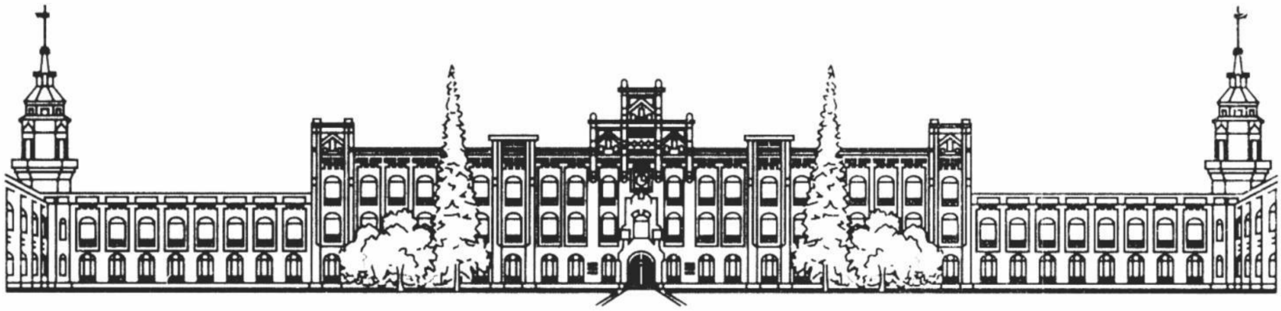


Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського» Інститут прикладного системного аналізу

Лабораторна робота №5

курсу «Чисельні методи 1»

з теми «Інтерполювання функцій»

Варіант №9

Виконав студент 2 курсу групи КА-91

Косицький Вадим Вікторович

перевірила старший викладач

Хоменко Ольга Володимирівна

Київ-2021

Завдання1

1. Для заданої для кожного варіанту функції y=f(x) самостійно обрати відрізок інтерполяції $[a; b]$ та вузли $x_0 = a, x_1, ..., x_n = b$, по яких буде виконуватись інтерполяція. Кількість вузлів – 4 або більше.

2. Визначити значення функції в обраних вузлах та побудувати таблицю скінченних різниць.

3. Написати програму, яка за заданими вузлами будує інтерполяційний поліном Лагранжа, перший інтерполяційний поліном Ньютона, другий інтерполяційний поліном Ньютона та обчислює значення функції в невузлових точках на основі цих поліномів.

4. Використовуючи одержані поліноми обчислити значення функції в кількох невузлових точках (на вибір).

5. Побудувати графіки отриманих поліномів та графік функції f(x) на одному рисунку. Зробити висновки.

Завдання2 (виконати письмово та вставити в звіт)

1. Обрати довільні три вузли по яких буде виконуватись інтерполяція задано функції f(x). Знайти значення функції в цих вузлах.

2. Побудувати за обраними вузлами інтерполяційний кубічний сплайн дефекту 1 (зразок оформлення див. в класрумі приклад «сплайни») з детальними поясненнями.

3. За трьома обраними вище вузлами побудувати інтерполяційний многочле Лагранжа.

4. Побудувати графіки одержаного інтерполяційного кубічного сплайну дефекту 1, полінома Лагранжа та графік функції y=f(x) на одному рисунку. Зробити висновки.

$$9.\quad y = 2\frac{\sqrt{2^x-1}-\operatorname{arctg}\sqrt{2^x-1}}{\ln 2}.$$

Завдання 1

Програмний код:

Файл: main.py

```
from func import *

eps = 0.00001

#main

x = list(map(float, input("Enter nodes: ").split()))
fx =  input("Enter func: ")
f = lambda x:  eval(fx)
values = []
for i in x:
    values.append(f(i))
    print("f(",i,") = ", values[-1] )


deltas = build_table_deltas(x , values )
L = inerpol_polinom_Lagranga(x, f, 1.5)
P = inerpol_polinom_Newtone(x, f, 1.5, 1)
P2= inerpol_polinom_Newtone(x, f, 1.5, 2)

arange = np.linspace(0, 10)
plt.plot(arange, list(map(f, arange)), 'r',arange, list(map(lambdify(t, L), arange)), 'b--'
)
plt.show()
```

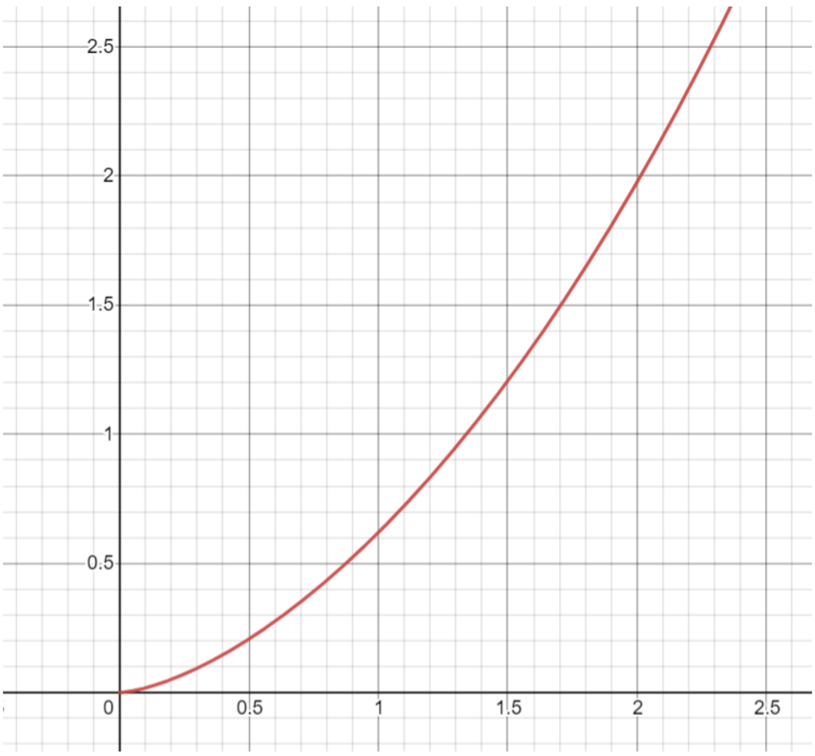
Файл: func.py

```
import numpy as np
import pandas as pd
import pathlib
from pathlib import Path
from math import *
from sympy import *
import matplotlib.pyplot as plt
t = symbols('t')
eps1 = 0.0001
def build_sep_deltas(x , y):
    deltas = []
    deltas.append(y.copy())
    delta = []
    for i in range(len(x)-1):
        for j in range(len(x)-i-1):
            delta.append((deltas[-1][j+1] - deltas[-1][j])/(x[j+i+1]-x[j]))
        deltas.append(delta.copy())
        delta.clear()
    return deltas
def build_table_deltas(x , y):
    print("\nTable deltas")
    print("x: ", x)
    print("y: ", y)
    deltas = []
    deltas.append(y.copy())
    delta = []
    for i in range(len(x)-1):
        for j in range(len(x)-i-1):
            delta.append(deltas[-1][j+1] - deltas[-1][j])
```

```python
        print("Δ^",(i+1)," y: ", delta)
        deltas.append(delta.copy())
        delta.clear()
    return deltas
def inerpol_polinom_Lagranga(x, f, value):
    print("\nInerpolation polinom of Lagrange")
    n = len(x)
    L = sympify(0)
    for i in range(n):
        add = sympify(1)
        for j in range(n):
            if(j!=i): add = add * (t-x[j]) /(x[i] - x[j])
        L = expand(L + add*f(x[i]) ).evalf()
    print("L (t) = ", L)
    print("L(", value, ") = ",L.subs(t, value))
    return L

def inerpol_polinom_Newtone(x, f, value, index):
    print("\nInerpolation polinom of Newtone ", index)
    if(index==2): x.reverse()
    values = [f(i) for i in x]
    deltas = build_sep_deltas(x, values)
    n = len(x)
    P = sympify(0)
    for i in range(n):
        add = sympify(deltas[i][0])
        for j in range(i):
            add = add * (t-x[j])
        P = expand(P + add).evalf()
    print("P (t) = ", P)
    print("P(", value, ") = ",P.subs(t, value))
    return P
```

Тепер подивимось на вигляд заданої функції:



Бачимо, що вона дуже нагадує гілку параболи.

Оберемо наступні вузли: 0, 2, 4, 6, 8

Введемо дані про вузли і нашу функцію в програму, та подивимось на результати.

```
Enter nodes: 0 2 4 6 8
Enter func: 2*(sqrt(2**x-1)-atan(sqrt(2**x-1)))/np.log(2)
f( 0.0 ) =  0
f( 2.0 ) =  1.97606879340989
f( 4.0 ) =  7.37178869425884
f( 6.0 ) =  18.7313333145860
f( 8.0 ) =  41.7240789928376

Table deltas
x:  [0.0, 2.0, 4.0, 6.0, 8.0]
y:  [0, 1.97606879340989, 7.37178869425884, 18.7313333145860, 41.7240789928376]
Δ^ 1  y:  [1.97606879340989, 5.39571990084895, 11.3595446203272, 22.9927456782516]
Δ^ 2  y:  [3.41965110743906, 5.96382471947823, 11.6332010579244]
Δ^ 3  y:  [2.54417361203917, 5.66937633844613]
Δ^ 4  y:  [3.12520272640696]

Inerpolation polinom of Lagrange
L (t) =  0.00813854876668479*t**4 - 0.0446589682827347*t**3 + 0.467530832659118*t**2 + 0.166500214384173*t
L( 1.5 ) =  1.19217208023639

Inerpolation polinom of Newtone  1
P (t) =  0.00813854876668479*t**4 - 0.0446589682827348*t**3 + 0.467530832659117*t**2 + 0.166500214384172*t
P( 1.5 ) =  1.19217208023638

Inerpolation polinom of Newtone  2
P (t) =  0.00813854876668479*t**4 - 0.0446589682827348*t**3 + 0.467530832659117*t**2 + 0.166500214384171*t - 3.5527136788005e-15
P( 1.5 ) =  1.19217208023638
```

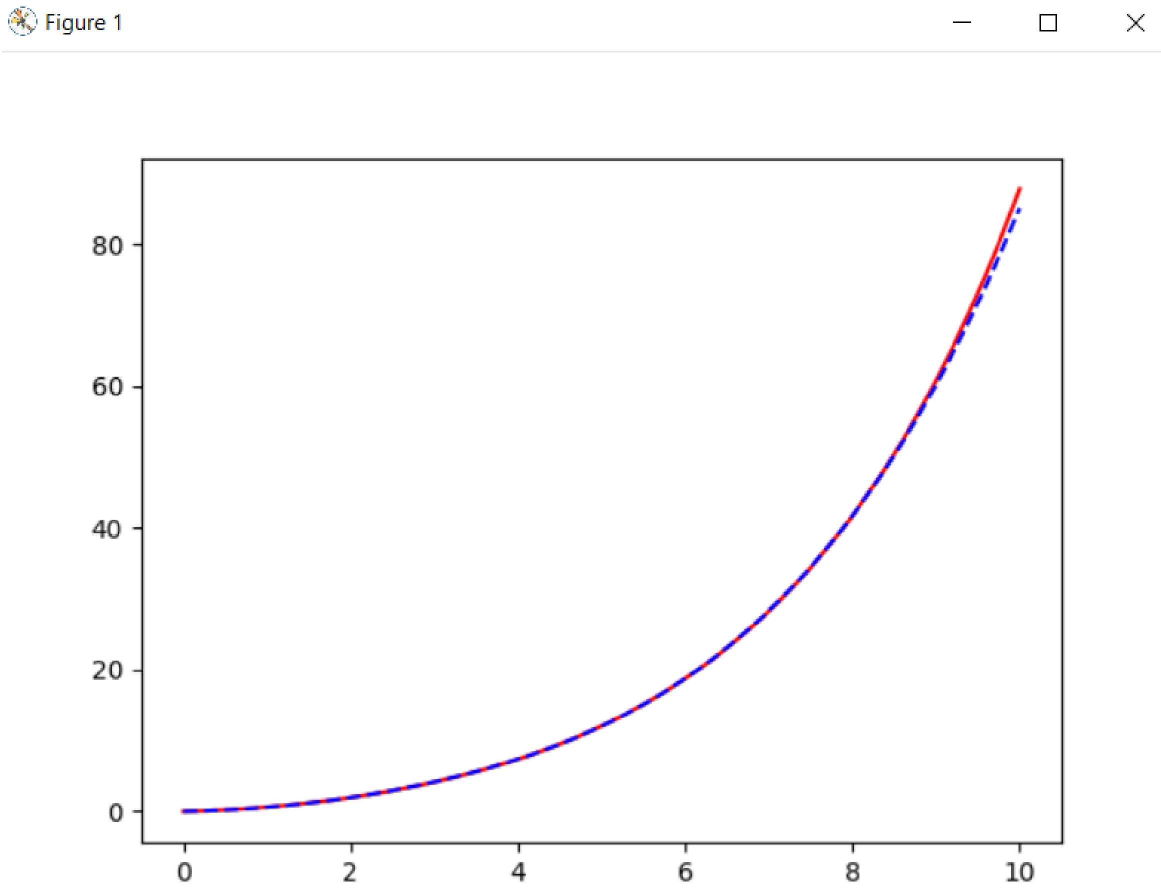
Спочатку програма видає значення функці в вузлових точках. Потім виводить «таблицю» скінчених різниць. Потім побудовані поліноми і значення поліномів в невузловій точці 1.5.

Функція **build_sep_deltas –** повертає масив з розділеними різницями.

Функція **build_table_deltas –** повертає масив зі скінченними різницями.

Функція **inerpol_polinom_Lagranga** – будує інтерполяційний поліном Лагранжа.

Функція **inerpol_polinom_Newtone** – будує як перший так і другий інтерполяційний поліном Н'ютона (параметр index – флаг для зміни.)



Бачимо, що і поліноми усі співпали, і їх графік на інтервалі [0, 8] наче лягає на графік заданої функції.

Висновок:

Виконуючи дане завдання, я навчився будувати інтерполяційні поліноми Лагранжа і Ньютона. Отриманий поліном:

```
0.00813854876668479*t**4 - 0.0446589682827347*t**3 + 0.467530832659118*t**2 + 0.166500214384173*t
```

Також я побудував таблицю скінченних різниць та вивів значення функції у вузлових точках. Функція для Ньютона підходить і для нерівновіддалених вузлів. Також програма не фіксована на одну функцію, і може працювати з уведенною функцією користувача.

Значення полінома в невузловій точці 1.5 ≈1.19217

При цьому в даній точці значення функції ≈ 1.20658

Бачимо, що розходження менше ніж на 0.02 при тому, що ми обрали всього 5 вузлів.

Завдання 2 Косицький Вадим

1) $f(x) = 2\frac{\sqrt{2^x-1} - arctg\sqrt{2^x-1}}{\ln 2}$

Оберемо наступні вузли: $x_0 = 0$, $x_1 = 1$, $x_2 = 2$

$f(x_0) = 0$

$f(x_1) = \frac{4-\pi}{\ln 4} \approx 0.61921$

$f(x_2) = \frac{6\sqrt{3} - 2\pi}{\ln 8} \approx 1.97607$

Отже

x	0	1	2
y	0	0.61921	1.97607

2) Побудуємо інтерполяційний кубічний сплайн дефекту 1

$$g(x) = \begin{cases} a_1 + b_1(x-1) + c_1(x-1)^2 + d_1(x-1)^3, & x \in [0;1] \quad g_1 \\ a_2 + b_2(x-2) + c_2(x-2)^2 + d_2(x-2)^3, & x \in [1;2] \quad g_2 \end{cases}$$

З умов інтерполяції у вузлі $x_0 = 0$:

$a_1 + b_1(0-1) + c_1(0-1)^2 + d_1(0-1)^3 = 0 \Rightarrow$

$\Rightarrow \underline{a_1 - b_1 + c_1 - d_1 = 0}$

З умов інтерполяції у вузлі $x = 1$:

$a_1 + b_1(1-1) + c_1(1-1)^2 + d_1(1-1)^3 = 0.61921 \Rightarrow$

$\Rightarrow \underline{a_1 = 0.61921}$

$a_2 + b_2(1-2) + c_2(1-2)^2 + d_2(1-2)^3 = 0.61921 \Rightarrow$

$\underline{a_2 - b_2 + c_2 - d_2 = 0.61921}$

З умов інтерполяції у вузлі $x = 2$:

$a_2 + b_2(2-2) + c_2(2-2)^2 + d_2(2-2)^3 = 1.97607 \Rightarrow$

$\Rightarrow \underline{a_2 = 1.97607}$

З умов неперервності сплайну:

$g_1(1)=g_2(1);\ g_1'(1)=g_2'(1),\ g_1''(1)=g_2''(1)$

$g_1'(x)=b_1+2c_1(x-1)+3d_1(x-1)^2$

$g_2'(x)=b_2+2c_2(x-2)+3d_2(x-2)^2$

$g_1''(x)=2c_1+6d_1(x-1)$

$g_2''(x)=2c_2+6d_2(x-2)$

$g_1(1)=g_2(1) \Rightarrow \underline{a_1=a_2-b_2+c_2-d_2=0.61921}$

$g_1'(1)=g_2'(1) \Rightarrow \underline{b_1=b_2-2c_2+3d_2}$

$g_1''(1)=g_2''(1) \Rightarrow 2c_1=2c_2-6d_2 \Rightarrow \underline{c_1=c_2-3d_2}$

З умови $g_1''(0)=0$ і $g_2''(2)=0$ маємо:

① $2c_1-6d_1=0 \Rightarrow \underline{c_1=3d_1}$

② $2c_2=0 \Rightarrow \underline{c_2=0}$

Отже маємо:

$$
\begin{cases}
a_1-b_1+c_1-d_1=0 \\
a_1=0.61921 \\
a_1=a_2-b_2+c_2-d_2=0.61921 \\
a_2=1.97607 \\
b_1=b_2-2c_2+3d_2 \\
c_1=c_2-3d_2,\ c_1=3d_1,\ c_2=0
\end{cases}
$$

$$\begin{cases} 0.61921 - b_1 + 3d_1 - d_1 = 0 \\ 1.97607 - b_2 - d_2 = 0.61921 \\ b_1 = b_2 + 3d_2 \\ c_1 = -3d_2 = 3d_1 \end{cases}$$

$$\begin{cases} 0.61921 - b_1 + 2d_1 = 0 \\ b_2 + d_2 = 1.35686 \\ b_1 = b_2 + 3d_2 \\ c_1 = -3d_2 = 3d_1 \end{cases} \qquad \begin{cases} b_1 = 2d_1 + 0.61921 \\ b_1 + 3d_1 - d_1 = 1.35686 \\ b_1 = b_2 + 3d_2 \\ c_1 = -3d_2 = 3d_1 \end{cases} \Rightarrow$$

$$\begin{cases} b_1 = 2d_1 + 0.61921 \\ b_1 = -2d_1 + 1.35686 \end{cases} \Rightarrow b_1 = 0.988035 \Rightarrow d_1 = \frac{0.988035 - 0.61921}{2}$$

$$= 0.1844125 \Rightarrow d_2 = -0.1844125 \Rightarrow c_1 = 0.5532375 \Rightarrow$$
$$\Rightarrow b_2 = b_1 - 3d_2 = 1.5412725$$

Отже:

$a_1 = 0.61921$	$b_1 = 0.988035$	$c_1 = 0.5532375$	$d_1 = 0.1844125$
$a_2 = 1.97607$	$b_2 = 1.5412725$	$c_2 = 0$	$d_2 = -0.1844125$

Отже отримали сплайн

$$g(x) = \begin{cases} 0.61921 + 0.988035(x-1) + 0.5532375(x-1)^2 + 0.1844125(x-1)^3, & x \in [0;1] \\ 1.97607 + 1.5412725(x-2) - 0.1844125(x-2)^3, & x \in [1;2] \end{cases}$$

3) Побудуємо інтерполяційний многочлен Лагранжа

Запишемо інтерполяційний многочлен Лагранжа другого степеня:

$$L_2(x) = \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} y_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} y_1 +$$

$$+ \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} y_2$$

$$L_2(x) = \frac{(x-1)(x-2)}{(-1)(-2)} \cdot 0 + \frac{x(x-2)}{1\cdot(-1)} \cdot 0.61921 +$$

$$+ \frac{x(x-1)}{2\cdot 1} 1.97607 =$$

$$= (-x^2+2x) \cdot 0.61921 + \frac{x^2-x}{2} \cdot 1.97607 =$$

$$= x^2\left(\frac{1.97607}{2} - 0.61921\right) + x\left(2\cdot 0.61921 - \frac{1.97607}{2}\right) =$$

$$= 0.368825x^2 + 0.250385x$$

$$L_2(1.5) = 1.20543$$

Тепер побудуємо цсі необхідні графіки.

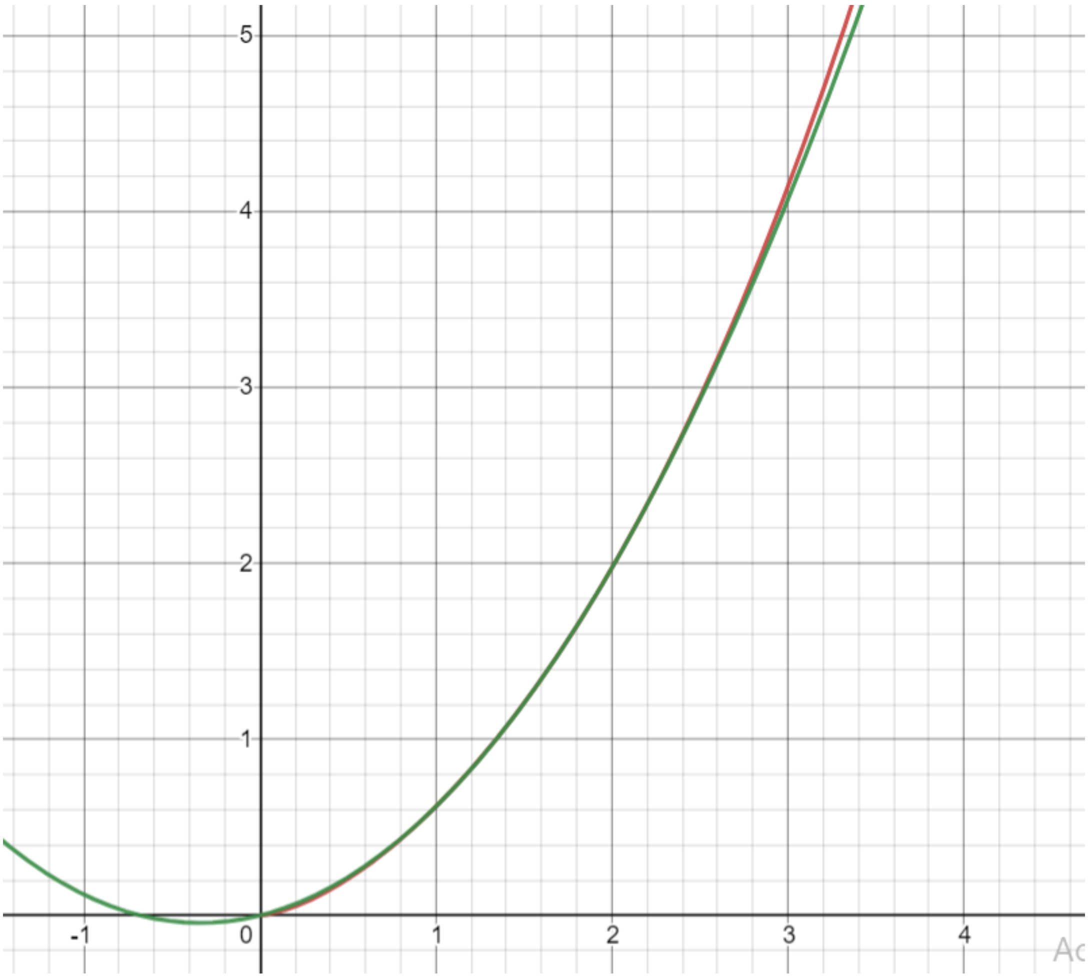
1) Червоний – графік функції f(x)

   Синій – графік g1(x)

   Помаранчевий – графік g2(x)

2) Червоний – графік функції f(x)

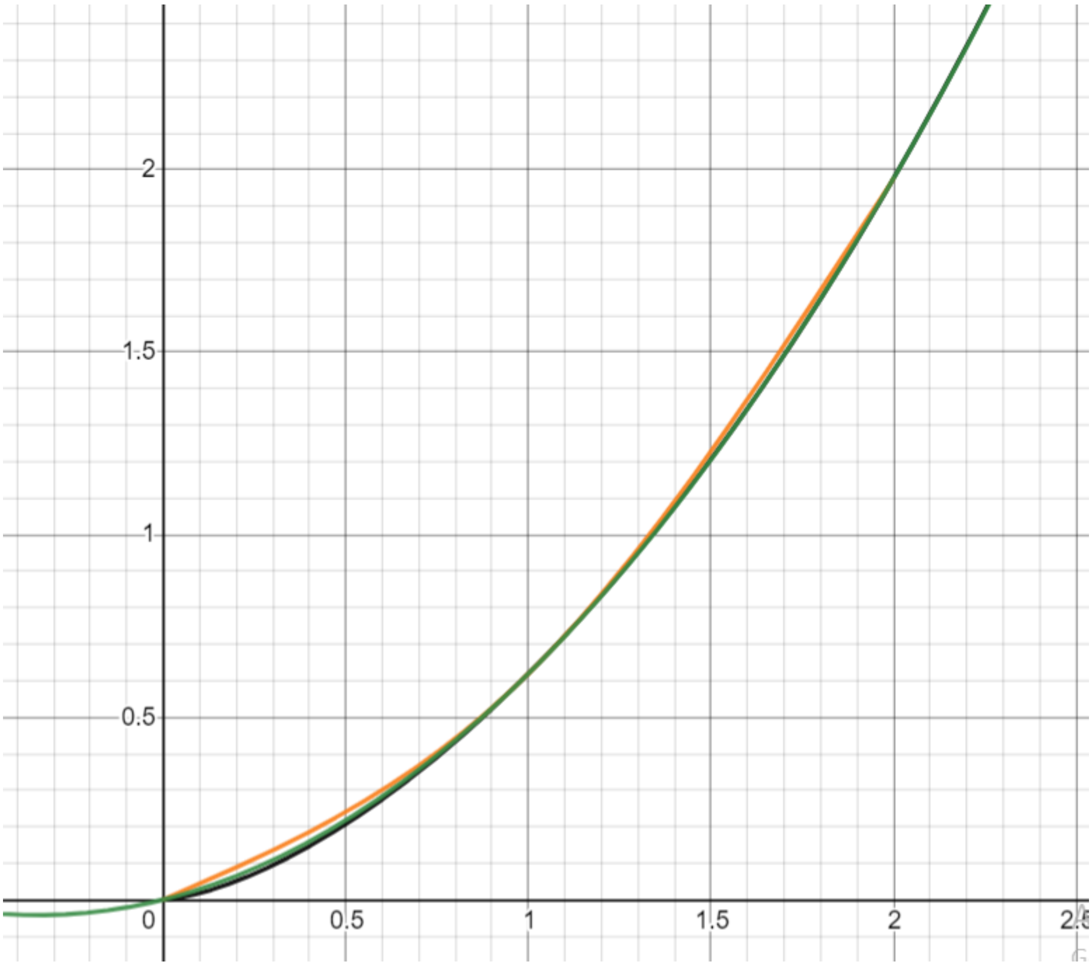
Зелений – графік інтерполяційного поліному Лагранжа 2-го степеня.



3) Чорний - f(x)

Помаранчевий – кубічний сплайн дефекту 1

Зелений - інтерполяційний поліном Лагранжа 2-го степеня



Висновок 2 завдання:

Виконуючи дане завдання я навчився будувати за обраними вузлами кубічний сплайн дефекту 1 та інтерполяційний многочлен Лагранжа.

Отриманий сплайн:

$$g(x) = \begin{cases} 0.61921 + 0.988035(x-1) + 0.5532375(x-1)^2 + 0.1844125(x-1)^3, x \in [0;1] \\ 1.97607 + \ 1.5412725(x-2) - 0.1844125(x-2)^3\ , x \in [1;2] \end{cases}$$

Отриманий інтерполяційни многочлен Лагранжа другого степеня:

$$L_2(x) = 0.368825x^2 + 0.250385x$$

Я побудував графіки цих функцій, та порівняв з тою, якою намагалися інтерполярувати (f(x)). Бачимо, що навіть взявши всього 3 вузли дані криві гарно описують початкову функцію. Слід зауважити, що при знаходжені сплану та поліному Лагранжа були використані певні округлення, при знаходжені значень функцій в вузлах. Точність обчислень $\varepsilon = 0.00001$.