dМинистерство образования и науки РФ

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 №13

*Выполнил:*

Студент группы P3217

Хачатрян Г.А.

*Проверила:*

Малышева Татьяна Алексеевна,

к.т.н., доцент



Санкт-Петербург, 2025

**Цель работы**

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

Исходные данные:

X1 = 1,168

𝑋2 = 1,463

**Задание**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x\_i | 0 | 0,4 | 0,8 | 1,2 | 1,6 | 2 | 2,4 | 2,8 | 3,2 | 3,6 | 4 |
| y\_i | 0 | 0,952 | 1,849 | 2,468 | 2,537 | 2,138 | 1,611 | 1,656 | 0,842 | 0,617 | 0,461 |

**Вычислительная реализация задачи**

Таблица конечных разностей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **xi** | **yi** | **∆yi** | **∆y2i** | **∆y3i** | **∆y4i** | **∆y5i** | **∆y6i** |
| 1,10 | 0,2234 | 1,0204 | 0,0002 | 0,0132 | -0,0368 | 0,0762 | -0,1313 |
| 1,25 | 1,2438 | 1,0206 | 0,0134 | -0,0236 | 0,0394 | -0,0551 |  |
| 1,40 | 2,2644 | 1,034 | -0,0102 | 0,0158 | -0,0157 |  |  |
| 1,55 | 3,2984 | 1,0238 | 0,0056 | 0,0001 |  |  |  |
| 1,70 | 4,3222 | 1,0294 | 0,0057 |  |  |  |  |
| 1,85 | 5,3516 | 1,0351 |  |  |  |  |  |
| 2,00 | 6,3867 |  |  |  |  |  |  |

**Вычисление значения функции для аргумента 𝑋1 с использованием 1 интерполяционной формулы Ньютона**

Воспользуемся первой интерполяционной формулой, так как точка находится слева от середины отрезка

Вычисление значения функции для аргумента X2 с использованием интерполяционной формулы Гаусса

𝑋2 = 1,463 (центр отрезка). A = 1,55 Т.к. X2<a, воспользуемся Второй интерполяционной формулой Гаусса.

**Листинг программы**

def lagrange(dots: Dots, X: float) -> float:

n = dots.get\_n()

xs = dots.get\_xs()

ys = dots.get\_xs()

res = 0

for i in range(n):

y\_i = ys[i]

q = 1

for j in range(n):

if i != j:

q \*= (X - xs[j]) / (xs[i] - xs[j])

print("l\_%d = %s" % (i, q))

res += y\_i \* q

return rounded(res)

def newton\_div(dots: Dots, X: float) -> float:

factors = get\_factors(dots)

n = dots.get\_n()

xs = dots.get\_xs()

ys = dots.get\_ys()

result = ys[0]

for i in range(1, n):

q = factors[0][i]

for j in range(0, i):

q \*= (X - xs[j])

result += q

return rounded(result)

def newton\_equidistant(dots: Dots, X: float) -> float:

n = dots.get\_n()

xs = dots.get\_xs()

if not \_check\_equidistancy(dots):

raise ValueError('Точки не равноудалены')

if X > (xs[len(xs) // 2 + len(xs) % 2]):

result = newton\_equidistant\_backward(dots, X)

return result

h = xs[1] - xs[0]

t = (X - xs[0]) / h

finite\_diff = get\_finite\_diffs(dots)

result = finite\_diff[0][0]

factor = 1

for i in range(1, n):

factor \*= (t - (i - 1)) / i

result += finite\_diff[0][i] \* factor

return rounded(result)

def newton\_equidistant\_backward(dots: Dots, X: float) -> float:

n = dots.get\_n()

xs = dots.get\_xs()

ys = dots.get\_ys()

if not \_check\_equidistancy(dots):

raise ValueError('Точки не равноудалены')

if n < 2:

if n == 1:

return rounded(ys[0])

return 0.0

h = xs[1] - xs[0]

t = (X - xs[n - 1]) / h

finite\_diffs = get\_finite\_diffs(dots)

result = finite\_diffs[n-1][0]

factor = 1

for i in range(1, n):

factor \*= (t + (i - 1)) / i

# Индекс для конечной разности в таблице get\_finite\_diffs

# Для k-ой разности назад, используемой с y\_n, это будет diffs[n-1-k][k]

diff\_index\_row = n - 1 - i

diff\_index\_col = i

if diff\_index\_row < 0 or diff\_index\_col >= len(finite\_diffs[0]):

# Это условие предотвращает выход за пределы таблицы разностей

# Такое может случиться, если n мало или таблица разностей неполная

break

term\_diff = finite\_diffs[diff\_index\_row][diff\_index\_col]

result += term\_diff \* factor

return rounded(result)

**Результаты выполнения программы**

Выберите действие [1-2]:

1. Выбрать функцию

2. Ввести функцию

1

Выберите функцию [1-3]:

1. первая

2. вторая

3. третья

3

Таблица конечных разностей:

x y Δ1y Δ2y Δ3y Δ4y Δ5y Δ6y

0 1.10 0.2234 1.0204 0.0002 0.0132 -0.0368 0.0762 -0.1313

1 1.25 1.2438 1.0206 0.0134 -0.0236 0.0394 -0.0551

2 1.40 2.2644 1.034 -0.0102 0.0158 -0.0157

3 1.55 3.2984 1.0238 0.0056 0.0001

4 1.70 4.3222 1.0294 0.0057

5 1.85 5.3516 1.0351

6 2.00 6.3867

Введите точку поиска:

2

Многочлен Лагранжа

l\_0 = 0.0

l\_1 = -0.0

l\_2 = 0.0

l\_3 = -0.0

l\_4 = 0.0

l\_5 = -0.0

l\_6 = 1.0

Y = 2.0

Многочлен Ньютона с разделенными разностями

Y = 6.3867

Полином Ньютона с разделенными разностями: 0.2234 + 6.802667\*(x - 1.1) + 0.004443\*(x - 1.1)\*(x - 1.25) + 0.651853\*(x - 1.1)\*(x - 1.25)\*(x - 1.4) + -3.028807\*(x - 1.1)\*(x - 1.25)\*(x - 1.4)\*(x - 1.55) + 8.362152\*(x - 1.1)\*(x - 1.25)\*(x - 1.4)\*(x - 1.55)\*(x - 1.7) + -16.009809\*(x - 1.1)\*(x - 1.25)\*(x - 1.4)\*(x - 1.55)\*(x - 1.7)\*(x - 1.85)

Многочлен Ньютона с конечными разностями

Y = 2.0

Обратный обход

Полином обратного обхода: N\_n(t) = 2.0000 + 5.3516\*t/1! + 1.0294\*t(t+1)/2! + 0.0056\*t(t+1)(t+2)/6! + 0.0158\*t(t+1)(t+2)(t+3)/24! + 0.0394\*t(t+1)(t+2)(t+3)(t+4)/120! + 0.0762\*t(t+1)(t+2)(t+3)(t+4)(t+5)/720!

**Вывод**

В результате выполнения данной лабораторной работы я узнал как выполнять интерполяцию для заданного множества точек с помощью методов Ньютона и Лагранжа, а также реализовал их в программе.