

Отчет по лабораторной работе № 1

Вариант № 3

Винницкая Дина Сергеевна

Группа: Б9122-02-03-01сст

Цель работы

1. Построить интерполяционный многочлен Лагранжа для заданной функции.
2. Построить таблицу абсолютной и относительной погрешностей остаточного члена для каждой n .
3. Построить график зависимостей $\Delta f_n(n)$, $r_n(n)$.
4. Сделать вывод.

Входные данные:

1. **Функция:** $y = x^2 + \ln(x) - 4$
2. **[a;b]:** [1.5, 2.0]
3. $n \in 3, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100$

Ход работы:

1 Используемые библиотеки

Для реализации необходимой программы используются следующие библиотеки языка Python:

- **numpy** - библиотека для работы с массивами, векторами и матрицами, включающая функции для линейной алгебры, преобразования Фурье и генерации псевдослучайных чисел.
- **math** - модуль для выполнения математических операций, таких как вычисление факториала и другие математические функции.
- **pandas** - библиотека для обработки и анализа данных, предоставляющая удобные структуры данных и операции для их анализа.
- **matplotlib** - библиотека для визуализации данных, позволяющая создавать различные виды графиков, диаграмм и других визуальных элементов.

```
1 from numpy import linspace
2 from math import factorial
3 import pandas as pd
4 import numpy as np
```

2 Определение функции и нахождение производной

```
1 def function(x: float):
2     return x ** 2 + np.log(x) - 4
3
4 def derivative_function(x: float, n: int = 2) -> float:
5     return 2 - 1 / (x ** 2)
6
7 value_range = [3, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
8
9 graph_range = (1.5, 2.0)
```

В коде представлено определение двух функций, одна из которых - исходная функция, вторая же, представляет собой ее производную, списка точек и кортежа значений заданного промежутка.

3 Графическое отображение функции

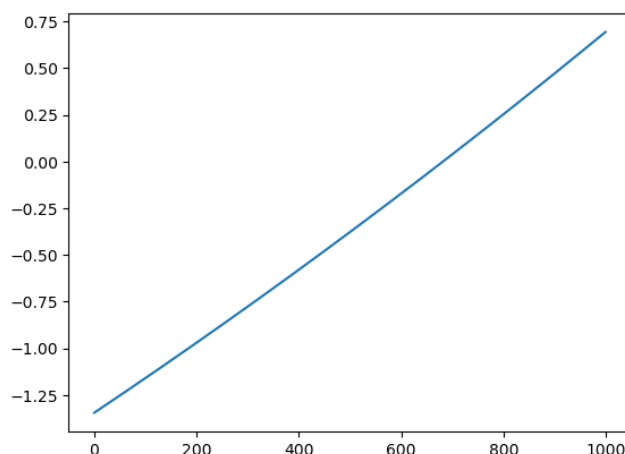


Рис. 1: Отображение функции

```
1 import matplotlib.pyplot as plt
2
3 plt.plot(function(linspace(*graph_range, num=10 ** 3)))
```

4 Реализация полинома Лагранжа

```
1 def generate_points(rng: tuple[float, float], count_points: int, function)
2   -> list[tuple[float, float]]:
3
4     result = []
5     for current_position in linspace(*rng, count_points):
6         result.append((current_position, function(current_position)))
7     return result
8
9 def lagrange_interpolation(bp: float, points: list[tuple[float, float]])
10  -> float:
11
12     count_points = len(points)
13     result = 0
14     for k, point in enumerate(points):
15         multiply = point[1]
16         for j in range(0, k - 1 + 1):
17             x = points[j][0]
18             multiply *= ((bp - x) / (point[0] - x))
19         for i in range(k + 1, count_points):
20             x = points[i][0]
21             multiply *= ((bp - x) / (point[0] - x))
22         result += multiply
23     return result
```

- Функция `lagrange_interpolation` принимает базовую точку `bp` и список кортежей значений `points`, и возвращает значение интерполяции методом Лагранжа. Внутри функции выполняется итерация по всем точкам из списка `points`. Для каждой точки выполняются вычисления, используя формулу интерполяции Лагранжа, и результаты суммируются для получения окончательного значения интерполяции.

5 Вычисление ошибок

Норма

```
1 def calculate_max_absolute_value(function, rng: tuple[float, float],
2 *args) -> float:
3
4     return max(abs(function(linspace(*rng, num=10 ** 3), *args)))
```

- Функция `calculate_max_absolute_value` принимает функцию `function`, диапазон `rng` и дополнительные аргументы `*args`, и возвращает максимальное абсолютное значение функции в указанном диапазоне. В этой функции используется функция `linspace` для генерации значений в указанном диапазоне, после чего вычисляется максимальное абсолютное значение функции с помощью функции `max`.

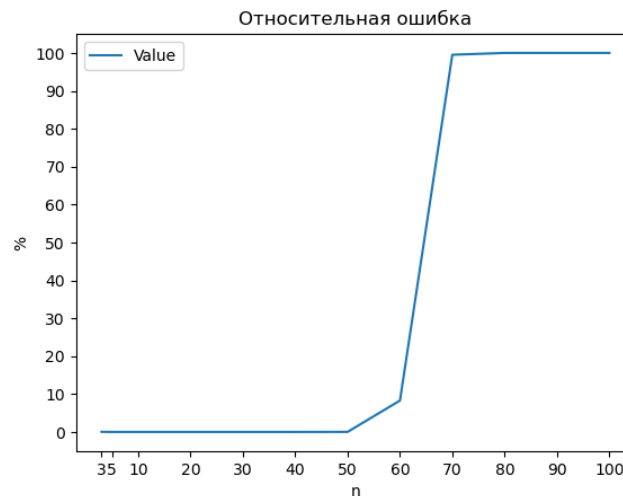
Относительная ошибка

```
1 def relative_error(abs_error: float, func_norm: float) -> float:
2
3     return (abs_error / func_norm) * 100
```

- Функция `relative_error` принимает абсолютную ошибку `abs_error` и норму функции `func_norm` в качестве входных данных. Она возвращает относительную ошибку, которая вычисляется как отношение абсолютной ошибки к норме функции, умноженное на 100.

Графическое представление

```
1 rel_df.plot(xticks=value_range, yticks=range(0, 100 + 1, 10),
2 xlabel="n", ylabel="%")
```



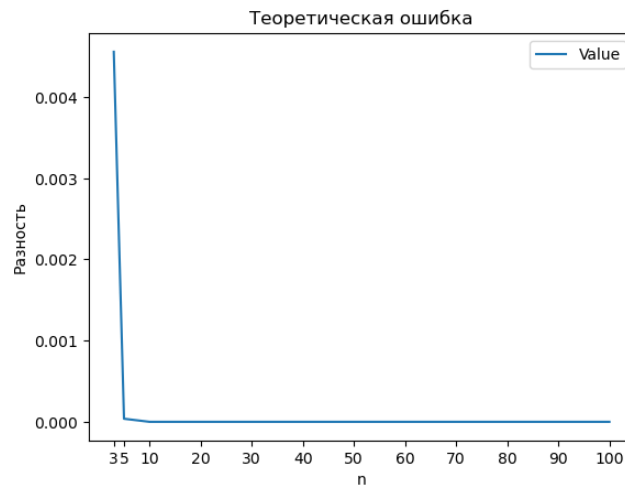
Теоретическая ошибка

```
1 def theoretical_error(count_points: int, range_tuple: tuple[float, float])
2 -> float:
3
4     return (calculate_max_absolute_value(derivative_function, range_tuple,
5 count_points + 1) / factorial(count_points + 1)) * (
6         (range_tuple[1] - range_tuple[0]) ** (count_points + 1))
```

- Функция `theoretical_error` принимает количество точек `count_points` и диапазон значений `range_tuple` в виде кортежа. Она возвращает теоретическую ошибку для указанного количества точек. Внутри функции используется функция `calculate_max_absolute_value` для вычисления максимального абсолютного значения производной функции `derivative_function` и факториал для указанного количества точек, чтобы получить теоретическую ошибку.

Графическое представление

```
1 abs_df.plot(xticks=value_range, xlabel="n")
```



6 Основной цикл

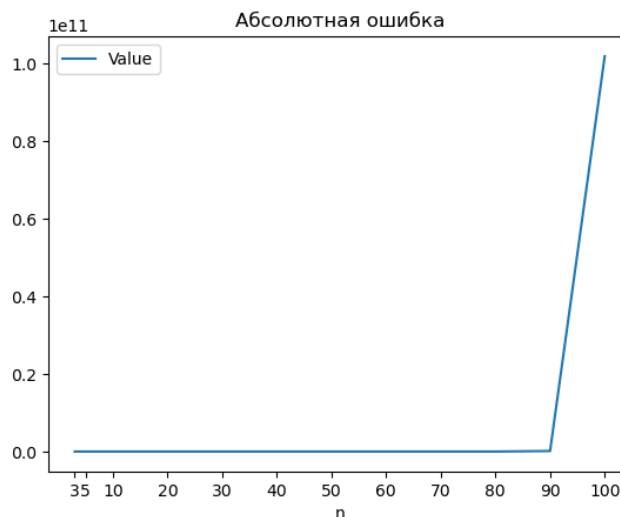
```
1 absolute_error_arr = []
2 relative_error_arr = []
3 theoretical_error_arr = []
4
5 for num_points in value_range:
6
7     current_points = generate_points(graph_range, num_points, function)
8     norm = calculate_max_absolute_value(function, graph_range)
9
10    lagrange_norm = calculate_max_absolute_value(lagrange_interpolation,
11    graph_range, current_points)
12
13    absolute_error = max(abs(lagrange_interpolation(linspace(*graph_range,
14    num=10 ** 3), current_points) - function(
15        linspace(*graph_range, num=10 ** 3))))
16
17    der_e = calculate_max_absolute_value(derivative_function, graph_range)
18    rel_e = relative_error(absolute_error, lagrange_norm)
19    ter_e = theoretical_error(num_points, graph_range)
20
21    absolute_error_arr.append(absolute_error)
22    relative_error_arr.append(rel_e)
23    theoretical_error_arr.append(ter_e)
24
25 abs_df = pd.DataFrame(absolute_error_arr, value_range, columns=["Value"])
26
27 rel_df = pd.DataFrame(relative_error_arr, value_range, columns=["Value"])
28
29 ter_df = pd.DataFrame(theoretical_error_arr, value_range, columns=["Value"])
30
31 total_df = pd.DataFrame({"Absolute error": absolute_error_arr,
32     "Relative error": relative_error_arr,
33     "Theoretical error": theoretical_error_arr},
34     index=value_range)
35 total_df.to_csv("ansv.csv")
```

- Для каждого значения `num_points` из набора значений `value_range`:
 - Генерация точек `current_points` с помощью функции `generate_points` для указанного диапазона `graph_range` и количества точек `num_points` с использованием функции `function`.
 - Вычисление нормы `norm` с помощью функции `calculate_max_absolute_value` для функции `function` и заданного диапазона `graph_range`.
 - Вычисление нормы лагранжевой интерполяции `lagrange_norm` с помощью функции `calculate_max_absolute_value` для лагранжевой интерполяции `lagrange_interpolation`, указанного диапазона `graph_range` и точек `current_points`.
 - Вычисление абсолютной ошибки `absolute_error` как максимального значения модуля разницы между значением лагранжевой интерполяции и значением функции на тысяче равномерно распределенных точках в заданном диапазоне `graph_range`.
 - Вычисление нормы производной `der_e` с использованием функции `calculate_max_absolute_value` для производной функции `derivative_function` в указанном диапазоне `graph_range`.
 - Вычисление относительной ошибки `rel_e` с использованием функции `relative_error` на основе абсолютной ошибки и нормы лагранжевой интерполяции.
 - Вычисление теоретической ошибки `ter_e` с использованием функции `theoretical_error` для указанного количества точек `num_points` и диапазона `graph_range`.
 - Добавление вычисленных значений абсолютной ошибки, относительной ошибки и теоретической ошибки в соответствующие списки `absolute_error_arr`, `relative_error_arr` и `theoretical_error_arr` соответственно.
- Создание трех `DataFrame` с использованием библиотеки `pandas` для абсолютной ошибки, относительной ошибки и теоретической ошибки под названиями `abs_df`, `rel_df` и `ter_df`, соответственно. Эти `DataFrame` содержат вычисленные значения, а индексы установлены равными значениям из `value_range`.
- Построение графиков для каждой ошибки с использованием функции `plot` из библиотеки `pandas`, где каждый график имеет заголовок, оси и отметки на осях, основанные на соответствующих значениях из `value_range`.

Абсолютная ошибка

```
1 ter_df.plot(xticks=value_range, xlabel="n")
```

Графическое представление



7 Таблица

- В результате работы цикла создается `csv` файл, содержащий в каждой строке количество точек, а также значения ошибок при этом количестве точек.

n	Абсолютная ошибка	Относительная ошибка	Теоретическая ошибка
3	0.0004041198030879656	0.030056475702117818	0.004557291666666667
5	$1.5256277570152577 \times 10^{-6}$	0.00011346881112684361	$3.7977430555555556 \times 10^{-5}$
10	$5.257128066205041 \times 10^{-12}$	$3.9099975001823635 \times 10^{-10}$	$2.1406830895763187 \times 10^{-11}$
20	$6.290523657526137 \times 10^{-13}$	$4.678587142260786 \times 10^{-11}$	$1.6332934790103295 \times 10^{-26}$
30	$3.5273739484864564 \times 10^{-10}$	$2.6234900780620463 \times 10^{-8}$	$9.91029116659731 \times 10^{-44}$
40	$4.1140015138996233 \times 10^{-7}$	$3.0597952784334175 \times 10^{-5}$	$2.3789170643336584 \times 10^{-62}$
50	0.00022682959886166643	0.016870488094400032	$5.010294122165272 \times 10^{-82}$
60	0.11674814466636829	8.28336936102891	$1.4952149433536201 \times 10^{-102}$
70	140.79557454487764	99.51802119595312	$8.71455012283797 \times 10^{-124}$
80	77993.4186208717	100.00088295864624	$1.248520445119785 \times 10^{-145}$
90	139505887.07198614	99.99999950636955	$5.227948020826171 \times 10^{-168}$
100	101879015502.40543	99.9999999932184	$7.322905978848986 \times 10^{-191}$

8 Вывод

- В результате проделанной работы была реализована функция полинома Лагранжа, а в заключении необходимо проанализировать поведение графиков ошибок:
 - Можно сделать вывод о том, что до 11 точек абсолютная ошибка стремилась к 0, но после она начала стремиться к $+\infty$.
 - Так же исходя из представленного ранее графика можно понять, что теоретическая ошибка стремится к 0.
 - Обращая внимание на график относительной ошибки, можно прийти к выводу о том, что она стремится к 100%.

