = \varphi_i(x) = a_i x^2 + b_i x + c$$

$$x_{i-1} \leq x \leq x_{i+1} \text{ (на каждом отрезке)}$$

Коэффициенты a_i , b_i , c_i могут быть найдены из условий равенства многочлена в точках x_i соответствующим табличным значениям $f(x_i)$

В качестве $\varphi_i(x)$ можно взять интерполяционный многочлен Лагранжа, проходящий через точки

$$M_{i-1}(x_{i-1}, y_{i-1}); M_{i-1}(x_i, y_i); M_{i-1}(x_{i+1}, y_{i+1})$$

Элементарная функция S_i - площадь криволинейной трапеции (рис. 8) вычисляется с помощью определенного интеграла, подынтегральной функцией которого является многочлен Лагранжа:

$$S_i = \int_{x_{i-1}}^{x_{i+1}} \varphi_i(x) dx = \frac{1}{2h^2} \int_{x_{i-1}}^{x_{i+1}} [(x - x_i)(x - x_{i+1})y_{i-1} - 2(x - x_{i-1})(x - x_{i+1})y_i + (x - x_{i-1})(x - x_i)y_{i+1}] dx$$

$$S_i = \frac{h}{3}(y_{i-1} + 4y_i + y_{i+1})$$

$$S = \frac{h}{3}(y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + \dots + 2y_{n-2} + 4y_{n-1} + y_n)$$

Данное выражение принимается в качестве значения определенного интеграла:

$$\int_a^b f(x) dx = \frac{h}{3}(y(a) + 4y(a+h) + 2y(a+2h) + 4y(a+3h) + \dots + 2y(b-2h) + 4y(b-h) + y(b))$$

Это формула Симпсона

Исходные данные:

$$\int_{0.8}^{1.6} \frac{\lg(x^2 + 1)}{x} dx$$

$$n1 = 8; n2 = 20$$

Решение в Excel:

При $n=8$:

$$I_{\text{симпсона}} = \frac{0.1}{3} (y_0 + 4(y_1 + y_3 + y_5 + y_7) + 2(y_2 + y_4 + y_6) + y_8) = 0.25397$$

Вычисление интеграла методом Симпсона			
Нижний предел	Верхний предел	n	h
0,80	1,60	8,00	0,10
k	x	Yk(x)	
0	0,80	0,26855481	
1	0,90	0,286309528	
2	1,00	0,301029996	
3	1,10	0,313083885	
4	1,20	0,322824855	
5	1,30	0,330578677	
6	1,40	0,336636936	
7	1,50	0,341255574	
8	1,60	0,344656249	
		=	0,253970176

При $n=20$:

$$I_{\text{симпсона}} = \frac{0.04}{3} \left(y_0 + 4(y_1 + y_3 + y_5 + y_7 + y_9 + y_{11} + y_{13} + y_{15} + y_{17} + y_{19}) + 2(y_2 + y_4 + y_6 + y_8 + y_{10} + y_{12} + y_{14} + y_{16} + y_{18}) + y_{20} \right) = 0.25397$$

Вычисление интеграла методом Симпсона			
Нижний предел	Верхний предел	n	h
0,80	1,60	20,00	0,04
k	x	Yk(x)	
0	0,80	0,26855481	
1	0,84	0,276044271	
2	0,88	0,283013101	
3	0,92	0,289484556	
4	0,96	0,295482281	
5	1,00	0,301029996	
6	1,04	0,30615123	
7	1,08	0,310869118	
8	1,12	0,315206228	
9	1,16	0,319184442	
10	1,20	0,322824855	
11	1,24	0,326147714	
12	1,28	0,329172374	
13	1,32	0,331917273	
14	1,36	0,334399928	
15	1,40	0,336636936	
16	1,44	0,338643991	
17	1,48	0,340435903	
18	1,52	0,342026627	
19	1,56	0,343429295	
20	1,60	0,344656249	
		=	0,253970271

Программа на Python:

```
import math

def func(x: float) -> float:
    return math.log10(x ** 2 + 1) / x # подынтегральная функция

def simpson_rule(a: float, b: float, n: int) -> float:
    """
    Вычисляет определенный интеграл функции с использованием правила Симпсона.

    Параметры:
        a (float): Нижний предел интегрирования.
        b (float): Верхний предел интегрирования.
        n (int): Количество подинтервалов. Должно быть четным числом.

    Возвращает:
        float: Приближенное значение определенного интеграла.
    """
    if n % 2 != 0:
        raise ValueError("число n должно быть четным")

    h = (b - a) / n # шаг
    integral = 0 # аккумулятор значения интеграла

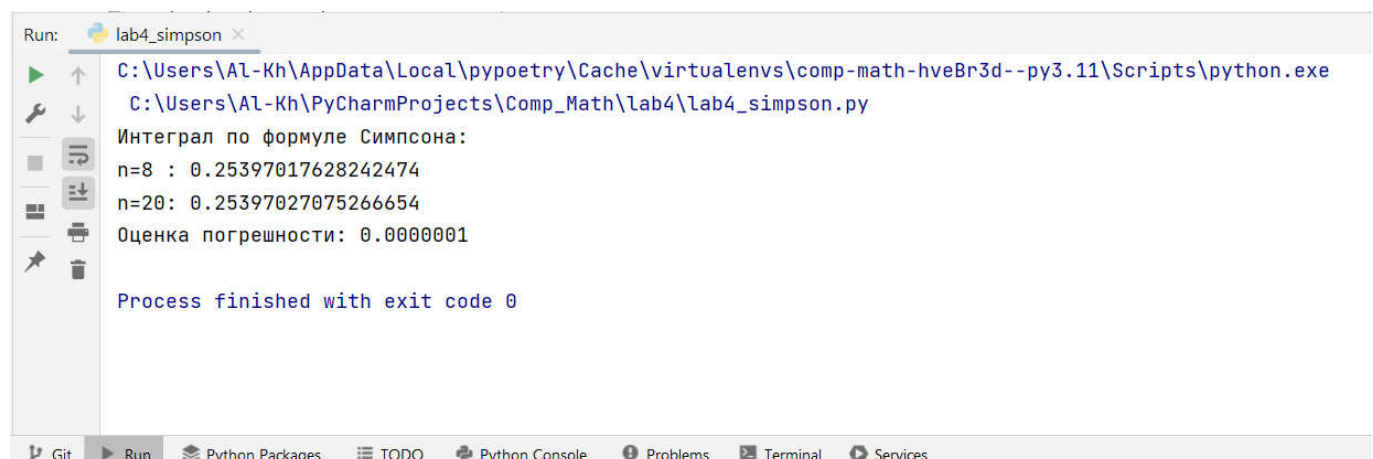
    for i in range(n + 1):
        x = a + i * h # вычисление значения x на основании шага и индекса текущей итерации
        if i == 0 or i == n:
            integral += func(x) # первый и последний индекс без множителя
        elif i % 2 == 0:
            integral += 2 * func(x) # для четных индексов множитель = 2
        else:
            integral += 4 * func(x) # для нечетных индексов множитель = 4

    integral *= h / 3 # умножаем полученную сумму на коэффициент (шаг деленный на 3)
    return integral

def main() -> None:
    # Вычисление интегралов
    integral_simpson_8 = simpson_rule(a=0.8, b=1.6, n=8)
    integral_simpson_20 = simpson_rule(a=0.8, b=1.6, n=20)
    error_rectangular = abs(integral_simpson_20 - integral_simpson_8)
    print("Интеграл по формуле Симпсона:")
    print(f"n=8 : {integral_simpson_8}\nn=20: {integral_simpson_20}")
    print(f"Оценка погрешности: {error_rectangular:.7f}")

if __name__ == '__main__':
    main()
```

Вывод программы:



The screenshot shows a Python IDE window titled "lab4_simpson". The output console displays the following text:

```
Run: C:\Users\Al-Kh\AppData\Local\pypoetry\Cache\virtualenvs\comp-math-hveBr3d--py3.11\Scripts\python.exe
C:\Users\Al-Kh\PyCharmProjects\Comp_Math\lab4\lab4_simpson.py
Интеграл по формуле Симпсона:
n=8 : 0.25397017628242474
n=20: 0.25397027075266654
Оценка погрешности: 0.0000001
Process finished with exit code 0
```

The IDE interface includes a sidebar with icons for Run, Python Packages, TODO, Python Console, Problems, Terminal, and Services. The bottom status bar shows the current file is "lab4_simpson.py".

Вывод:

В данной лабораторной работе были изучены следующие методы численного решения определенных интегралов:

- метод прямоугольников
- метод трапеций
- метод Симпсона

Были получены следующие значения точности вычислений при разбиении области интегрирования на 8 и 20 равных частей:

Метод прямоугольников : 0.0001

Метод трапеций : 0.0001

Метод Симпсона : 0.0000001

Таким образом, метод Симпсона является наиболее точным, поэтому его чаще всего используют при работе на ЭВМ. На ЭВМ есть стандартные программы, вычисляющие значения определенного интеграла методом Симпсона