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 №18

Выполнил:
Студент группы P3213
Хафизов Булат Ленарович
Преподаватель:
Малышева Татьяна
Алексеевна

УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2022

Цель работы

Изучить методы интерполяции функции и реализовать два из них средствами программирования. Понять их сходства и различия.

Ход работы

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	X_2
0,545	0,627

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$	$\Delta^6 y_i$
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						

Вычисление X_1. Первая интерполяционная формула Ньютона.

$$t = (x - x_0)/h = \frac{0{,}545 - 0{,}5}{0{,}05} = 0{,}9$$

$$y(0{,}545) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + \frac{t(t-1)(t-2)}{3!}\Delta^3 y_0 +$$

$$+ \frac{t(t-1)(t-2)(t-3)}{4!}\Delta^4 y_0 + \frac{t(t-1)(t-2)(t-3)(t-4)}{5!}\Delta^5 y_0 +$$

$$+ \frac{t(t-1)(t-2)(t-3)(t-4)(t-5)}{6!}\Delta^6 y_0 = 2{,}4353$$

Вычисление X_2. Первая интерполяционная формула Ньютона.

$$t = (x - x_2)/h = \frac{0{,}627 - 0{,}6}{0{,}05} = 0{,}54$$

$$y(0{,}627) = y_2 + t\Delta y_2 + \frac{t(t-1)}{2!}\Delta^2 y_2 + \frac{t(t-1)(t-2)}{3!}\Delta^3 y_2 +$$

$$+ \frac{t(t-1)(t-2)(t-3)}{4!}\Delta^4 y_2 = 4{,}0842$$

x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$	$\Delta^6 y_i$
x_{-3}=0,50	1,5320	1,0036	0,0014	-0,0008	-0,0012	0,0059	-0,0166
x_{-2}=0,55	2,5356	1,0050	0,0006	-0,0020	0,0047	-0,0107	
x_{-1}=0,60	3,5406	1,0056	-0,0014	0,0027	-0,0060		
x_0=0,65	4,5462	1,0042	0,0013	-0,0033			
x_1=0,70	5,5504	1,0055	-0,0020				
x_2=0,75	6,5559	1,0035					
x_3=0,80	7,5594						

Вычисление X_1. Первая формула Гаусса.

$$t = (x - x_0)/h = \frac{0{,}545 - 0{,}65}{0{,}05} = -2{,}1$$

$$y(0{,}545) = y_0 + t\Delta y_{-1} + \frac{t(t+1)}{2!}\Delta^2 y_{-1} + \frac{t(t+1)(t-1)}{3!}\Delta^3 y_{-2} +$$
$$+ \frac{t(t+1)(t-1)(t+2)}{4!}\Delta^4 y_{-2} + \frac{t(t+1)(t-1)(t+2)(t-2)}{5!}\Delta^5 y_{-3}$$
$$+ \frac{t(t+1)(t-1)(t+2)(t-2)(t+3)}{6!}\Delta^6 y_{-3} = 2{,}4376$$

Вычисление X_2. Первая формула Гаусса.

$$t = (x - x_0)/h = \frac{0{,}627 - 0{,}65}{0{,}05} = -0{,}46$$

$$y(0{,}627) = y_0 + t\Delta y_{-1} + \frac{t(t+1)}{2!}\Delta^2 y_{-1} + \frac{t(t+1)(t-1)}{3!}\Delta^3 y_{-2} +$$
$$+ \frac{t(t+1)(t-1)(t+2)}{4!}\Delta^4 y_{-2} + \frac{t(t+1)(t-1)(t+2)(t-2)}{5!}\Delta^5 y_{-3}$$
$$+ \frac{t(t+1)(t-1)(t+2)(t-2)(t+3)}{6!}\Delta^6 y_{-3} = 4{,}0835$$

Блок-схемы используемых методов

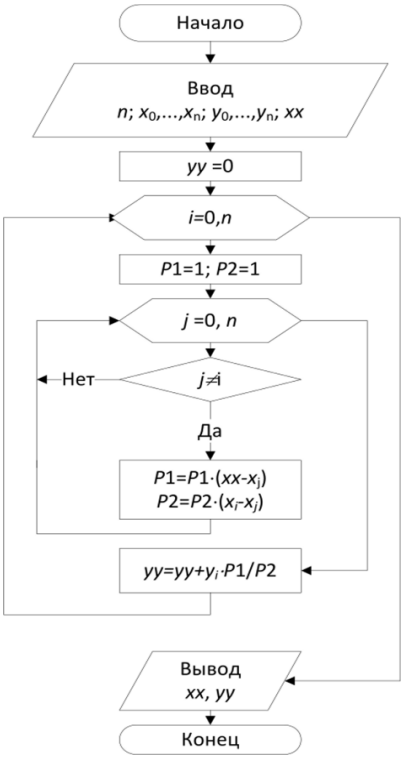

Рисунок 1 - Блок-схема многочлена Лагранжа

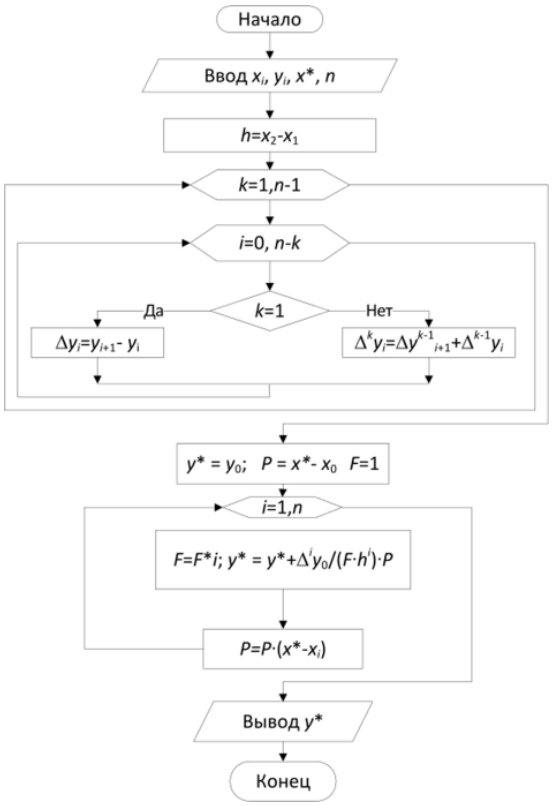

Рисунок 2 - Блок-схема многочлена Ньютона

Листинг программы

```
import numpy as np
import matplotlib.pyplot as plt
```

```
from math import sin, sqrt, factorial


def lagrange(dots, x):
    result = 0
    n = len(dots)
    for i in range(n):
        l1 = l2 = 1
        for j in range(n):
            if i != j:
                l1 *= x - dots[j][0]
                l2 *= dots[i][0] - dots[j][0]
        result += dots[i][1] * l1 / l2
    return result


def t_calc(t, n, forward=True):
    result = t
    for i in range(1, n):
        if forward:
            result *= t - i
        else:
            result *= t + i
    return result


def newton(dots, x):
    n = len(dots)
    h = dots[1][0] - dots[0][0]
    a = [[0] * n for _ in range(n)]
    for i in range(n):
        a[i][0] = dots[i][1]
    for i in range(1, n):
        for j in range(n - i):
            a[j][i] = a[j + 1][i - 1] - a[j][i - 1]
    if x <= dots[n // 2][0]:
        x0 = n - 1
        for i in range(n):
            if x <= dots[i][0]:
                x0 = i - 1
                break
        if x0 < 0:
            x0 = 0
        t = (x - dots[x0][0]) / h
        result = a[x0][0]
        for i in range(1, n):
            result += (t_calc(t, i) * a[x0][i]) / factorial(i)
    else:
        t = (x - dots[n - 1][0]) / h
        result = a[n - 1][0]
        for i in range(1, n):
            result += (t_calc(t, i, False) * a[n - i - 1][i]) / factorial(i)
    return result


def plot(x, y, plot x, plot y):
    plt.gcf().canvas.manager.set_window_title("График")
    ax = plt.gca()
    ax.spines['left'].set_position('zero')
    ax.spines['bottom'].set_position('zero')
    ax.spines['right'].set_color('none')
    ax.spines['top'].set_color('none')
    ax.plot(1, 0, marker=">", ms=5, color='k',
            transform=ax.get_yaxis_transform(), clip_on=False)
```

```
    ax.plot(0, 1, marker="^", ms=5, color='k',
            transform=ax.get_xaxis_transform(), clip_on=False)
    plt.plot(x, y, 'o', plot_x, plot_y)
    plt.show()


def getfunc(func_type):
    if func_type == '1':
        return lambda x: sqrt(x)
    elif func_type == '2':
        return lambda x: x ** 2
    elif func_type == '3':
        return lambda x: sin(x)
    else:
        return None


def make_dots(f, a, b, n):
    dots = []
    h = (b - a) / (n - 1)
    for i in range(n):
        dots.append((a, f(a)))
        a += h
    return dots


def get_data():
    data = {}
    print("Выберите метод интерполяции.")
    print(" 1 — Многочлен Лагранжа")
    print(" 2 — Многочлен Ньютона с конечными разностями")
    while True:
        method = input("Метод решения: ")
        if method == '1' or method == '2':
            break
        print("Метода нет в списке!")
    data['method'] = method
    print("Выберите способ ввода исходных данных.")
    print(" 1 — Набор точек")
    print(" 2 — Функция")
    while True:
        input_type = input("Способ ввода: ")
        if input_type == '1' or input_type == '2':
            break
        print("Метода нет в списке!")
    dots = []
    if input_type == '1':
        print("Вводите координаты через пробел, каждая точка с новой
строки.")
        print("Чтобы закончить, введите 'END'.")
        while True:
            line = input()
            if line == 'END':
                if len(dots) < 2:
                    print("Минимальное количество точек - две!")
                else:
                    break
            x, y = map(float, line.split())
            dots.append((x, y))
    elif input_type == '2':
        print("Выберите функцию.")
        print(" 1 — √x")
        print(" 2 - x²")
        print(" 3 — sin(x)")
```

```
        while True:
            func_type = input("Функция: ")
            func = getfunc(func_type)
            if func is None:
                print("Функции нет в списке!")
            else:
                break
        print("Введите границы отрезка через пробел.")
        a, b = map(float, input("Границы отрезка: ").split())
        if a > b:
            a, b = b, a
        print("Выберите количество узлов интерполяции.")
        while True:
            n = int(input("Количество узлов: "))
            if n < 2:
                print("Количество узлов должно быть целым числом > 1!")
            else:
                break
        dots = make_dots(func, a, b, n)
    data['dots'] = dots
    print("Введите значение аргумента для интерполирования.")
    x = float(input("Значение аргумента: "))
    data['x'] = x
    return data


def main():
    print("Лабораторная работа №5")
    print("Вариант №18")
    print("Интерполяция функций")
    data = get_data()
    x = np.array([dot[0] for dot in data['dots']])
    y = np.array([dot[1] for dot in data['dots']])
    plot_x = np.linspace(np.min(x), np.max(x), 100)
    plot_y = None
    if data['method'] == '1':
        ans = lagrange(data['dots'], data['x'])
        plot_y = [lagrange(data['dots'], x) for x in plot_x]
    elif data['method'] == '2':
        ans = newton(data['dots'], data['x'])
        plot_y = [newton(data['dots'], x) for x in plot_x]
    else:
        ans = None
    if ans is not None:
        plot(x, y, plot_x, plot_y)
    print("Результаты вычисления.")
    print(f"Приближенное значение функции: {ans}")


main()
```

Результаты выполнения программы

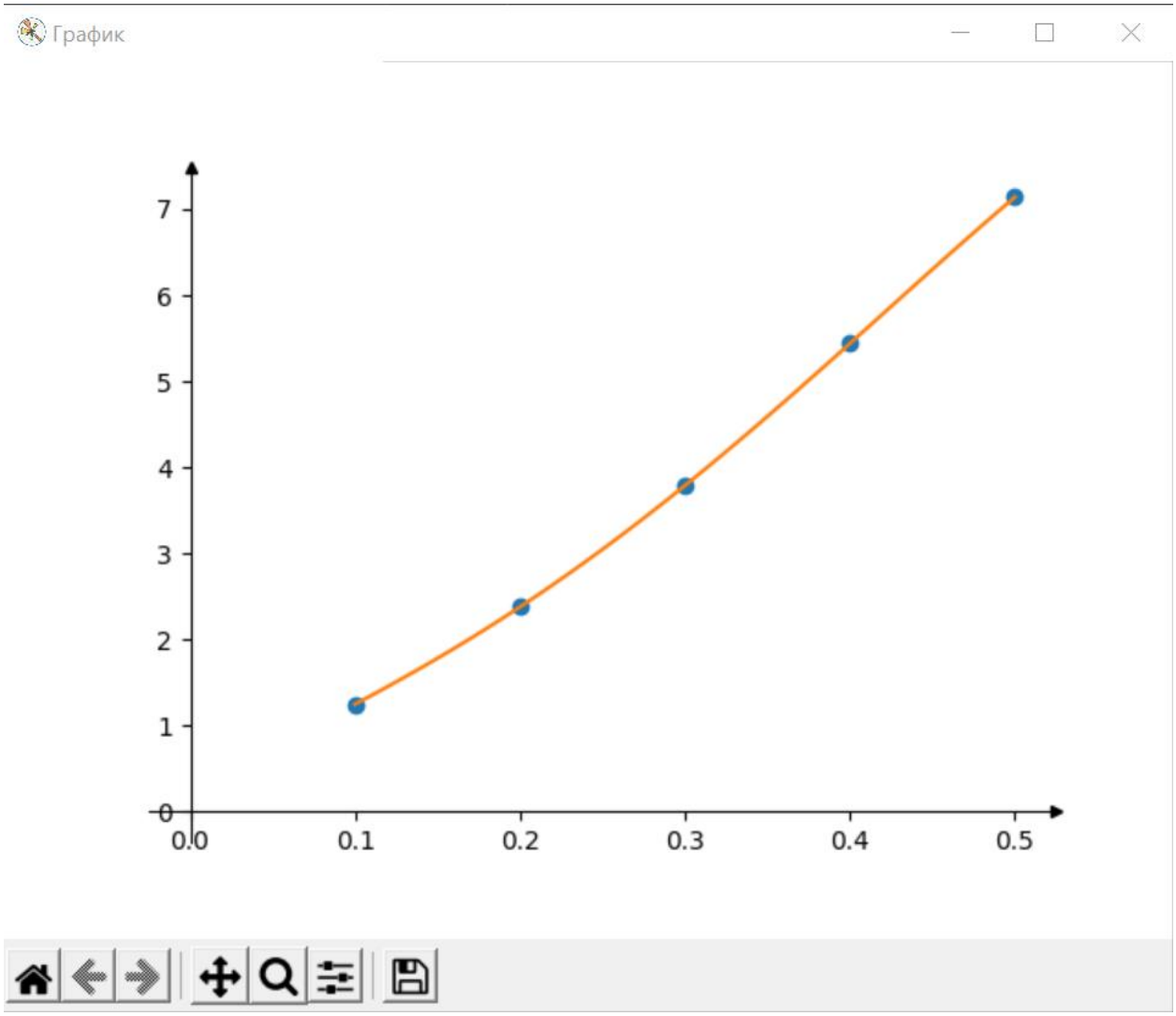
График
7
6
5
4
3
2
1
0
0.0
0.1
0.2
0.3
0.4
0.5

```
Вариант №18
Интерполяция функций
Выберите метод интерполяции.
 1 – Многочлен Лагранжа
 2 – Многочлен Ньютона с конечными разностями
Метод решения: 1
Выберите способ ввода исходных данных.
 1 – Набор точек
 2 – Функция
Способ ввода: 1
Вводите координаты через пробел, каждая точка с новой строки.
Чтобы закончить, введите 'END'.
0.1 1.25
0.2 2.38
0.3 3.79
0.4 5.44
0.5 7.14
END
Введите значение аргумента для интерполирования.
Значение аргумента: 0.15
Результаты вычисления.
Приближенное значение функции: 1.7833593749999992
```

Вывод

В результате выполнения данной лабораторной работой я познакомился с методами интерполяции функции и реализовал метод с использованием многочлена Лагранжа и метод с использованием многочлена Ньютона с конечными разностями на языке программирования Python, закрепив знания.