МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

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

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №5

‘ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА’

Вариант №25

Студент:
Хоанг Ван Куан
Группа P3266

Преподаватель:
Машина Екатерина Александровна

Санкт-Петербург, 2024

1. Цель работы

Цель лабораторной работы: решить задачу интерполяции, найти значения функции при заданных значениях аргумента, отличных от узловых точек. Для исследования использовать:

- многочлен Лагранжа
- многочлен Ньютона
- многочлен Гаусса

2. Порядок выполнения работы

<u>Обязательное задание</u>

- 1 часть: Вычислительная реализация задачи

X	y	X_1	X_2
2,10	3,7587	2,112	2,205
2,15	4,1861	2,355	2,254
2,20	4,9218	2,114	2,216
2,25	5,3487	2,359	2,259
2,30	5,9275	2,128	2,232
2,35	6,4193	2,352	2,284
2,40	7,0839	2,147	2,247

1. Построить таблицу конечных разностей для заданной таблицы. Таблицу отразить в отчете
2. Вычислить значения функции для аргумента $X1$, используя первую или вторую интерполяционную формулу Ньютона. Обратить внимание какой конкретно формулой необходимо воспользоваться
3. Вычислить значения функции для аргумента $X2$, используя первую или вторую интерполяционную формулу Гаусса. Обратить внимание какой конкретно формулой необходимо воспользоваться
4. Подробные вычисления привести в отчете

- 2 часть: Программная реализация

1. Исходные данные задаются тремя способами:
   a) в виде набора данных (таблицы x,y), пользователь вводит значения с клавиатуры
   b) в виде сформированных в файле данных (подготовить не менее трех тестовых вариантов)
   c) на основе выбранной функции, из тех, которые предлагает программа, например, $\sin x$. Пользователь выбирает уравнение, исследуемый интервал и количество точек на интервале (не менее двух функций).
2. Сформировать и вывести таблицу конечных разностей
3. Вычислить приближенное значение функции для заданного значения аргумента, введенного с клавиатуры, указанными методами. Сравнить полученные значения

4. Построить графики заданной функции с отмеченными узлами интерполяции и интерполяционного многочлена Ньютона/Гаусса (разными цветами)
5. Программа должна быть протестирована на различных наборах данных, в том числе и некорректных.
6. Проанализировать результаты работы программы.

3. Необязательное задание

1. Реализовать в программе вычисление значения функции для заданного значения аргумента, введенного с клавиатуры, используя схемы Стирлинга
2. Реализовать в программе вычисление значения функции для заданного значения аргумента, введенного с клавиатуры, используя схемы Бесселя.

4. Рабочие формулы

1. Интерполяционные формулы Ньютона

$$N_n(x) = f(x_0) + \sum_{k=1}^{n} f(x_0, x_1, \dots, x_k) \prod_{j=0}^{k-1} (x - x_j)$$

Для равноотстоящих узлов

$$N_n(x) = y_1 + t\Delta y_1 + \frac{t(t-1)}{2!}\Delta^2 y_1 + \frac{t(t-1)(t-2)}{3!}\Delta^3 y_1 + \frac{t(t-1)(t-2)(t-3)}{4!}\Delta^4 y_1 + \frac{t(t-1)(t-2)(t-3)(t-4)}{5!}\Delta^5 y_1$$

2. Первая интерполяционная формула Гаусса

$$P_n(x) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_{-1} + \frac{(t+1)t(t-1)}{3!}\Delta^3 y_{-1} + \frac{(t+1)t(t-1)(t-2)}{4!}\Delta^4 y_{-2} + \frac{(t+2)(t+1)t(t-1)(t-2)}{5!}\Delta^5 y_{-2} \dots + \frac{(t+n-1)\dots(t-n+1)}{(2n-1)!}\Delta^{2n-1} y_{-(n-1)} + \frac{(t+n-1)\dots(t-n)}{(2n)!}\Delta^{2n} y_{-n}$$

3. Вторая интерполяционная формула Гаусса

$$P_n(x) = y_0 + t\Delta y_{-1} + \frac{t(t+1)}{2!}\Delta^2 y_{-1} + \frac{(t+1)t(t-1)}{3!}\Delta^3 y_{-2} + \frac{(t+2)(t+1)t(t-1)}{4!}\Delta^4 y_{-2} + \dots + \frac{(t+n-1)\dots(t-n+1)}{(2n-1)!}\Delta^{2n-1} y_{-n} + \frac{(\boldsymbol{t}+\boldsymbol{n})(t+n-1)\dots(t-n+\mathbf{1})}{(2n)!}\Delta^{2n} y_{-n}$$

4. Интерполяционные многочлены Стирлинга

$$P_n(x) = y_0 + t\frac{\Delta y_{-1} + \Delta y_0}{2!} + \frac{t^2}{2}\Delta^2 y_{-1} + \frac{t(t^2-1^2)}{3!}\cdot\frac{\Delta^3 y_{-2} + \Delta^3 y_{-1}}{2} + \frac{t^2(t^2-1^2)}{4!}\Delta^4 y_{-2} + \dots + \frac{t(t^2-1^2)(t^2-2^2)\dots(t^2-(n-1)^2)}{(2n)!}\cdot\frac{\Delta^{2n-1} y_{-n} + \Delta^{2n-1} y_{-(n-1)}}{2} + \frac{t^2(t^2-1^2)(t^2-2^2)\dots(t^2-(n-1)^2)}{(2n)!}\Delta^{2n} y_{-n}$$

5. Интерполяционные многочлены Бесселя

$$P_n(x) = \frac{y_0 + y_{-1}}{2} + \frac{1}{8}\frac{\Delta^2 y_{-1} + \Delta^2 y_0}{2} + \frac{3}{128}\frac{\Delta^4 y_{-2} + \Delta^2 y_{-1}}{2} + \dots + (-1)^n\frac{(1\cdot 3\cdot 5\dots(2n-1))^2}{2^{2n}(2n)!}\frac{\Delta^{2n} y_{-n} + \Delta^{2n} y_{-(n-1)}}{2}$$

5. Вычислительная часть

- <u>1 часть: Вычислительная реализация задачи</u>

x	y	X_1	X_2
2,10	3,7587	2,112	2,205
2,15	4,1861	2,355	2,254
2,20	4,9218	2,114	2,216
2,25	5,3487	2,359	2,259
2,30	5,9275	2,128	2,232
2,35	6,4193	2,352	2,284
2,40	7,0839	2,147	2,247

1) Построить таблицу конечных разностей для заданной таблицы. Таблицу отразить в отчете

   Конечные разности функций удобно располагать в таблице:

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

2,10	3,7587	0,4274	0,3083	-0,6171	1,0778	-1,7774	2,9757
2,15	4,1861	0,7357	-0,3088	0,4607	-0,6996	1,1983	
2,20	4,9218	0,4269	0,1519	-0,2389	0,4987		
2,25	5,3487	0,5788	-0,0870	0,2598			
2,30	5,9275	0,4918	0,1728				
2,35	6,4193	0,6646					
2,40	7,0839						

2) Вычислить значения функции для аргумента $X1$, используя первую или вторую интерполяционную формулу Ньютона.

Номер	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	2,10	3,7587	0,4274	0,3083	-0,6171	1,0778	-1,7774	2,9757
1	2,15	4,1861	0,7357	-0,3088	0,4607	-0,6996	1,1983	
2	2,20	4,9218	0,4269	0,1519	-0,2389	0,4987		
3	2,25	5,3487	0,5788	-0,0870	0,2598			
4	2,30	5,9275	0,4918	0,1728				
5	2,35	6,4193	0,6646					
6	2,40	7,0839						

Воспользуемся формулой Ньютона для интерполирования вперед, т.к. $x = 2{,}112$, $x = 2{,}114$, $x = 2{,}128$, $x = 2{,}147$ лежат в левой половине отрезка.

$$N_6(x) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + \frac{t(t-1)(t-2)}{3!}\Delta^3 y_0 + \frac{t(t-1)(t-2)(t-3)}{4!}\Delta^4 y_0 + \frac{t(t-1)(t-2)(t-3)(t-4)}{5!}\Delta^5 y_0 + \frac{t(t-1)(t-2)(t-3)(t-4)(t-6)}{6!}\Delta^6 y_0$$

x	t	$N_6(x)$
2.112	0.24	3.645469
2.114	0.28	3.648865
2.128	0.56	3.785927
2.147	0.94	4.128177

Воспользуемся формулой Ньютона для интерполирования назад, т.к. $x = 2{,}352$, $x = 2{,}355$, $x = 2{,}359$ лежит в второй половине отрезка

$$N_6(x) = y_6 + t\Delta y_5 + \frac{t(t+1)}{2!}\Delta^2 y_4 + \frac{t(t+1)(t+2)}{3!}\Delta^3 y_3 + \frac{t(t+1)(t+2)(t+3)}{4!}\Delta^4 y_2 + \frac{t(t+1)(t+2)(t+3)(t+4)}{5!}\Delta^5 y_1 + \frac{t(t+1)(t+2)(t+3)(t+4)(t+6)}{6!}\Delta^6 y_0$$

x	t	$N_6(x)$
2.352	-0.96	6.432536
2.355	-0.9	6.452021
2.359	-0.82	6.477829

3) Вычислить значения функции для аргумента $X2$, используя первую или вторую интерполяционную формулу Гаусса.

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}=2,10	3,7587	0,4274	0,3083	-0,6171	1,0778	-1,7774	2,9757
x_{-2}=2,15	4,1861	0,7357	-0,3088	0,4607	-0,6996	1,1983	

x_{-1}=2,20	4,9218	0,4269	0,1519	-0,2389	0,4987		
x_0=2,25	5,3487	0,5788	-0,0870	0,2598			
x_1=2,30	5,9275	0,4918	0,1728				
x_2=2,35	6,4193	0,6646					
x_3=2,40	7,0839						

Воспользуемся первой интерполяционной формуле Гаусса для $x = 2{,}254$, $x = 2{,}259$, $x = 2{,}284$

$$P_6(x) = y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_{-1} + \frac{(t+1)t(t-1)}{3!}\Delta^3 y_{-1} + \frac{(t+1)t(t-1)(t-2)}{4!}\Delta^4 y_{-2} + \frac{(t+2)(t+1)t(t-1)(t-2)}{5!}\Delta^5 y_{-2} + \frac{(t+2)(t+1)t(t-1)(t-2)(t-3)}{6!}\Delta^6 y_{-3}$$

x	t	$N_6(x)$
2.254	0.08	5.387469
2.259	0.18	5.438215
2.284	0.68	5.726761

Воспользуемся второй интерполяционной формуле Гаусса для $x = 2{,}205$, $x = 2{,}216$, $x = 2{,}232$, $x = 2{,}247$

$$P_6(x) = y_0 + t\Delta y_{-1} + \frac{t(t+1)}{2!}\Delta^2 y_{-1} + \frac{(t+1)t(t-1)}{3!}\Delta^3 y_{-2} + \frac{(t+2)(t+1)t(t-1)}{4!}\Delta^4 y_{-2} + \frac{(t+2)(t+1)t(t-1)(t-2)}{5!}\Delta^5 y_{-3} + \frac{(t+3)(t+2)(t+1)t(t-1)(t-2)}{6!}\Delta^6 y_{-3}$$

x	t	$N_6(x)$
2.205	-0.9	4.968647
2.216	-0.68	5.062639
2.232	-0.36	5.191328
2.247	-0.06	5.3487

6. Листинг программы

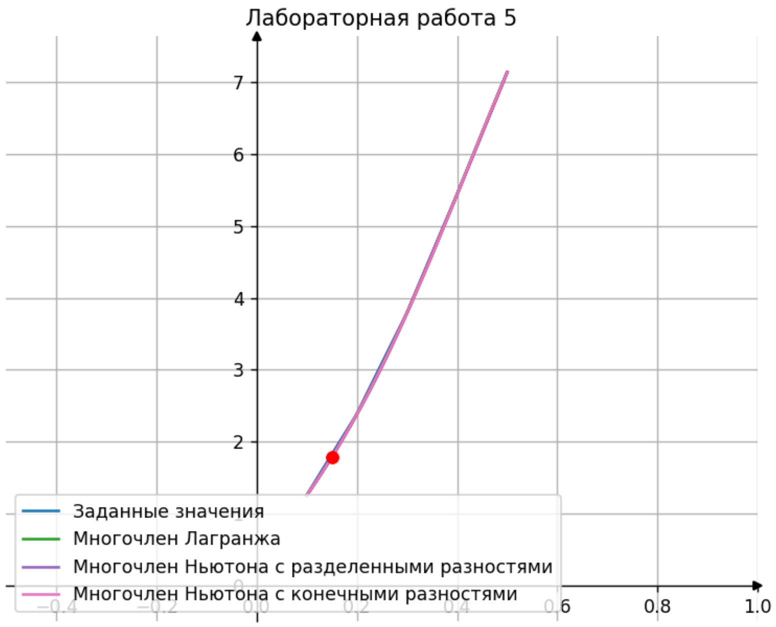
```
import math
import numpy as np
from prettytable import PrettyTable

def FiniteDifferenceTable(x, y):
    table = [[0 for _ in range(len(y) + 1)] for _ in range(len(y))]
    for i in range(len(y)):
        table[i][0] = f"{float(x[i]):.2f}"
        table[i][1] = f"{float(y[i]):.4f}"

    for j in range(2, len(y) + 1):
        for i in range(len(y) - j + 1):
            table[i][j] = float(table[i+1][j-1]) - float(table[i][j-1])
            table[i][j] = f"{table[i][j]:.5f}"

    if(len(y) < 3):_FiniteDifferenceTable = PrettyTable(["x", "y",
*[f"Δ{chr(184+j)}y" for j in range(1, len(y))]])
```

```
    elif(len(y) < 5):_FiniteDifferenceTable = PrettyTable(["x", "y",
*[f"Δ{chr(184+j)}y" for j in range(1, 2)], *[f"Δ{chr(176+j)}y" for j in range(2,
len(y))]])
    else:_FiniteDifferenceTable = PrettyTable(["x", "y", *[f"Δ{chr(184+j)}y" for
j in range(1, 2)], *[f"Δ{chr(176+j)}y" for j in range(2, 4)],
*[f"Δ{chr(8304+j)}y" for j in range(4, len(y))]])
    for row in table:
        _FiniteDifferenceTable.add_row(row)
    print(_FiniteDifferenceTable)

def Lagrange(x, y, value):
    result = 0
    for i in range(len(x)):
        c1 = c2 = 1
        for j in range(len(x)):
            if i != j:
                c1 *= value - x[j]
                c2 *= x[i] - x[j]
        result += y[i] * c1 / c2

    return round(result,5)

def NewtonSeparatedDifferences(x, y, value):
    f = subNewtonSeparatedDifferences_createrTable(x, y)
    result = y[0]
    for j in range(1, len(f[0])):
        temp = f[0][j]
        for i in range(0, j): temp *= (value - x[i])
        result += temp
    return round(result,5)

def subNewtonSeparatedDifferences_createrTable(x, y):
    f = [[0 for _ in range(len(y) )] for _ in range(len(y))]
    for i in range(len(y)):
        f[i][0] = y[i]
    for j in range(1, len(y)):
        for i in range(len(y) - j):
            f[i][j] = (f[i+1][j-1] - f[i][j-1])/(x[i + j] - x[i])
    return f

def subNewton_createrTable(y):
    table = [[0 for _ in range(len(y))] for _ in range(len(y))]
    for i in range(len(y)):table[i][0] = y[i]
    for j in range(1, len(y)):
        for i in range(len(y) - j):
            table[i][j] = table[i+1][j-1] - table[i][j-1]
    return table

def NewtonFiniteDifferences(x, y, value):
    table = subNewton_createrTable(y)
    if value <= x[len(x) - 1]:
```

```
        x0 = 0
        for i in range(len(x) - 1, -1, -1):
            if value >= x[i]:
                x0 = i
                break
        t = (value - x[x0]) / (x[1] - x[0])
        result = table[x0][0]
        for i in range(1, len(table[x0])):
            temp = t
            for yi in range(1, i): temp *= (t - yi)
            result += (temp * table[x0][i]) / math.factorial(i)
    else:
        t = (value - x[len(x) - 1]) / (x[1] - x[0])
        result = table[len(x) - 1][0]
        for i in range(1, len(x)):
            temp = t
            for yi in range(1, i): temp *= (temp + yi)
            result += (temp * table[len(x) - i - 1][i]) / math.factorial(i)

    return round(result,5)

def createrTable_Guass(y):
    result = [y]
    for i in range(len(y) - 1):
        div_dif = []
        for j in range (len(result[i]) - 1):
            diff = result[i][j+1] - result[i][j]
            div_dif.append(diff)
        result.append(div_dif)

    return result

def Stirling(x, y, value):
    if(len(y) % 2 == 0):
        print("Четное число узло. Формула Стирлинга не применяется")
        return
    table = createrTable_Guass(y)
    mid = len(y)//2
    h = x[1] - x[0]
    t = (value - x[mid])/h
    if(abs(t) > 0.25): print("Результат по формуле Стирлинга содержит большую
погрешность")
    result = y[mid]
    for i in range(1, mid + 1):
        mul = 1
        for j in range(1, i):
            mul *= (t * t - j * j)
        result += t * mul * (table[2*i-1][-(i-1) + mid] + table[2 * i - 1][-i +
mid]) / (2 * math.factorial(2*i-1))
        result += t * t * mul * (table[2 * i][-i + mid]) / math.factorial(2*i)
    return result
```

```
def Bessel(x, y, value):
    if(len(y) % 2 != 0):
        print("Нечетное число узло. Формула Бесселя не применяется")
        return
    table = createrTable_Guass(y)
    mid = len(y)//2
    h = x[1] - x[0]
    t = (value - x[mid])/h
    if(abs(t) < 0.25 or abs(t) > 0.75): print("Результат по формуле Бесселя 
содержит большую погрешность")
    result = (y[mid] + y[mid+1])/2 + (t - 0.5)*table[1][mid]
    for i in range(2, mid):
        mul = 1
        for j in range(0, i):
            mul *= (t + math.pow(-1, j)*j)
        n = i - 1
        result += mul * (table[2*n][-n + mid] + table[2*i - 2][-(n-1) + mid]) / 
(2 * math.factorial(2*n))
        result += (t - 0.5) * mul * (table[2*n + 1][-n + mid]) / 
math.factorial(2*n + 1)
    return result
```

7. Результаты выполнения программы
Обязательное здание



```
ЛАБОРАТОРНАЯ РАБОТА
--------------------------------------------
Выберите способ ввода данных:
a) в виде набора данных (x,y)
b) в виде сформированных в файле данных
c) на основе выбранной функции
Способ ввода: b
Выберите файл: input1, input2, input3
Номер файла: 1
+------+--------+---------+---------+----------+----------+
|  x   |   y    |   Δ¹y   |   Δ²y   |   Δ³y    |   Δ⁴y    |
+------+--------+---------+---------+----------+----------+
| 0.10 | 1.2500 | 1.13000 | 0.28000 | -0.04000 | -0.15000 |
| 0.20 | 2.3800 | 1.41000 | 0.24000 | -0.19000 |    0     |
| 0.30 | 3.7900 | 1.65000 | 0.05000 |    0     |    0     |
| 0.40 | 5.4400 | 1.70000 |    0    |    0     |    0     |
| 0.50 | 7.1400 |    0    |    0    |    0     |    0     |
+------+--------+---------+---------+----------+----------+
Заданное значение аргумента: 0.15
Приближенное значения функции по многочлену Лагранжа: 1.78336
Приближенное значения функции по многочлену Ньютона с разделенными разностями: 1.78336
Приближенное значения функции по многочлену Ньютона с конечными разностями: 1.78336
Результат по формуле Стирлинга содержит большую погрешность
Приближенное значения функции по схеме Стирлинга: 1.7833593750000003
Нечетное число узло. Формула Бесселя не применяется
Приближенное значения функции по схеме Бесселя: None
```