МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ
Кафедра вычислительной математики

Отчёт

Лабораторная работа

“Интерполирование функций. Интерполяционный многочлен в форме Лагранжа и Ньютона”

Вариант 5

Благодарного Артёма Андреевича
студента 3 курса, 3 группы
специальности «Информатика»
дисциплина «Численные методы»
Преподаватель: Будник А.М

Минск, 2025

Содержание:

Постановка задачи

Рассмотрим набор различных точек на отрезке [a, b] от x_0 до x_n по возрастанию значений. В этих точках заданы f(x_i). Требуется восстановить значение f(x) в других точках на [a, b].

- отрезок [a,b] = [α_j , 1+α_j], где α_j = 0.1 + 0.05*j,
  где j - номер в подгруппе (в моем случае 5)
- x_i = α_j + i * h, h = 1/n, i = 0,…,n, n = 10
- f(x) = α_j * e^x + (1 - α_j) * sin(x)
- x* = x_0 + ⅔ * h
- x** = $x_{n/2}$ + ½ * h
- x*** = x_n - ⅓ * h

```
Точки x_i и f(x_i):
x_0 = 0.35000, f(x_0) = 0.71956
x_1 = 0.45000, f(x_1) = 0.83164
x_2 = 0.55000, f(x_2) = 0.94639
x_3 = 0.65000, f(x_3) = 1.06381
x_4 = 0.75000, f(x_4) = 1.18402
x_5 = 0.85000, f(x_5) = 1.30721
x_6 = 0.95000, f(x_6) = 1.43372
x_7 = 1.05000, f(x_7) = 1.56400
x_8 = 1.15000, f(x_8) = 1.69866
x_9 = 1.25000, f(x_9) = 1.83846
x_10 = 1.35000, f(x_10) = 1.98432

Специальные точки:
x* = 0.41667, f(x*) = 0.79398
x** = 0.90000, f(x**) = 1.37002
x*** = 1.31667, f(x***) = 1.93496
```

В моем случае:

- α_j = 0.35
- отрезок [a, b] = [0,35; 1,35], длина отрезка равна 1
- значения x_i и f(x_i) находятся выше

Алгоритм решения

1) При решении задачи будем использовать многочлен Лагранжа в виде:

$$P_n(x) = \sum_{i=0}^{n} y_i \Lambda_i(x) = \sum_{i=0}^{n} y_i \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}.$$

2) Для нахождения приближенного значения в точках x*, x**, x*** просто подставим эти точки в функцию $P_n(x)$
3) Для нахождения оценки величины M будем использовать формулу:

$$M = \max |f^{n+1}(x)|,$$

где $x \in [a, b]$. То есть нужно будет найти n+1 производную функции f(x) и найти ее максимум на отрезке [a, b]

4) Для того, чтобы рассчитать истинную погрешность, будем использовать формулу:

$$r_n(x^*) = f(x^*) - P_n(x^*)$$

то же самое с точками x**, x***

5) Для оценки погрешности используем формулу:

$$|r_n(x)| \leq \frac{M}{(n+1)!} * |w_{n+1}(x)|,$$

где $w_{n+1}(x) = (x - x_0) * (x - x_1) *\ldots* (x - x_n)$

Листинг кода

```
import sympy as sp
import numpy as np
import pandas as pd
import math

# Параметры задачи
j = 5
alpha_j = 0.1 + 0.05 * j
n = 10
h = 1 / n

# Определение функции
def f(x):
    return alpha_j * np.exp(x) + (1 - alpha_j) * np.cos(x)

# Значения функции в узлах интерполирования
x_vals = np.array([alpha_j + i * h for i in range(n + 1)])
f_vals = f(x_vals)

# Таблица значений функции
table = pd.DataFrame({"x_i": x_vals, "f(x_i)": f_vals})
table_transposed = table.T

# Точки для проверки интерполяции
x_star = x_vals[0] + (2 / 3) * h
x_star2 = x_vals[len(x_vals) // 2] + (1 / 2) * h
x_star3 = x_vals[-1] - (1 / 3) * h

f_x_star = f(x_star)
f_x_star2 = f(x_star2)
f_x_star3 = f(x_star3)

# Интерполяционный многочлен Лагранжа
def lagrange_interpolation(x_vals, y_vals, x):
    n = len(x_vals)
    result = 0.0
    for i in range(n):
        term = y_vals[i]
        for j in range(n):
            if i != j:
                term *= (x - x_vals[j]) / (x_vals[i] - x_vals[j])
        result += term
    return result

P_x_star = lagrange_interpolation(x_vals, f_vals, x_star)
P_x_star2 = lagrange_interpolation(x_vals, f_vals, x_star2)
P_x_star3 = lagrange_interpolation(x_vals, f_vals, x_star3)

# Результаты интерполяции
data = {
    "Точка": ["x*", "x**", "x***"],
```

```
    "Значение x": [x_star, x_star2, x_star3],
    "f(x)": [f_x_star, f_x_star2, f_x_star3],
    "P(x) (полином)": [P_x_star, P_x_star2, P_x_star3]
}

df = pd.DataFrame(data)

# Производная (n+1)-го порядка
x = sp.Symbol('x')
f_sym = alpha_j * sp.exp(x) + (1 - alpha_j) * sp.sin(x)
f_derivative = sp.diff(f_sym, x, n + 1)

# Максимум абсолютного значения производной на отрезке [0.4, 1.4]
f_derivative_abs = sp.lambdify(x, sp.Abs(f_derivative), 'numpy')
x_test = np.linspace(0.4, 1.4, 1000)
M_max = np.max(f_derivative_abs(x_test))

# Истинная погрешность
r_x_star = f_x_star - P_x_star
r_x_star2 = f_x_star2 - P_x_star2
r_x_star3 = f_x_star3 - P_x_star3

# Оценка погрешности по неравенству
factorial = math.factorial(n + 1)
x_stars = [x_star, x_star2, x_star3]
r_x_stars = [r_x_star, r_x_star2, r_x_star3]
error_bound_stars = []

for x_val in x_stars:
    prod_term = np.prod([abs(x_val - xi) for xi in x_vals])
    error_bound = M_max / factorial * prod_term
    error_bound_stars.append(error_bound)

# Проверка выполнения неравенства
is_error_bound_stars_valid = [
    abs(r_x_stars[i]) <= error_bound_stars[i] for i in range(3)
]

# Таблица ошибок
error_table = pd.DataFrame({
    "Точка": ["x*", "x**", "x***"],
    "Значение x": x_stars,
    "Левая часть |r_n(x)|": [abs(r) for r in r_x_stars],
    "Правая часть оценки погрешности": error_bound_stars,
    "M = max|f^(n+1)(x)|": [M_max] * 3,
    "Неравенство выполняется?": is_error_bound_stars_valid
})

# Вывод таблиц
display(table_transposed)
display(df)
display(error_table)
```

Результаты

- Исходная таблица:

	0	1	2	3	4	5	6	7	8	9	10
x_i	0.350000	0.4500	0.550000	0.650000	0.750000	0.850000	0.950000	1.050000	1.150000	1.25000	1.350000
f(x_i)	1.107266	1.1342	1.160779	1.187894	1.216548	1.247865	1.283092	1.323599	1.370884	1.42658	1.492453

- Значение функции и полинома Лагранжа в точках x*, x**, x***;

	Точка	Значение x	f(x)	P(x) (полином)
0	x*	0.416667	1.125302	1.125302
1	x**	0.900000	1.264908	1.264908
2	x***	1.316667	1.469249	1.469249

- Проверка неравенства для истинной погрешности

	Точка	Значение x	r истинная	оценка погрешности	M = max|f^(n+1)(x)|	Неравенство выполняется?
0	x*	0.416667	6.239453e-14	9.928288e-14	2.059862	True
1	x**	0.900000	1.998401e-15	2.475192e-15	2.059862	True
2	x***	1.316667	1.374456e-13	2.116106e-13	2.059862	True

Анализ

Высокая точность интерполяции:
Истинная погрешность $r_n(x)$ во всех приближенных точках очень мала (порядка 10^{-13}, 10^{-14}), что свидетельствует о высокой точности интерполяции.
Оценка погрешности подтверждается:
Во всех случаях левая часть $|r_n(x)|$ (истинная погрешность) меньше правой части (теоретическая оценка погрешности) из-за M – max.
Разные точки — разный порядок истинной погрешности:

В точке x* погрешность ≈ $6.24*10^{-14}$, но всё же меньше оценочной границы.

В точке x** погрешность минимальна (≈ $1.99*10^{-15}$), что может быть связано с попаданием точки ближе к узлу интерполяции. В точке x*** погрешность снова возрастает (≈ 1.37×10^{-13}), но остаётся в пределах оценки.

Метод **интерполяции Лагранжа** работает точно для данной функции и набора узлов. Оценка погрешности выполняется, а истинные погрешности остаются значительно ниже предсказанных границ, что подтверждает надежность метода в данном случае.