Лабораторная работа №1

Аналитический Метод

Заданная функция

function

1.1 Докажите, что f измерима по Лебегу на E

Рассмотрим измеримость f по определению. Исходя из факта, о равносильности измеримости Лебеговых множеств, проверим с помощью множеств вида $\{f \geq a\}_{a \in \mathbb{R}}$

$$a > \pi \Leftrightarrow \{f \geq a\} = \emptyset \rightarrow \text{измеримо}$$

$$a \in (3; \pi] \Leftrightarrow \{f \geq a\} = \{0\} \rightarrow \text{измеримо}$$

$$a \in (2; 3] \Leftrightarrow \{f \geq a\} = \{0, 3\} \rightarrow \text{измеримо}$$

$$a \in (1; 2] \Leftrightarrow \{f \geq a\} = [2, 3] \cup \{0\} \rightarrow \text{измеримо}$$

$$a \in (0.8; 1] \Leftrightarrow \{f \geq a\} = [1, 3] \cup \{0\} \rightarrow \text{измеримо}$$

$$a \in (0; 0.8]$$

- имеем множество случаев, рассмотрим каждый из них

1: $a = 0, 0 \ldots 0a_k$ (до a_k идет $k > 0$ нулей; причем после a_k идут только нули)

1.1: $a_k < 5 \rightarrow \{f \geq a\} = [a; 3] \cup \{0\} - \text{измеримо}$

1.2: $5 \leq a_k < 9 \rightarrow \{f \geq a\} = [0, 0 \ldots 0(a_k + 1); 3] \cup \{0\} - \text{измеримо}$

1.3:
$a_k = 9 \to \{f \geq a\} = [0,0\dots01;3] \quad (k-1 \text{ нулей}) \quad \cup \{0\} - \text{измеримо}$

2: $a = 0,0\dots0a_k\dots a_m$ (разница с первым пунктом в том, что после a_k у нас может появиться еще одна цифра [к примеру a_m])

2.1: $a_k < 4 \to \{f \geq a\} = [0,0\dots0(a_k+1);3] \cup \{0\} - \text{измеримо}$

2.2:
$4 \leq a_k < 8 \to \{f \geq a\} = [0,0\dots0(a_k+2);3] \cup \{0\} - \text{измеримо}$

2.3:
$a_k \geq 8 \to \{f \geq a\} = [0,0\dots01;3] \quad (k-1 \text{ нулей}) \quad \cup \{0\} - \text{измеримо}$

$$a \in (0;0.8] \Leftrightarrow \{f \geq a\} - \text{измеримо}$$

$$a \leq 0 \Leftrightarrow \{f \geq a\} = [0;3] - \text{измеримо}$$

Значит, что

$$\forall a \in \mathbb{R} \to \{f \geq a\} - \text{измеримо} \quad \Rightarrow$$

$$\Rightarrow \text{все Лебеговы множества измеримы} \Rightarrow$$

$$\Rightarrow f - \text{измеримо по определению}$$

1.2 Постройте последовательность простых ф-ий $f_n(x)$ такую, чтобы $f_n \leq f$, $f_n \to f$ почти всюду на E

$$f_n = \begin{cases} \pi, & \text{for } x = 0 \\ 1, & \text{for } x \in [1;2) \\ 2, & \text{for } x \in [2;3) \\ 3, & \text{for } x = 3 \\ f(x), \quad \text{for } x = 0,0\dots a_k\dots & (a_k > 0, k \le n) \\ 0, & \text{otherwise} \end{cases}$$

Знаем что

$$\forall n \in \mathbb{N}, \text{ на отрезке } [1;3] : f_n = f$$

$$\forall n \in \mathbb{N} : f_n(0) = f(0)$$

$$x \in (0;1) \Leftrightarrow f_n = f(x) \text{ или } f_n = 0 \Rightarrow f_n \le f \Rightarrow \forall x \in E : f_n \le f$$

$$n \to \infty \Rightarrow \text{мера } X = \{\, x \,|\, f_n(x) \ne f(x)\} \to 0 \Rightarrow x \in (0;1) \Leftrightarrow f_n \to f \text{ т}$$

$$\text{на } [1;3] \cup \{0\} \Rightarrow f_n = f, \text{ означает } f_n \to f \text{ на } E$$

Также

$$f_{n+1} > f_n$$

так как f_{n+1} отличается от f_n только в тех точках, где $f_n = 0$, вдовесок мы знаем что f_{n+1} доопределено в этих точках "≥ 0" значениями $\Rightarrow f_n$ - возрастающая последовательность

1.3 Запишите определение интеграла Лебега функции f по E, используя построенную последовательность f_n. Вычислите аналитически, ссылаясь на соответствующие теоремы

$$\int f \, d\lambda = \sup\left\{ \int g \, d\lambda, \; 0 \le g \le f, \; g - \text{простая} \right\}$$

где

$$\int_E g\,d\lambda = \sum_{i=1}^{N} c_i\,\lambda E_i \quad E = \bigsqcup_{i=1}^{N} E_i$$

$$f_n \to f; f_n - \text{измеримо}; 0 \le f_n \le f; f_n - \text{возрастающая} \Rightarrow$$

по теореме Леви

$$\Rightarrow \int_E f d\mu = \lim_{n\to\infty} \int_E f_n d\mu$$

$$\int_E f d\mu = \int_{[0,1)} f d\mu + \int_{[1,2)} f d\mu + \int_{[2,3)} f d\mu + \int_{\{3\}} f d\mu$$

$$\int_{[1,2)} f d\lambda = 1 \cdot \lambda([1;2)) = 1$$

т.к на этом полуинтервале, f - простая функция

Аналогично

$$\int_{[2,3)} f d\lambda = 2 \cdot \lambda([2;3)) = 2$$

Также

$$\int_{\{3\}} f d\lambda = 0 \Leftarrow \lambda(\{3\}) = 0$$

Нужно посчитать

$$\int_{[0,1)} f d\lambda$$

Чтобы применить т. Леви, в первую очередь нужно доказать что f_n - измеримая $\Rightarrow \{f \geq a\}_{\forall a \in \mathbb{R}} -$ докажем что измеримое

Доказательство аналогично доказательству измеримости f, кроме момента с $a := 0,0\dots0a_k \Leftrightarrow$ для таких a - $\{f \geq a\}-$ совпадает с уже измеримым множеством
$\{f \geq 0,0\dots01\}, (n \text{ нулей}) \Rightarrow f_n - \text{измеримо } \forall n \in \mathbb{N}$

Используем теорему Леви

$$\int_{[0,1)} f_n d\lambda + \int_{\{0\}} f_n d\lambda + \int_{(0,1)} f_n d\lambda = \int_{(0,1)} f_n d\lambda =$$

$$= \sum_{j=1}^{n} \frac{1}{10^j} \cdot (1+2+3+4+4+5+6+7+8) \cdot \frac{1}{10^j} = 40 \sum_{j=1}^{n} \frac{1}{10^{2j}}$$

$$\lim_{n\to\infty} 40 \sum_{j=1}^{n} \frac{1}{10^{2j}} = 40 \cdot \lim_{n\to\infty} \sum_{j=1}^{n} \frac{1}{10^{2j}} = 40 \cdot \frac{1}{10^2} \cdot \frac{1}{1-\frac{1}{10^2}} = \frac{40}{99} \Rightarrow$$

$$\Rightarrow \int_E f \, d\lambda = 3 + \frac{40}{99} = \frac{337}{99}$$

1.4 Докажите (опираясь на соответствующие теоремы), что данная функция F задает меру Лебега-Стилтьеса на μ_F

Требуется доказать что функция возрастающая и непрерывная слева

$3x$ возрастающая

$[-2x]$ убывает $\Rightarrow -[-2x]$ возрастающая

$\Rightarrow 3x - [-2x]$ возрастающая

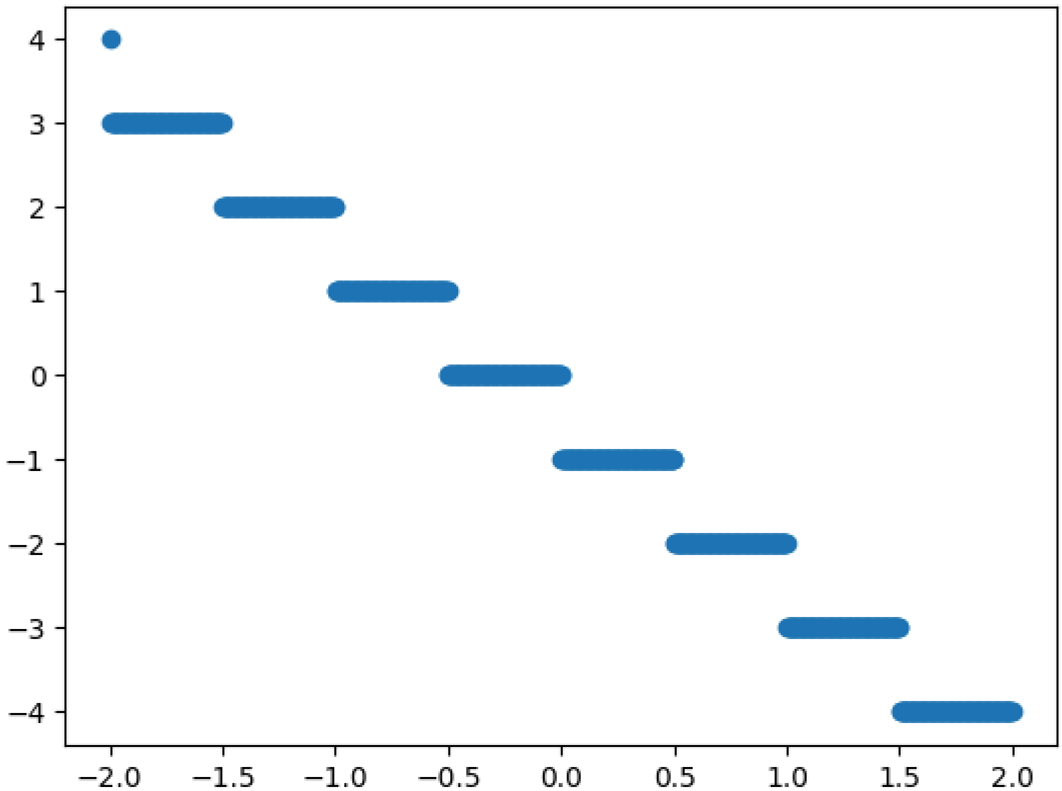
$3x$ - очевидно непрерывная слева

$[-2x]$ - непрерывная слева, можно увидеть на графике

In [2]:
```
def func(x):
    return np.floor(-2*x)
```

In [10]:
```
X = np.linspace(-2, 2, 1000)
y = func(X)

plt.scatter(X, y)
```

Out[10]: <matplotlib.collections.PathCollection at 0x12d404950>



$F(x) = 3x - [-2x]$ - непрерывная функция $\Rightarrow$ задает меру Лебега-Стилтьеса

1.5 Вычислите интеграл Лебега-Стилтьеса

$$\int_E f\, d\mu_F$$

аналитически

$$\int_E f d\mu_F = \int_{[0,1)} f d\mu_F + \int_{[1,2)} f d\mu_F + \int_{[2,3)} f d\mu_F + \int_{\{3\}} f d\mu_F$$

$$\int_{[1,2)} f d\mu_F = 1 \cdot \mu_F([1;2)) = 1 \cdot (F(2) - F(1)) = 5$$

$$\int_{[2,3)} f d\mu_F = 2 \cdot \mu_F([2;3)) = 2 \cdot (F(3) - F(2)) = 10$$

$$\int_{\{3\}} f d\mu_F = 3 \cdot \mu_F(\{3\}) = 3 \cdot (F(3^+) - F(3)) = 3$$

$$\int_{[0,1)} f d\mu_F = \int_{\{0\}} f d\mu_F + \int_{(0,1)} f d\mu_F = \pi \cdot (F(0^+) - F(0)) + \int_{(0,1)} f d\mu_F = \pi + \int_{(0,1)}$$

Рассмотрим

$$\int_{(0,1)} f d\mu_F = \int_{(0,0.1)} f d\mu_F + \int_{[0.1,1)} f d\mu_F$$

Так как $\forall x \in (0; 0.1) : [-2x] = -1 \Rightarrow \mu_F(\text{ячейки}) = 3 \cdot \lambda(\text{ячейки})$

По теореме Леви

$$\int_{(0,0.1)} f d\mu_F = \lim_{n\to\infty} \int_{(0,0.1)} f_n d\mu_F = \sum_j \frac{3}{10^j} \cdot (1+2+3+4+4+5+6+7+8) \cdot \frac{}{1}$$

$$\int_{[0.1,1)} f d\mu_F = \lim_{n\to\infty} \int_{[0.1,1)} f_n d\mu_F = \frac{3}{10} \cdot \frac{1}{10} + \frac{3}{10} \cdot \frac{2}{10} + \frac{3}{10} \cdot \frac{3}{10} + \frac{3}{10} \cdot \frac{4}{10} +$$

$$+ \frac{4}{10} \cdot \mu([0.5; 0.6)) + \frac{3}{10} \cdot \frac{5}{10} + \frac{3}{10} \cdot \frac{6}{10} + \frac{3}{10} \cdot \frac{7}{10} + \frac{3}{10} \cdot \frac{8}{10} = \frac{8}{5}$$

Следовательно

$$\int_E f\, d\mu_F = \pi + \frac{12}{990} + \frac{8}{5} + 18$$

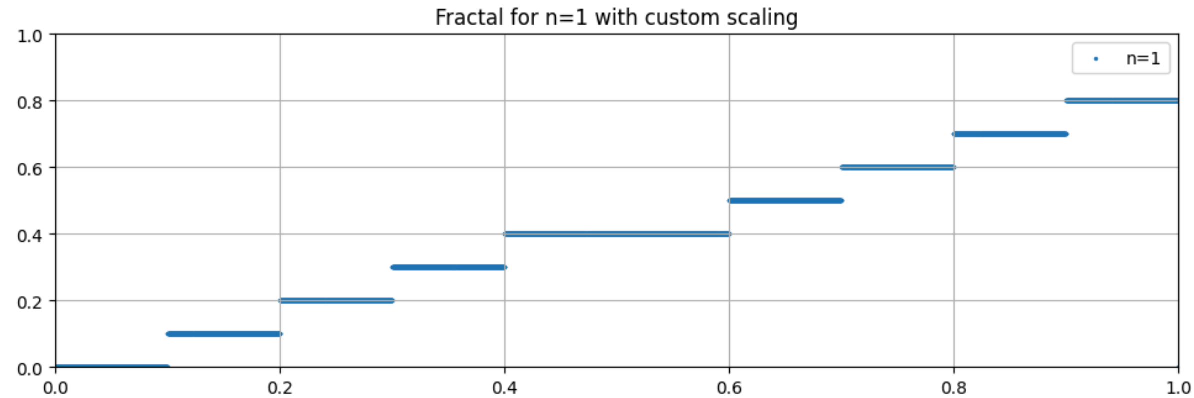
Численный метод

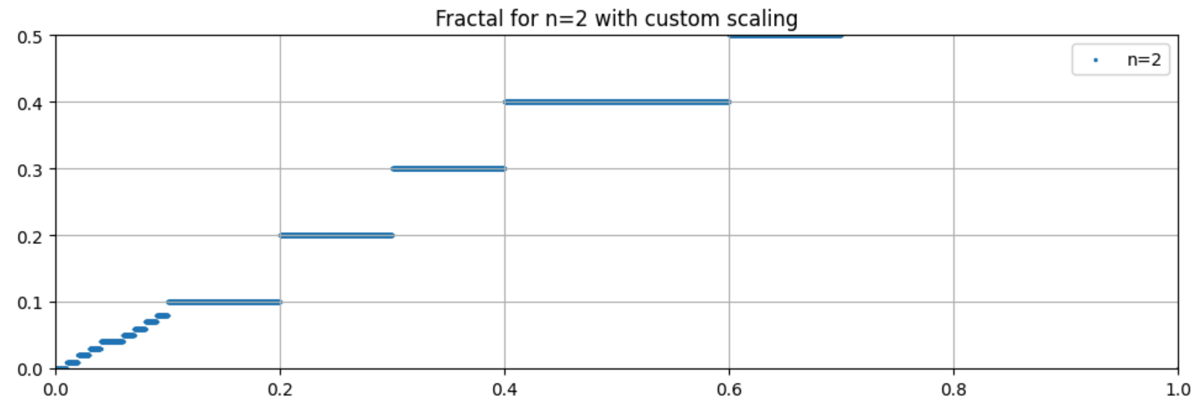
In [34]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sympy
from sympy import limit, symbols
import time
from functools import wraps
from decimal import *
```

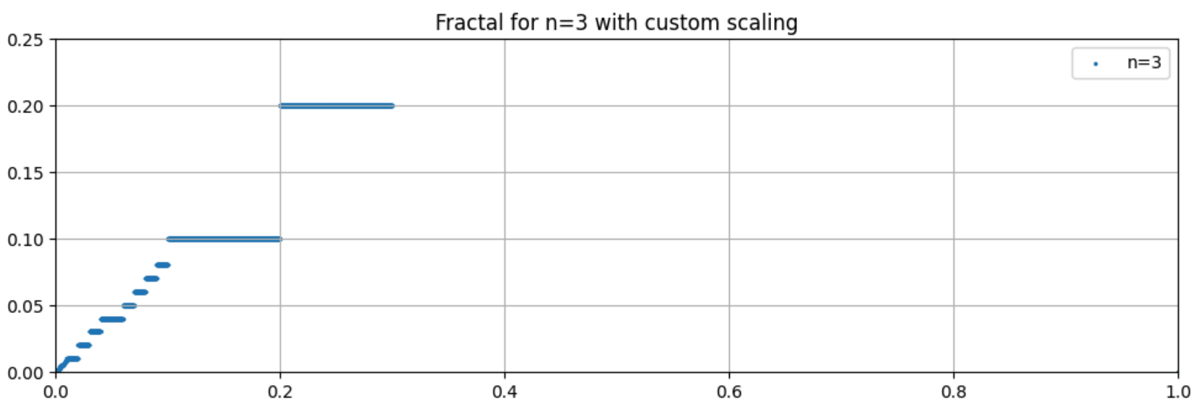
2.1 Изобразите графики, f_n при нескольких значениях n или анимированный график (при увеличении n)

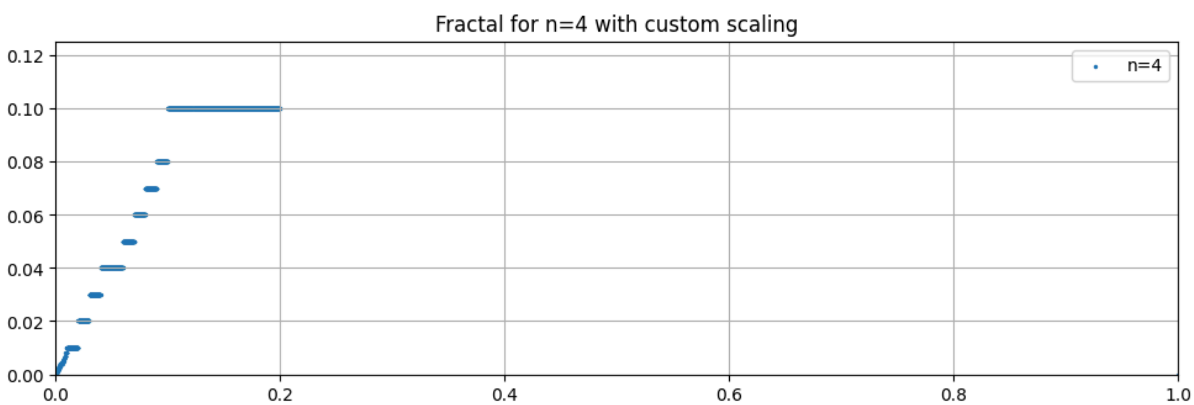
In [35]:
```
def function_formula(x, L, variant):
    if variant == 1:
        condition = (4 >= np.floor(x * 10**L)) & (np.floor(x * 10**L) > 0)
        return condition * (np.floor(x * 10**L) / 10**L)
    elif variant == 2:
        condition = (9 >= np.floor(x * 10**L)) & (np.floor(x * 10**L) > 4)
        return condition * (np.floor(x * 10**L - 1) / 10**L)
```

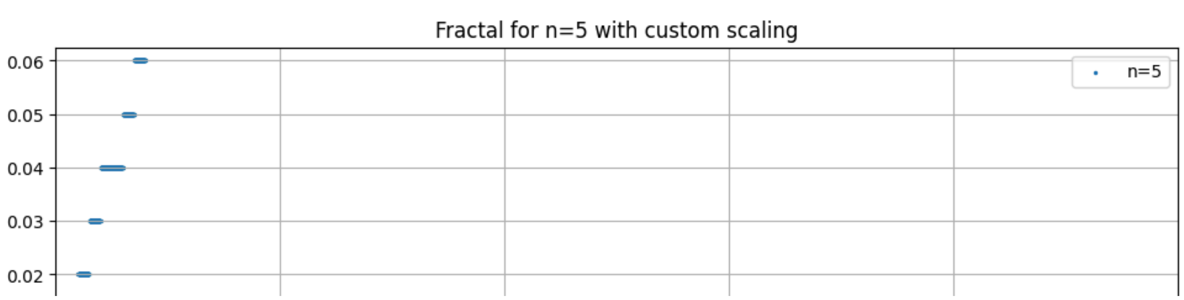
In [36]:
```
def generate_fractal(n, x_range, num_points):
    x = np.linspace(*x_range, num_points)
    y = np.zeros_like(x)

    for L in range(1, n + 1):
        y += function_formula(x, L, 1) + function_formula(x, L, 2)

    return x, y

x_range = (0, 1)
num_points = 10000

n_values = [1, 2, 3, 4, 5, 10]
fig, axes = plt.subplots(len(n_values), 1, figsize=(10, 20))

scaling_factors = [1, 2, 4, 8, 16, 32]

assert len(scaling_factors) == len(n_values)

for i, n in enumerate(n_values):
    x, y = generate_fractal(n, x_range, num_points)
    axes[i].scatter(x, y, label=f'n={n}', s=2)
    axes[i].set_xlim(0, 1)
    axes[i].set_ylim(0, 1 / scaling_factors[i])
    axes[i].set_title(f'Fractal for n={n} with custom scaling')
    axes[i].legend()
    axes[i].grid(True)

plt.tight_layout()
plt.show()
```

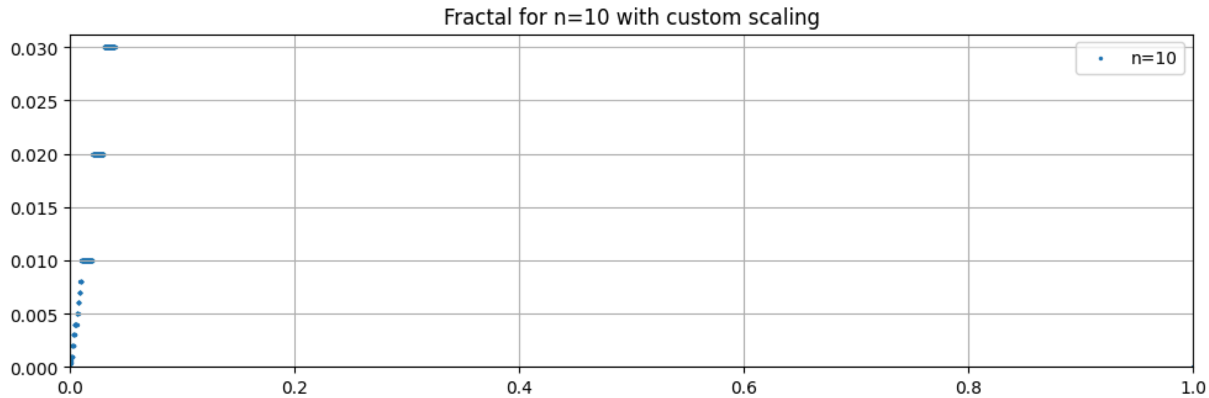

Fractal for n=1 with custom scaling
n=1


Fractal for n=2 with custom scaling
n=2


Fractal for n=3 with custom scaling
n=3


Fractal for n=4 with custom scaling
n=4


Fractal for n=5 with custom scaling
n=5



2.2 & 2.3 Вычислите интеграл Лебега от f_n по E для нескольких (больших) значений n. Сравните результат с аналитическим. Вычислите интеграл Лебега-Стилтьеса для функции f_n при нескольких (больших) значениях n. Сравните результат с аналитическим

In [38]:

```
# Задаем константы
getcontext().prec = 15

# Определение функций и глобальных переменных
x = symbols('x')
measure_functions = {
    "Lebegue": x,
    "Lebegue-Stieltjes": 3 * x - sympy.floor(-2 * x)
}
real_values = {
    "Lebegue": 337 / 99,
    "Lebegue-Stieltjes": np.pi + 18 + 8 / 5 + 12 / 990
}

intervals = [(0, 0, True), (1, 2, True), (2, 3, True), (3, 3, True)]
significant_digit_numbers = [10, 100, 200]
number_range = range(1, 10)
```

In [39]:

```
# Зададим функцию
def f_n(x):
    match x:
        case 0:
            return np.pi
        case 3:
            return 3
        case _ if 1 <= x < 2:
            return 1
        case _ if 2 <= x < 3:
            return 2
        case _:
            check_num = str(x)
```

```python
            for i, char in enumerate(check_num):
                if char.isdigit() and (new_number := int(char)) > 0:
                    if new_number > 4:
                        new_number -= 1
                    return Decimal(check_num[:i] + str(new_number))
            return 0
```

In [41]:
```python
# Считаем меру ячеек, точек и интервалов
def calculate_interval_measure(interval_simple, measure):
    left_point_measure = limit(measure, x, interval_simple[0], "+") - measur
    if interval_simple[0] == interval_simple[1]:
        return left_point_measure
    cell_measure = measure.subs(x, interval_simple[1]) - measure.subs(x, int
    return cell_measure if interval_simple[2] else cell_measure - left_point


# Считающаем значение интеграла простой ф-ции
def calculate_integral_simple(interval_simple, measure, function):
    interval_measure = Decimal(str(calculate_interval_measure(interval_simpl
    mid_point = (Decimal(interval_simple[0]) + Decimal(interval_simple[1]))
    return interval_measure * Decimal(function(mid_point))


def profile_function(func):  #  Функция-декоратор для профилирования вызовов
    stats = {
        'call_count': 0,
        'total_time': 0
    }

    @wraps(func)
    def wrapper(*args, **kwargs):
        stats['call_count'] += 1
        start_time = time.time()
        result = func(*args, **kwargs)
        end_time = time.time()
        elapsed_time = end_time - start_time
        stats['total_time'] += elapsed_time
        print(f"Function {func.__name__} call {stats['call_count']}: {elapse
        return result

    wrapper.stats = stats
    return wrapper


@profile_function
def compute_integral(option, measure_func):
    results = []
    for number in significant_digit_numbers:
        integral = sum(calculate_integral_simple(interval, measure_func, f_r
        integral += sum(
            sum(
                calculate_integral_simple(
                    (Decimal(l) * Decimal(10) ** Decimal(-i), Decimal(l + 1)
                    measure_func,
                    f_n
                ) for l in number_range
```

```
            ) for i in range(1, number + 1)
        )
        diff = integral - Decimal(real_values[option])
        results.append({
            'option': option,
            'n': number,
            'integral': integral,
            'analytical_value': real_values[option],
            'diff': diff
        })
    return results


# Сборка данных в DataFrame
all_results = []
for option, measure_func in measure_functions.items():
    all_results.extend(compute_integral(option, measure_func))

df = pd.DataFrame(all_results)

# Вывод статистики профилирования
print(f"Function {compute_integral.__name__} was called {compute_integral.st
print(f"Total time spent in {compute_integral.__name__}: {compute_integral.s
```

```
Function compute_integral call 1: 0.392600 seconds
Function compute_integral call 2: 8.860082 seconds
Function compute_integral was called 2 times
Total time spent in compute_integral: 9.252682 seconds
```

In [42]:
```
df
```

Out[42]:

	option	n	integral	analytical_value	diff
0	Lebegue	10	3.40404484804000	3.404040	0.00000444399959602240
1	Lebegue	100	3.40404484848444	3.404040	0.00000444444403602240
2	Lebegue	200	3.40404484848444	3.404040	0.00000444444403602240
3	Lebegue-Stieltjes	10	22.7537271977098	22.753714	0.0000133319987937077
4	Lebegue-Stieltjes	100	22.7537271990431	22.753714	0.0000133333320937077
5	Lebegue-Stieltjes	200	22.7537271990431	22.753714	0.0000133333320937077

In [ ]: