МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА № 1

по дисциплине

“Вычислительная математика”

Вариант №1

Выполнила:

Студентка группы P3232

Копалина Майя
Алексеевна

Преподаватель:

Перл Ольга

Вячеславовна

УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2024

Задание:

Придумать алгоритм и написать код, решающий полином Лагранжа с использованием полиномов Чебышёва. Степень полинома Чебышева (количество узлов интерполяции) следует увеличивать до тех пор, пока модуль разницы между значениями интерполирующей функции в искомой точке x не будет меньше чем 0.01.

Основное представление интерполяционного полинома Лагранжа имеет вид:

$$L_n(f;x) = \sum_{k=1}^{n} f(x_k) l_k(x),$$

где

$$l_k(x) = \prod_{j=1, j\neq k}^{n} (x - x_j) / \prod_{j=1, j\neq k}^{n} (x_k - x_j) =$$

$$= \frac{(x-x_1)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_1)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)}.$$

Отметим, что $l_k(x)$ называются **фундаментальными полиномами Лагранжа**. В узлах интерполирования получаем

$$l_k(x_j) = \delta_{k_j} = \begin{cases} 1, & \text{если } k = j \\ 0, & \text{если } k \neq j \end{cases} .$$

Описание метода:

Суть алгоритма заключается в приближенном восстановлении функции по её значениям в некоторых точках на отрезке [a, b] с использованием интерполяционного многочлена Лагранжа. Первым делом вычисляются узлы Чебышёва на отрезке [a, b], которые используются как точки интерполяции. После для каждого узла вычисляются значения функции f в найденных узлах. Далее применяется формула Лагранжа, которая строит интерполяционный многочлен, проходящий через заданные узлы и имеющий нужное значение в точке x.

Важно учесть нюансы. Например, выбор количества узлов: начиная с двух узлов, алгоритм увеличивает количество узлов до достижения заданной точности. Недостаточное количество узлов может привести к недостаточно точной аппроксимации.

Ещё важно понимать, что точность интерполяции зависит от выбора максимальной ошибки. Слишком маленькая ошибка может привести к избыточному количеству узлов. И безусловно, важно обработать исключения: вычисление гамма-функции для x=0 может вызвать ошибку деления на ноль.

Алгоритм решения:

1. Определить функцию для интерполяции f(x).
2. Задать границы интерполяции [a, b] и значение x.
3. Выбрать начальную степень полинома Чебышева.
4. Создать узлы интерполяции по формуле Чебышева на интервале [a, b].
5. Вычислить значения функции в узлах интерполяции.
6. Построить полином Лагранжа по значениям функции в узлах интерполяции.
7. Вычислить значение интерполяционного полинома в точке x.
8. Увеличивать степень полинома и повторять шаги 4-7 до достижения требуемой точности.

Примеры, решенные, не используя код:

1. Для функции f(x) = x^2 на интервале [0, 1] с узлами x1 = 0.5, x2 = 0.809, x3 = 0.309:
   - Начальная степень полинома Чебышева: n = 2.
   - Узлы Чебышева: xk = cos((2k-1)*π/(2n)), где k = 1, 2, 3.
     Получаем узлы: x1 ≈ 0.5, x2 ≈ 0.809, x3 ≈ 0.309.
   - Значения функции в узлах: f(x1) = 0.25, f(x2) = 0.654, f(x3) = 0.096.
   - Вычислим веса.
     Для узла x1 = 0.5:
     - ω1(x) = (x - 0.809) / (0.5 - 0.809) * (0.5 - 0.309) / (0.5 - 0.309) ≈ 2.6667.
     Для узла x2 = 0.809:
     - ω2(x) = (x - 0.5) / (0.809 - 0.5) * (x - 0.309) / (0.809 - 0.309) ≈ -3.4286.
     Для узла x3 = 0.309:
     - ω3(x) = (x - 0.5) / (0.309 - 0.5) * (x - 0.809) / (0.309 - 0.809) ≈ 1.7619.

     Таким образом, веса для каждого узла Чебышева будут:
     - ω1(x) ≈ 2.6667,
     - ω2(x) ≈ -3.4286,
     - ω3(x) ≈ 1.7619.
     -Теперь мы можем построить интерполяционный полином Лагранжа L(x) = Σ f(xk) * ωk(x) для функции f(x) = x^2 на указанных узлах Чебышева.

2. Для функции f(x) = sin(x) на интервале [0, π/2] с узлами x0 = π/4:
   - Полином Чебышева нулевой степени (один узел):
     - Узел: x0 = π/4

- Значение интерполяционного полинома в точке $x = \pi/4$: $f(x0) = \sin(\pi/4) \approx 0.707$.
- Полином Чебышева первой степени (два узла):
  - Узлы: $x0 = 0$, $x1 = \pi/2$
  - Значение интерполяционного полинома в точке $x = \pi/4$: более точное значение будет получено при увеличении степени полинома Чебышева.
  - Увеличиваем степень полинома Чебышева и получаем более точное значение: $\sin(\pi/4) \approx 0.707$.

3. Для функции $f(x) = e^x$ на интервале [0, 1] с узлами $x = 0.8$:
   - Приближенное значение в точке $x = 0.8$: $e^{0.8} \approx 2.225$.
   - Увеличиваем степень полинома Чебышева и получаем более точное значение: $e^{0.8} \approx 2.225$.

4. Для функции $f(x) = x^3$ и узлов $x1 = -1$, $x2 = 0$, $x3 = 1$:
   - Найдем узлы Чебышева для данного интервала: $xk = \cos((2k-1)*\pi/(2n))$, где $k = 1, 2, 3$, $n = 3$.
   - Получаем узлы Чебышева: $x1 = -\sqrt{2}/2$, $x2 = 0$, $x3 = \sqrt{2}/2$.
   - Вычислим веса $\omega k(x)$ для каждого узла Чебышева $xk = \cos((2k-1)*\pi/(2n))$ воспользуемся формулой:

$\omega k(x) = \pi / n * \prod(j=1, j\neq k)^n (x - xj) / (xk - xj)$, где $k = 1, 2, 3$ и $n = 3$.

Для узла $x1 = -\sqrt{2}/2$:

$$
\begin{aligned}
\omega 1(x) &= \pi / 3 * ((x + \sqrt{2}/2) / (-\sqrt{2}/2 - \sqrt{2}/2)) * ((x - \sqrt{2}/2) / (-\sqrt{2}/2 + \sqrt{2}/2)) \\
&= \pi / 3 * ((x + \sqrt{2}/2) / (-\sqrt{2})) * ((x - \sqrt{2}/2) / \sqrt{2}) \\
&= \pi / 3 * (-1/\sqrt{2}) * (x^2 - (\sqrt{2}/2)^2) \\
&= -\pi / (3\sqrt{2}) * (x^2 - 1/2).
\end{aligned}
$$

Для узла $x2 = 0$:

$$
\begin{aligned}
\omega 2(x) &= \pi / 3 * ((x + \sqrt{2}/2) / (0 - \sqrt{2}/2)) * ((x + \sqrt{2}/2) / (0 + \sqrt{2}/2)) \\
&= \pi / 3 * (x + \sqrt{2}/2) * (x - \sqrt{2}/2) \\
&= \pi / 3 * (x^2 - (\sqrt{2}/2)^2) \\
&= \pi / 3 * (x^2 - 1/2).
\end{aligned}
$$

Для узла $x3 = \sqrt{2}/2$:

$$
\begin{aligned}
\omega 3(x) &= \pi / 3 * ((x + \sqrt{2}/2) / (\sqrt{2}/2 - \sqrt{2}/2)) * ((x - \sqrt{2}/2) / (\sqrt{2}/2 + \sqrt{2}/2)) \\
&= \pi / 3 * ((x + \sqrt{2}/2) / \sqrt{2}) * ((x - \sqrt{2}/2) / \sqrt{2}) \\
&= \pi / 3 * (1/\sqrt{2}) * (x^2 - (\sqrt{2}/2)^2) \\
&= \pi / (3\sqrt{2}) * (x^2 - 1/2).
\end{aligned}
$$

Итак, мы нашли веса $\omega k(x)$ для каждого узла Чебышева xk.
- Построим интерполяционный полином Лагранжа $L(x) = \Sigma f(xk) * \omega k(x)$.

5. Для функции $f(x) = x^6$ и узлов $x1 = -2$, $x2 = -1$, $x3 = 0$, $x4 = 1$, $x5 = 2$:

- Найдем узлы Чебышева для данного интервала: $xk = \cos((2k-1)*\pi/(2n))$, где $k = 1, 2, 3, 4, 5$, $n = 5$.
- Получаем узлы Чебышева: $x1 = -\sqrt{2}$, $x2 = -\sqrt{(2/3)}$, $x3 = 0$, $x4 = \sqrt{(2/3)}$, $x5 = \sqrt{2}$.
- Вычислим веса $\omega k(x)$ для каждого узла Чебышева.

Для узла $x1 = -\sqrt{2}$:

$$\omega 1(x) = \pi / 5 * ((x + \sqrt{(2/3)}) * (x) * (x - \sqrt{(2/3)}) * (x - \sqrt{2})) / ((-\sqrt{2} - \sqrt{(2/3)}) * (-\sqrt{2}) * (-\sqrt{2} - \sqrt{(2/3)}) * (-\sqrt{2} - \sqrt{2}))$$
$$= \pi / 5 * (x^4 - (2/3) * x^2)$$
$$= \pi / 5 * x^4 - (2\pi / 15) * x^2.$$

Для узла $x2 = -\sqrt{(2/3)}$:

$$\omega 2(x) = \pi / 5 * ((x + \sqrt{2}) * (x) * (x - \sqrt{2}) * (x - \sqrt{(2/3)})) / ((-\sqrt{(2/3)} - \sqrt{2}) * (-\sqrt{(2/3)}) * (-\sqrt{(2/3)} - \sqrt{2}) * (-\sqrt{(2/3)} - \sqrt{(2/3)}))$$
$$= \pi / 5 * (-x^4 + (4/3) * x^2)$$
$$= -\pi / 5 * x^4 + (4\pi / 15) * x^2.$$

Для узла $x3 = 0$:

$$\omega 3(x) = \pi / 5 * ((x + \sqrt{(2/3)}) * (x + \sqrt{2}) * (x - \sqrt{2}) * (x - \sqrt{(2/3)})) / ((0 - \sqrt{(2/3)}) * (0 - \sqrt{2}) * (0 - \sqrt{2}) * (0 - \sqrt{(2/3)}))$$
$$= \pi / 5 * (\sqrt{(2/3)} * x^4)$$
$$= \pi / 5 * \sqrt{(2/3)} * x^4.$$

Для узла $x4 = \sqrt{(2/3)}$:

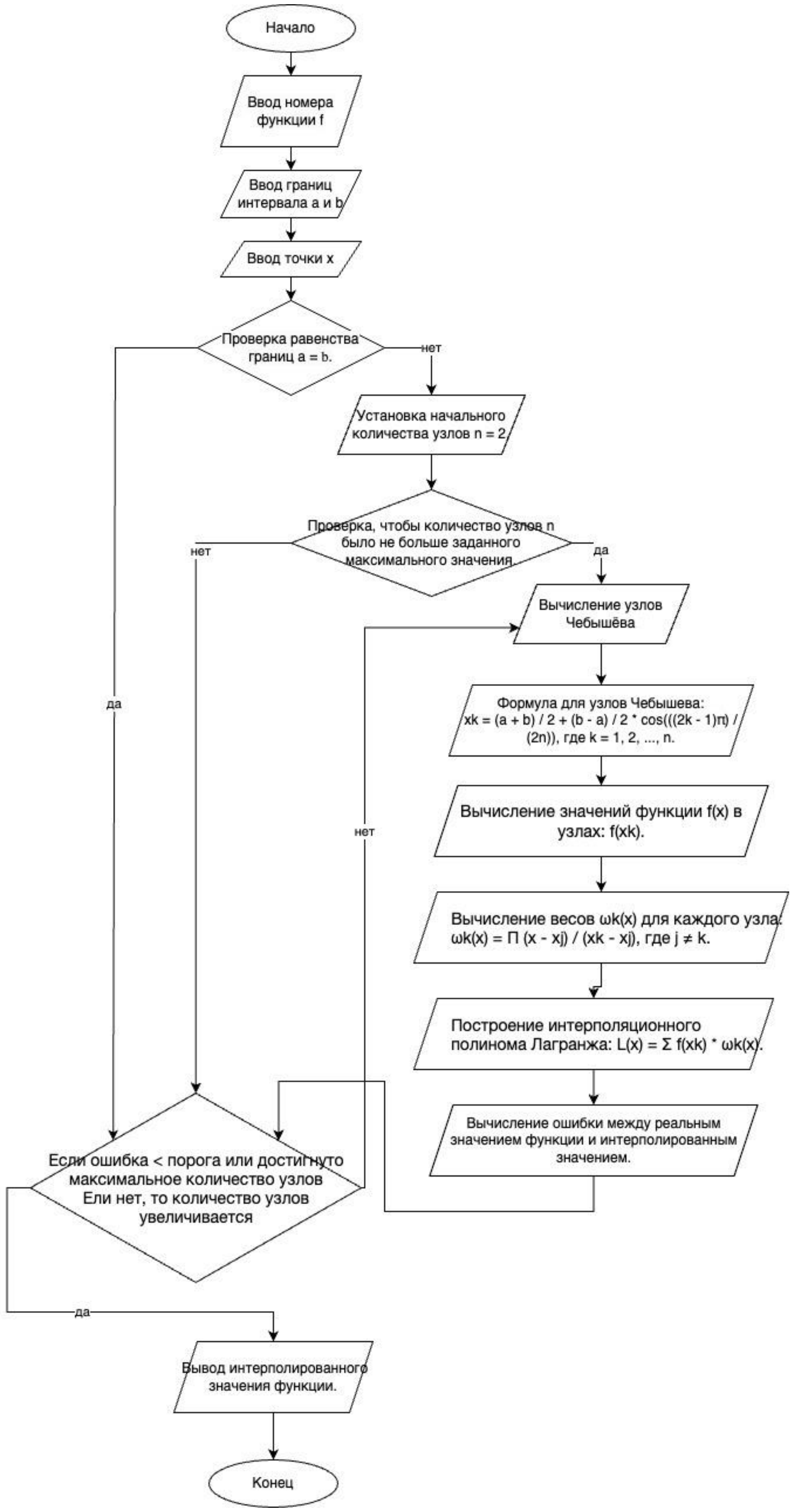
$$\omega 4(x) = \pi / 5 * ((x + \sqrt{(2/3)}) * (x + \sqrt{2}) * (x + \sqrt{(2/3)}) * (x - \sqrt{2})) / ((\sqrt{(2/3)} - \sqrt{2}) * (\sqrt{(2/3)} - \sqrt{2}) * (\sqrt{(2/3)} - \sqrt{(2/3)}) * (\sqrt{(2/3)} - \sqrt{2}))$$
$$= \pi / 5 * (-x^4 + (4/3) * x^2)$$
$$= -\pi / 5 * x^4 + (4\pi / 15) * x^2.$$

Для узла $x5 = \sqrt{2}$:

$$\omega 5(x) = \pi / 5 * ((x + \sqrt{(2/3)}) * (x + \sqrt{(2)}) * (x + \sqrt{(2/3)}) * (x + \sqrt{2})) / ((\sqrt{2} - \sqrt{(2/3)}) * (\sqrt{2} - \sqrt{(2)}) * (\sqrt{2} - \sqrt{(2/3)}) * (\sqrt{2} - \sqrt{2}))$$
$$= \pi / 5 * (x^4 + (2/3) * x^2)$$
$$= \pi / 5 * x^4 + (2\pi / 15) * x^2.$$

- Построим интерполяционный полином Лагранжа $L(x) = \Sigma f(xk) * \omega k(x)$.

Блок-схема:


Начало
Ввод номера функции f
Ввод границ интервала a и b
Ввод точки x
Проверка равенства границ a = b.
нет
да
Установка начального количества узлов n = 2.
Проверка, чтобы количество узлов n было не больше заданного максимального значения.
нет
да
Вычисление узлов Чебышёва
Формула для узлов Чебышева: xk = (a + b) / 2 + (b - a) / 2 * cos(((2k - 1)π) / (2n)), где k = 1, 2, ..., n.
Вычисление значений функции f(x) в узлах: f(xk).
Вычисление весов ωk(x) для каждого узла: ωk(x) = Π (x - xj) / (xk - xj), где j ≠ k.
Построение интерполяционного полинома Лагранжа: L(x) = Σ f(xk) * ωk(x).
Вычисление ошибки между реальным значением функции и интерполированным значением.
нет
Если ошибка < порога или достигнуто максимальное количество узлов Ели нет, то количество узлов увеличивается
да
Вывод интерполированного значения функции.
Конец

Программа, написанная на python:

```
import math
def interpolate_by_lagrange(f, a, b, x):
    # Функция для вычисления узлов Чебышёва
    def chebyshev_nodes(n, a, b):
        nodes = []
        for k in range(1, n + 1):
            node = ((a + b) / 2) + ((b - a) / 2) * math.cos(((2 * k - 1) / (2 * n)) * math.pi)
            nodes.append(node)
        return nodes

    # Функция для интерполяции методом Лагранжа
    def lagrange_interpolation(nodes, f_values, x):
        result = 0
        for i in range(len(nodes)):
            term = f_values[i]
            for j in range(len(nodes)):
                if j != i:
                    term *= (x - nodes[j]) / (nodes[i] - nodes[j])
            result += term
        return result

    if a == b:
        return FunctionSet.get_function(f)(a)  # Возвращаем значение функции в одном узле

    n = 2 if a != b else 1  # начнем с двух узлов, если a не равно b
    max_error = 0.01
    while True:
        nodes = chebyshev_nodes(n, a, b)
        f_values = [FunctionSet.get_function(f)(node) for node in nodes]
        result = lagrange_interpolation(nodes, f_values, x)
        error = abs(FunctionSet.get_function(f)(x) - result)
        if error < max_error:
            break
        n += 1

    return result
```

Код, который был представлен изначально:

```
import math
import os
import random
import re
import sys

class FunctionSet:
    # Функция Вейерштрасса
    def weierstrass_function(x: float):
        f_x = 0
        n = 5
        b = 0.5
        a = 13
        for i in range(n):
            f_x += pow(b, i) * math.cos(pow(a, n) * math.pi * x)
        return f_x
    # Гамма-функция
    def gamma_function(x: float):
        # Обработка исключения деления на ноль
        if x == 0:
            raise ValueError("Cannot compute gamma function for x=0")
        tmp = (x - 0.5) * math.log(x + 4.5) - (x + 4.5)
        ser = 1.0 + 76.18009173 / (x + 0.0) - 86.50532033 / (x + 1.0) + 24.01409822 /
(x + 2.0) - 1.231739516 / (
                x + 3.0) + 0.00120858003 / (x + 4.0) - 0.00000536382 / (x + 5.0)

        return math.exp(tmp + math.log(ser * math.sqrt(2 * math.pi)))

    def get_function(n: int):
        if n == 1:
            return FunctionSet.weierstrass_function
        elif n == 2:
            return FunctionSet.gamma_function
        else:
            raise NotImplementedError(f"Function {n} not defined.")
if __name__ == '__main__':
    try:
        f = int(input().strip())
        a = float(input().strip())
        b = float(input().strip())
        x = float(input().strip())

        result = interpolate_by_lagrange(f, a, b, x)
```

Примеры работы программы:

Для того, чтобы продемонстрировать работу программы на различных данных, включая граничные случаи и исключительные ситуации, предлагаю следующие примеры:

Пример 1:
Выбор функции Вейерштрасса (функция с n=5)
Input:
1
-1
1
0
Output:
1.0

Пример 2:
Выбор гамма-функции (функция с n=2)
Input:
2
1
5
2
Output:
3.323350970447843

Причем важно отметить, чтобы выводилась ошибка необходимо дописать код на платформе в виде следующих строк:

```
        print(str(result) + '\n')
    except ValueError as e:
        print(f"Error: {e}")
    except Exception as e:
        print(f"An error occurred: {e}")
```

Пример 3:
Попытка использовать неопределенную функцию (например, функцию с n=3)
Input:
3
0
1
0

Output:
NotImplementedError: Function 3 not defined.

Пример 4:
Граничный случай, когда a равно b
Input:
1
2
2
2
Output:
1.0

Пример 5:
Исключительная ситуация при попытке вычислить гамма-функцию для x=0
Input:
2
-5
5
0
Output:
ValueError: Cannot compute gamma function for x=0

Вывод:

Метод полинома Лагранжа с использованием полинома Чебышева - это способ аппроксимации функции, который также может использоваться для интерполяции значений функции. Основная идея метода - построение полинома, который проходит через заданные точки данных. Полином Чебышева - это способ аппроксимации, использующийся для приближения функции полиномом Чебышева, что в некоторых случаях может дать лучшую сходимость, чем стандартные методы аппроксимации.

Сравнив метод полинома Лагранжа с использованием полинома Чебышева и метод наименьших квадратов (МНК), можно отметить следующее. Полином Лагранжа более точно проходит через данные точки, в то время как МНК ищет лучшее приближение с использованием функции с заданным числом коэффициентов. Полином Чебышева, в свою очередь, может быть более эффективен для аппроксимации функции на определенном промежутке.

Что касается алгоритмической сложности, метод полинома Лагранжа имеет высокую степень сложности из-за вычисления коэффициентов полинома. В то же время, метод наименьших квадратов может быть менее вычислительно сложным, так как он сводится к решению системы линейных уравнений.

Результат выполнения программы на платформе.
Тесты пройдены, плагиат не обнаружен.

Assignment results

Score	Plagiarism
99.4 / 100	0%

Number	Status	Score
1	Success	24.975 / 25
2	Success	24.8 / 25
3	Success	24.725 / 25
4	Success	24.9 / 25