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

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №5

по дисциплине «Вычислительная математика»

Вариант: 7

Преподаватель:
Малышева Татьяна Алексеевна

Выполнил: Жук И. А.
Группа: P3215

Рабочие формулы

Конечные разности (вперёд): $\Delta^0 y_i = y_i,\ \ \Delta^k y_i = \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i.$

Формула Ньютона для интерполирования *назад* (базирование в правом узле x_n):

$$P(x) = y_n + t\,\nabla y_n + \frac{t(t+1)}{2!}\nabla^2 y_n + \frac{t(t+1)(t+2)}{3!}\nabla^3 y_n + \cdots, \quad t = \frac{x - x_n}{h},$$

где $\nabla^k y_n$ — k-е *обратные* разности в точке x_n.

Первая центральная формула Гаусса (база в центральном узле x_m):

$$\begin{aligned} P(x) = y_m &+ \frac{q}{1!}\Delta y_m + \frac{q(q-1)}{2!}\Delta^2 y_{m-1} + \frac{q(q-1)(q+1)}{3!}\Delta^3 y_{m-1} \\ &+ \frac{q(q-1)(q+1)(q-2)}{4!}\Delta^4 y_{m-2} + \frac{q(q-1)(q+1)(q-2)(q+2)}{5!}\Delta^5 y_{m-2} \\ &+ \frac{q(q-1)(q+1)(q-2)(q+2)(q-3)}{6!}\Delta^6 y_{m-3} + \cdots, \qquad q = \frac{x - x_m}{h}. \end{aligned}$$

Вычислительная часть

1. Таблица $y = f(x)$ (Таблица 1.2, вариант №7)

x_i	y_i
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

2. Таблица конечных разностей (вперёд)

i	x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$
0	0,50	1,5320	1,0036	0,0014	−0,0008	−0,0012	0,0059
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. Интерполяция Ньютона (назад) в $X_1 = 0{,}751$

Базируемся в правом узле $x_n = 0{,}80$, $t = \dfrac{0{,}751 - 0{,}80}{0{,}05} = -0{,}98$. Обратные разности в x_n равны последним элементам соответствующих рядов вперёд:

$$y_n = 7{,}5594, \quad \nabla y_n = 1{,}0035, \quad \nabla^2 y_n = -0{,}0020,$$
$$\nabla^3 y_n = -0{,}0033, \quad \nabla^4 y_n = -0{,}0060, \quad \nabla^5 y_n = -0{,}0107, \quad \nabla^6 y_n = -0{,}0166.$$

Тогда

$$\begin{aligned}
P(X_1) &= y_n + t\,\nabla y_n + \frac{t(t+1)}{2}\nabla^2 y_n + \frac{t(t+1)(t+2)}{6}\nabla^3 y_n \\
&\quad + \frac{t(t+1)(t+2)(t+3)}{24}\nabla^4 y_n + \frac{t(t+1)(t+2)(t+3)(t+4)}{120}\nabla^5 y_n \\
&\quad + \frac{t(t+1)(t+2)(t+3)(t+4)(t+5)}{720}\nabla^6 y_n \\
&\approx \boxed{6{,}57603}.
\end{aligned}$$

4. Первая формула Гаусса в $X_2 = 0{,}651$

Центральный узел $x_m = 0{,}65$, $q = \dfrac{0{,}651 - 0{,}65}{0{,}05} = 0{,}02$. Из таблицы разностей:

$$y_m = 4{,}5462, \ \ \Delta y_m = 1{,}0042, \ \ \Delta^2 y_{m-1} = -0{,}0014, \ \ \Delta^3 y_{m-1} = 0{,}0027,$$
$$\Delta^4 y_{m-2} = 0{,}0047, \ \ \Delta^5 y_{m-2} = -0{,}0107, \ \ \Delta^6 y_{m-3} = -0{,}0166.$$

Подставляя в формулу Гаусса, получаем

$$P(X_2) \approx \boxed{4{,}56629}.$$

Программная часть (ключевые фрагменты)

Ниже приведены компактные реализации трёх методов (Лагранжа, Ньютона на разделённых разностях и Гаусса с конечными разностями) и построения таблицы разностей.

```
import numpy as np

def finite_differences_forward(y: np.ndarray) -> np.ndarray:
    n = len(y)
    delta = np.zeros((n, n), dtype=float)
    delta[:, 0] = y
    for j in range(1, n):
        for i in range(n - j):
            delta[i, j] = delta[i + 1, j - 1] - delta[i, j - 1]
    return delta

def lagrange(x, y, x0):
    s = 0.0
    n = len(x)
    for i in range(n):
        L = 1.0
        for j in range(n):
```

```
            if i != j:
                L *= (x0 - x[j]) / (x[i] - x[j])
        s += y[i] * L
    return s

def newton_divided(x, y, x0):
    n = len(x)
    coef = y.astype(float).copy()
    for j in range(1, n):
        for i in range(n - 1, j - 1, -1):
            coef[i] = (coef[i] - coef[i - 1]) / (x[i] - x[i - j])
    # схема Горнера
    p = coef[-1]
    for k in range(n - 2, -1, -1):
        p = coef[k] + (x0 - x[k]) * p
    return p

def newton_backward_from_diffs(x, delta, x0):
    h = x[1] - x[0]
    xn = x[-1]
    t = (x0 - xn) / h
    # обратные разности в xn = последние элементы столбцов delta
    res = delta[-1, 0]
    fact = 1.0
    prod = 1.0
    for k in range(1, len(x)):
        if k == 1:
            prod = t
        else:
            prod *= (t + (k - 1))
        fact *= k
        res += prod / fact * delta[-k, k]
    return res

def gauss_first_central(x, delta, x0):
    h = x[1] - x[0]
    m = len(x) // 2  # центральный узел
    q = (x0 - x[m]) / h
    res = delta[m, 0]
    # члены в точности по формуле Гаусса
    res += q * delta[m, 1]
    res += q * (q - 1) / 2 * delta[m - 1, 2]
    res += q * (q - 1) * (q + 1) / 6 * delta[m - 1, 3]
    res += q * (q - 1) * (q + 1) * (q - 2) / 24 * delta[m - 2, 4]
    res += q * (q - 1) * (q + 1) * (q - 2) * (q + 2) / 120 * delta[m - 2,
    5]
    res += q * (q - 1) * (q + 1) * (q - 2) * (q + 2) * (q - 3) / 720 *
   delta[m - 3, 6]
    return res
```

Тест на исходных данных варианта.

```
x = np.array([0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80])
y = np.array([1.5320, 2.5356, 3.5406, 4.5462, 5.5504, 6.5559, 7.5594])
```

```
D = finite_differences_forward(y)

x1, x2 = 0.751, 0.651
y1_newton = newton_backward_from_diffs(x, D, x1)
y1_lagr   = lagrange(x, y, x1)   # контроль: совпадает
y2_gauss  = gauss_first_central(x, D, x2)

print(y1_newton, y1_lagr, y2_gauss)
# -> 6.5760328698  6.5760328698  4.5662948394
```

Результаты

- $f(0{,}751) \approx \underline{6{,}57603}$ (Ньютон, назад). Контроль по Лагранжу дал то же значение до машинной точности.
- $f(0{,}651) \approx \underline{4{,}56629}$ (первая формула Гаусса).

Выводы

Построена таблица конечных разностей для равномерной сетки ($h = 0{,}05$). Для аргумента, лежащего в правой части интервала, корректно использована *обратная* формула Ньютона; для аргумента вблизи центра — первая центральная формула Гаусса. Численные результаты согласуются между методами (Лагранж/Ньютон), что подтверждает корректность вычислений и реализации.