Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО».

Факультет программной инженерии и компьютерной техники

Лабораторная работа №5

Вариант №12

Интерполяция функции

Выполнил

Путинцев Данил Денисович

Группа P3207

Проверил(а)

Преподаватель: Рыбаков Степан Дмитриевич

Санкт-Петербург 2025 год

## Цели лабораторной работы

Решить задачу интерполяции, найти значения функции при заданных значениях аргумента, отличных от узловых точек

## Порядок выполнения работы

**1. Выбрать таблицу y = f(x):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | x | y | № Варианта | X1 | X2 |
| Таблица 1.2 | 0.50 | 1.5320 | 12 | 0.523 | 0.639 |
| 0.55 | 2.5356 |
| 0.60 | 3.5406 |
| 0.65 | 4.5462 |
| 0.70 | 5.5504 |
| 0.75 | 6.5559 |
| 0.80 | 7.5594 |

**2. Построить таблицу конечных разностей для заданной таблицы:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | xi | yi | Δyi | Δ2yi | Δ3yi | Δ4yi | Δ5yi | Δ6yi |
| 0. | 0.50 | 1.5320 | 1.0036 | 0.0014 | -0.0008 | -0.0012 | 0.0059 | -0.0166 |
| 1. | 0.55 | 2.5356 | 1.0050 | 0.0006 | -0.0020 | 0.0047 | -0.0107 |  |
| 2. | 0.60 | 3.5406 | 1.0056 | -0.0014 | 0.0027 | -0.0060 |  |  |
| 3. | 0.65 | 4.5462 | 1.0042 | 0.0013 | -0.0033 |  |  |  |
| 4. | 0.70 | 5.5504 | 1.0055 | -0.0020 |  |  |  |  |
| 5. | 0.75 | 6.5559 | 1.0035 |  |  |  |  |  |
| 6. | 0.80 | 7.5594 |  |  |  |  |  |  |

**3. Вычислить значения функции для аргумента X1, используя первую или вторую интерполяционную формулу Нью**тона.

Воспользуемся первой интерполяционной формулой Ньютона для интерполирования вперед, так как X1 лежит в левой половине отрезка.

Для X1 = 0.523:

+

+

+

y(0.523) = 1.9940

**4. Вычислить значение функции для аргумента X2, используя первую или вторую интерполяционную формулу Гаусса.**

Центральная точка a = 0.65, X2 = 0.639 < 0.65, то есть x < a => используем вторую интерполяционную формулу Гаусса

+

+

+

y(0.639) = 4.3256

## Листинг программы

import numpy as np  
import math  
  
def create\_divided\_differences(table):  
 x = table[0]  
 y = table[1]  
 n = len(x)  
  
 diff\_table = [y.copy()]  
  
 for i in range(1, n):  
 level = []  
 for j in range(n - i):  
 delta = (diff\_table[i - 1][j + 1] - diff\_table[i - 1][j]) / (x[j + i] - x[j])  
 level.append(delta)  
 diff\_table.append(level)  
  
 return diff\_table  
  
def create\_difference\_table(table):  
 y\_values = table[1]  
 difference\_table = [y\_values.copy()]  
  
 current\_level = y\_values  
 while len(current\_level) > 1:  
 next\_level = []  
 for i in range(1, len(current\_level)):  
 next\_level.append(current\_level[i] - current\_level[i - 1])  
 difference\_table.append(next\_level)  
 current\_level = next\_level  
  
 return difference\_table  
  
def method\_langrange(x, table):  
 x\_values = table[0]  
 y\_values = table[1]  
 n = len(x\_values)  
 result = 0.0  
  
 for i in range(n):  
 term = y\_values[i]  
 for j in range(n):  
 if j != i:  
 term \*= (x - x\_values[j]) / (x\_values[i] - x\_values[j])  
 result += term  
  
 return result  
  
  
def method\_gauss(x, table):  
 x\_values = np.array(table[0])  
 y\_values = np.array(table[1])  
 n = len(x\_values)  
  
 # Проверка равноотстоящих узлов  
 h = x\_values[1] - x\_values[0]  
 if not np.allclose(np.diff(x\_values), h, atol=1e-6):  
 raise ValueError("Узлы не равноотстоящие")  
  
 mid\_idx = np.argmin(np.abs(x\_values - x))  
 mid\_idx = max(1, min(mid\_idx, n - 2))  
 a = x\_values[mid\_idx]  
  
 t = (x - a) / h  
  
 use\_first\_formula = (t < 0.5 and t >= -0.5)  
  
 diff\_table = create\_difference\_table(table)  
  
 result = y\_values[mid\_idx]  
 product = 1.0  
  
 for i in range(1, len(diff\_table)):  
 if use\_first\_formula:  
 idx = mid\_idx - (i // 2)  
 else:  
 idx = mid\_idx - (i // 2) + (i % 2)  
  
 if idx < 0 or idx >= len(diff\_table[i]):  
 break  
  
 delta = diff\_table[i][idx]  
  
 term = 1.0  
 for j in range(i):  
 if use\_first\_formula:  
 k = j - (i // 2)  
 else:  
 k = (i // 2 - 1) + j  
 term \*= (t - k)  
  
 result += delta \* term / math.factorial(i)  
  
 return result  
  
def method\_newton(x, table):  
 x\_nodes = table[0]  
 y\_nodes = table[1]  
 n = len(x\_nodes)  
  
 diff\_table = create\_divided\_differences(table)  
  
 result = diff\_table[0][0]  
 product = 1.0  
  
 for i in range(1, n):  
 product \*= (x - x\_nodes[i - 1])  
 result += diff\_table[i][0] \* product  
  
 return result

## Результаты выполнения программы

Вариант №12

Интерполяция функций

Добро пожаловать в программу, которая осуществляет интерполяцию функций

Список команд доступен по команде /help

1. /help - вывести список команд с их описанием

2. /exit - выход из программы

3. /info - вывести информацию о введеденных данных

4. /start - запуск программы

5. /clear - очистка введенных данных

6. /input\_table - ввод таблицы y = f(x) из консоли

7. /input\_table\_file - ввод таблицы y = f(x) из файла

8. /input\_interval - ввод интервала

9. /input\_count\_point - ввод количества точек на интервале

10. /choice\_equations - выбор функции

!("/exit" to quit) Введите команду: => 4

Отправка: 4

Выберете, каким образом будете вводить информацию

1. Введу таблицу

2. Введу таблицу из файла

3. Выберу функцию и введу параментры

Ваш выбор: 3

0. 2x^3 + 3.41x^2 - 1.943x + 2.12 = 0

1. sin(x) + cos(x) - 0.6 = 0

2. cos(x) - 0.34x - 0.21 = 0

3. -3.2x^3 - 3.2x - 2 = 0

4. -33x^3 + 21.23x^2 + 0.68 = 0

Введите номер функции: 2

Введите нижнюю границу интервала: -2

Введите верхнюю границу интервала: 2

Введите количество точек на интервале: 25

Выберете метод для интерполяции функции

1. Многочлен Лагранжа

2. Многочлен Ньютона с разделенными разностями

3. Многочлен Гаусса

4. Все методы

Выберете метод для интерполяции функции: 4

Введите значения для нахождения приближенного значения функции: 1.233

Метод Лагранжа: -0.29781123910873786

Метод Ньютона с разделенными разностями: -0.2978112391087393

Метод Гаусса: -0.3021653864189758

## Выводы

В рамках данной лабораторной работы были изучены и применены методы интерполяции Ньютона и Гаусса для анализа табличных данных. Эти методы позволяют находить значения функции в точках, не указанных в исходной таблице, что особенно полезно для прогнозирования и анализа данных.

Программа, разработанная в ходе работы, успешно вычислила приближенные значения функции для заданных аргументов, используя оба метода. Сравнение результатов показало, что и интерполяция Ньютона, и интерполяция Гаусса дают близкие результаты, однако их точность может варьироваться в зависимости от характера функции и расположения узловых точек.

Проведенная работа демонстрирует, что выбор метода интерполяции должен основываться на особенностях решаемой задачи, таких как требуемая точность и структура исходных данных. Оба метода являются мощными инструментами в численном анализе и могут быть эффективно использованы в практических приложениях.