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

Факультет программной инженерии и компьютерной техники
Направление подготовки: 09.03.04 — Системное и прикладное программное обеспечение
Дисциплина «Вычислительная математика»

Лабораторная работа №5

Вариант 7

Выполнил:

Капарулин Тимофей Иванович

Преподаватель:

Машина Екатерина Алексеевна

г.Санкт-Петербург 2025 г.

Цель работы

Решить задачу интерполяции, найти значения функ ции при заданных значениях аргумента, отличных от узловых точек.

Вычислительная реализация

1) **Таблица исходных данных**

x	y
0.50	*1.5320*
0.55	*2.5356*
0.60	*3.5406*
0.65	*4.5462*
0.70	*5.5504*
0.75	*6.5559*
0.80	*7.5594*

$X_1 = 0.751$, $X_2 = 0.651$

2) **таблица конечных разностей**

x	*y*	*Δy*	*Δ²y*	*Δ³y*	*Δ⁴y*	*Δ⁵y*	*Δ⁶y*
0.50	*1.5320*	*1.0036*	*0.0014*	*-0.0008*	*-0.0012*	*0.0059*	*-0.0166*
0.55	*2.5356*	*1.0050*	*0.0006*	*-0.0020*	*0.0047*	*-0.0107*	
0.60	*3.5406*	*1.0056*	*-0.0014*	*0.0027*	*-0.0060*		
0.65	*4.5462*	*1.0042*	*0.0013*	*-0.0033*			
0.70	*5.5504*	*1.0055*	*-0.0020*				
0.75	*6.5559*	*1.0035*					
0.80	*7.5594*						

3) **вычисление X_1**

Так как $X_1 \in [x_5, x_6]$ и она находится в левой части, то целесообразней использовать формулу Ньютона для интеполлирования вперед. Тогда узлы: 0.75, 0.8

$$t = \frac{X_1 - x_5}{0.05} = \frac{0.751 - 0.75}{0.05} = 0.02$$

$$N_1(0.751) = y_5 + t \cdot \Delta y_5 = 6.5559 + 0.02 \cdot 1.0035 = 6{,}576$$

4) **вычисление X_2**

Так как $X_2 \in [x_3, x_4]$ и она находится в левой части, то воспользуемся первой интерполяционной формулой Гаусса.

$$t = \frac{X_2 - x_3}{0.05} = \frac{0.651 - 0.65}{0.05} = 0.02$$

$$P_2(0.651) =$$
$$y_3 \;+\; t\cdot\Delta y_3 \;+\; t\frac{t-1}{2!}\Delta^2 y_2 \;+\; (t+1)t\frac{t-1}{3!}\Delta^3 y_2$$
$$= 4.5462 + 0.02\cdot 1.0042 + 0.02\cdot(-0.98)\cdot\frac{0.0013}{2} + 0.02\cdot(-0.98)\cdot 1.02\cdot\frac{0.0027}{6}$$
$$= 4.566$$

Листинг программы

```
# Построение таблицы конечных разностей
def finite_diff(y):
    n = len(y)
    table = np.zeros((n, n))
    table[:,0] = y
    for j in range(1, n):
        for i in range(n - j):
            table[i][j] = table[i+1][j-1] - table[i][j-1]
    return table

# 1. Интерполяция многочленом Лагранжа
def lagrange_interpolation(x, y, x_val):
    result = 0.0
    n = len(x)
    for i in range(n):
        term = y[i]
        for j in range(n):
            if j != i:
                term *= (x_val - x[j]) / (x[i] - x[j])
        result += term
    return result

# 2. Интерполяция Ньютона с разделенными разностями
def newton_divided_diff(x, y, x_val):
    n = len(x)
    coef = np.zeros([n, n])
    coef[:,0] = y

    for j in range(1, n):
        for i in range(n - j):
            coef[i][j] = (coef[i+1][j-1] - coef[i][j-1]) / (x[i+j] - x[i])

    result = coef[0][0]
    for j in range(1, n):
        term = coef[0][j]
        for k in range(j):
            term *= (x_val - x[k])
        result += term
    return result

# 3. Интерполяция Ньютона с конечными разностями (1 и 2 формулы)
def newton_forward(x, y, x_val):
    h = x[1] - x[0]
```

```
t = (x_val - x[0]) / h
diff = finite_diff(y)
result = y[0]
for i in range(1, len(x)):
    term = 1.0
    for j in range(i):
        term *= (t - j)
    result += (term / factorial(i)) * diff[0, i]
return result
```

Выводы

В данной работе были реализованы методы интерполяции. Методы были протестированы на различных примерах. Результаты показали, что реализованные алгоритмы успешно справляется с поставленной задачей и находят решения в пределах допустимых погрешностей.