НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС
"ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ"
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ
"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО"
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Лабораторна робота № 4
з предмету "Чисельні методи"
з теми «Наближення функцій»
Варіант № 13

Виконала:
студентка групи
КА-02
Шапошнікова Софія

Перевірила:
Хоменко О.В.

Київ 2022

1 Завдання

1. Для заданої для кожного варіанту функції y=f(x) самостійно обрати відрізок [a;b] інтерполяції 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. Записати інтерполяційний поліном Лагранжа та 1 та 2 інтерполяційні поліноми Ньютона.

4. Використовуючи одержані поліноми обчислити значення функції в кількох невузлових точках (на вибір) та порівняти зі значенням функції в цих точках.

5. Обрати три вузли з тих, які були використані в попередніх пунктах та побудувати за обраними вузлами інтерполяційний кубічний сплайн дефекту 1.

6. Побудувати графіки отриманих поліномів та графік функції f(x) на одному рисунку. Побудову графіків виконати різним кольором, вказавши яким кольором зображено функцію, інтерполяційні поліноми Лагранжа, Ньютона та інтерполяційний кубічний сплайн дефекту 1.

7. Зробити висновок, проаналізувавши отримані графіки.

Варіант 13

$$2x + x^2 - (x + 1)ln(2 + x)$$

Завдання 1: Для заданої для кожного варіанту функції y=f(x) самостійно обрати відрізок [a;b] інтерполяції та вузли.

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

Завдання 1.1

$f(x) = 2x + x^2 - (x+1)\ln(2+x)$

Візьмемо відрізок $[a, b] = [-1, 6]$

Крок: $h = 1{,}75$. Маємо 5 вузлів:

$x_0 = -1$
$x_1 = 0{,}75$
$x_2 = 2{,}5$
$x_3 = 4{,}25$
$x_4 = 6$

Завдання 1.2.

Знайдемо значення ф-ції в обраних вузлах з доп. прогр. засобів (Python):

$y_0 = -1$
$y_1 = 0{,}2921984$
$y_2 = 5{,}98572911$
$y_3 = 16{,}94144732$
$y_4 = 33{,}44390921$

Побудуємо таблицю скінченних різниць:

i	x_i	y_i	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$
0	-1	-1	1,293	4,4013	0,860855	-0,576299
1	0,75	0,292	5,6935	5,2622	0,284556	
2	2,5	5,986	10,9557	5,5467		
3	4,25	16,941	16,5025			
4	6	33,444				

Завдан

$$2x + x^2 - (x+1)ln(2+x)$$

Оберемо відрізок інтерполяції [a, b] та крок. Знайдемо вузли, по яких буде виконуватися інтерполяція

In [30]:
```
X = np.linspace(-1, 6, 5)
h = X[1] - X[0]
```

In [31]:
```
X, h
```

Out[31]:
```
(array([-1.  ,  0.75,  2.5 ,  4.25,  6.  ]), 1.75)
```

In [32]:
```
polinom = lambda x: 2*x+x**2-(x+1)* log(2+x)
```

In [33]:
```
Y = polinom(X)
Y
```

Out[33]:
```
array([-1.        ,  0.2921984 ,  5.98572911, 16.94144732, 33.44390921])
```

Побудуємо таблицю скінченних різниць

In [34]:
```
def ntable(Y):
    table = np.zeros(shape=(len(Y), len(Y)))
    table[0] = Y
    for i in range(1, len(table)):
        for j in range(1, len(table) - (i - 1)):
            table[i, j - 1] = table[i - 1, j] - table[i - 1, j - 1]
    return table
```

In [36]:
```
names_r = ["d^(" + str(i) + ")y" for i in range(len(ntable(Y)))]
names_c = ["y" + str(i) for i in range(len(ntable(Y)))]
t = pd.DataFrame(ntable(Y), index=names_r,
columns=names_c)
print("\nГоризонтальна таблиця скінченних різниць:")
print(t)
```

```
Горизонтальна таблиця скінченних різниць:
              y0        y1         y2         y3         y4
d^(0)y -1.000000  0.292198   5.985729  16.941447  33.443909
d^(1)y  1.292198  5.693531  10.955718  16.502462   0.000000
d^(2)y  4.401332  5.262187   5.546744   0.000000   0.000000
d^(3)y  0.860855  0.284556   0.000000   0.000000   0.000000
d^(4)y -0.576299  0.000000   0.000000   0.000000   0.000000
```

Завдання 3: Записати інтерполяційний поліном Лагранжа і 1ий та 2ий інтерполяційні поліноми Ньютона

Завдання 1.3

Інтерполяційний поліном Лагранжа, 1ий та 2ий поліноми Ньютона

а) Інт. пол. Лагранжа з-мо за ф-лою: $f(x) \approx L_n(x) = \sum_{i=0}^{n} y_i \prod_{j=0, j \ne i}^{n} \frac{x - x_i}{x_i - x_j}$

Запишемо інт. пол. Лагр 4го степеню:

$$L_4(x) = \frac{(x-x_1)(x-x_2)(x-x_3)(x-x_4)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)(x_0-x_4)} \cdot y_0 + \frac{(x-x_0)(x-x_2)(x-x_3)(x-x_4)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)(x_1-x_4)} \cdot y_1 + \frac{(x-x_0)(x-x_1)(x-x_3)(x-x_4)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)(x_2-x_4)} \cdot y_2 +$$
$$+ \frac{(x-x_0)(x-x_1)(x-x_2)(x-x_4)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)(x_3-x_4)} \cdot y_3 + \frac{(x-x_0)(x-x_1)(x-x_2)(x-x_3)}{(x_4-x_0)(x_4-x_1)(x_4-x_2)(x_4-x_3)} \cdot y_4$$

Запишемо інтерполяційний поліном Лагранжа

```python
def lagrange (xi, Y, x):
    s = 0
    for i in range (len(xi)):
        w = 1.0
        for j in range (len(xi)):
            if (i!=j):
                w = w * (x - xi[j])/(xi[i] - xi[j])
        s  = s + w *Y[i]
    return s
```

```python
# Вивід полінома Лагранжа
x_s, h_s = symbols('x h')
X_s = [symbols('x' + str(i)) for i in range(len(X))]
Y_s = [symbols('y' + str(i)) for i in range(len(X))]
print("P(x) = ", lagrange(X_s, Y_s, x_s))
```

```
P(x) =  y0*(1.0*x - 1.0*x1)*(x - x2)*(x - x3)*(x - x4)/((x0 - x1)*(x0 - x2)*(x0 - x3)*(x0 - x4)) + y1*(1.0*x - 1.0*x0)*(x - x2)*(x - x3)*(x - x4)/((-x0 + x1)*(x1 - x2)*(x1 - x3)*(x1 - x4)) + y2*(1.0*x - 1.0*x0)*(x - x1)*(x - x3)*(x - x4)/((-x0 + x2)*(-x1 + x2)*(x2 - x3)*(x2 - x4)) + y3*(1.0*x - 1.0*x0)*(x - x1)*(x - x2)*(x - x4)/((-x0 + x3)*(-x1 + x3)*(-x2 + x3)*(x3 - x4)) + y4*(1.0*x - 1.0*x0)*(x - x1)*(x - x2)*(x - x3)/((-x0 + x4)*(-x1 + x4)*(-x2 + x4)*(-x3 + x4))
```

$$L_4(x) = \frac{(x-0{,}75)(x-2{,}5)(x-4{,}25)(x-6)}{(-1-0{,}75)(-1-2{,}5)(-1-4{,}25)(-1-6)} \cdot (-1) + \frac{(x+1)(x-2{,}5)(x-4{,}25)(x-6)\ .0{,}292}{(0{,}75+1)(0{,}75-2{,}5)(0{,}75-4{,}25)(0{,}75-6)} \cdot 0{,}2921984 +$$

$$+ \frac{(x+1)(x-0{,}75)(x-4{,}25)(x-6)}{(2{,}5+1)(2{,}5-0{,}75)(2{,}5-4{,}25)(2{,}5-6)} \cdot 5{,}98572911 + \frac{(x+1)(x-0{,}75)(x-2{,}5)(x-6)}{(4{,}25+1)(4{,}25-0{,}75)(4{,}25-2{,}5)(4{,}25-6)} \cdot 16{,}94147732 +$$

$$+ \frac{(x+1)(x-0{,}75)(x-2{,}5)(x-4{,}25)}{(6+1)(6-0{,}75)(6-2{,}5)(6-4{,}25)} \cdot 33{,}44390921$$

б) Інт. пол. Ньютона (1ий)

$$P_n(x) = a_0 + a_1(x-x_0) + a_2(x-x_0)(x-x_1) + a_3(x-x_0)(x-x_1)(x-x_2) +$$
$$+ a_4(x-x_0)(x-x_1)(x-x_2)(x-x_3)$$

$$P_4(x) = y_0 + \frac{\Delta y_0}{h}(x-x_0) + \frac{\Delta^2 y_0}{2!\,h^2}(x-x_0)(x-x_1) + \ldots + \frac{\Delta^4 y_0}{4!\,h^4}(x-x_0)(x-x_1)(x-x_2)(x-x_3) =$$

$$= -1 + \frac{1{,}2921984}{1{,}75}(x+1) + \ldots + \frac{-0{,}576299}{4!\cdot(1{,}75)^4}(x+1)(x-0{,}75)(x-2{,}5)(x-4{,}25)$$

в) Інт пол. Ньютона (2ий)

$$P_4(x) = y_4 + \frac{\Delta y_3}{h}(x-x_4) + \frac{\Delta^2 y_2}{2!\,h^2}(x-x_4)(x-x_3) + \ldots + \frac{\Delta^4 y_0}{4!\,h^4}(x-x_4)(x-x_3)(x-x_2)(x-x_1)$$

$$= 33{,}44390521 + \frac{16{,}5025}{1{,}75}(x-6) + \ldots + \frac{(-0{,}576299)}{4!\,(1{,}75)^4}(x-6)(x-4{,}25)(x-2{,}5)(x-0{,}75) =$$

Завдання 1.4 Обчислити значення в кількох невузлових точках

Оберемо 3 точки: $x_1 = 1$, $x_2 = 3$, $x_3 = 5$ (Обчислимо значення з допомогою Python)

Запишемо 1 інтерполяційний поліном Ньютона

```
def newton1(X, T, x, h):
    if type(x) == sympy.core.symbol.Symbol:
        q = symbols('q')
    else:
        q = (x - X[0]) / h
    P, Q, fct = 0, 1, 1
    for i in range(len(X)):
        if type(x) == sympy.core.symbol.Symbol:
            P += Q * T[i] / fct
        else:
            P += Q * T[i, 0] / fct
            fct *= i + 1
            Q *= (q - i)
    return P
```

```
# Вивід полінома Ньютона 1
Y_s = [symbols("d^(" + str(i) + ")y0") for i in range(len(X))]
print("Newton 1: P(x) = ", newton1(X_s, Y_s, x_s, h_s))
```

```
Newton 1: P(x) =  d^(0)y0 + d^(1)y0 + d^(2)y0 + d^(3)y0 + d^(4)y0
```

Запишемо 2 інтерполяційний поліном Ньютона

```
def newton2(X, T, x, h):
    if type(x) == sympy.core.symbol.Symbol:
        q = symbols('q')
    else:
        q = (x - X[len(X) - 1]) / h
    P, Q, fct = 0, 1, 1
    for i in range(len(X)):
        if type(x) == sympy.core.symbol.Symbol:
            P += Q * T[i] / fct
        else:
            P += Q * T[i, (len(X) - 1) - i] / fct
            fct *= i + 1
            Q *= (q + i)
    return P
```

```
# Вивід полінома Ньютона 2
Y_s = [symbols("d^(" + str(i) + ")(Yn-" + str(i) + ")") for i in range(len(X))]
print("Newton 2: P(x) = ", newton2(X_s, Y_s, x_s, h_s))
```

```
Newton 2: P(x) =  d^(0)(Yn-0) + d^(1)(Yn-1) + d^(2)(Yn-2) + d^(3)(Yn-3) + d^(4)(Yn-4)
```

Завдання 4: Використовуючи одержані поліноми обчислити значення функції в кількох невузлових точках (на вибір) та порівняти зі значенням функції в цих точках

Обчислимо значення функції в кількох невузлових точках

```
print("\nВведіть невузлові точки (x1, x2, x3) в інтервалі (", min(X), ",",max(X), "): ")
x1, x2, x3 = input().split()
list_X = np.array([x1, x2, x3], dtype=float)
list_Y = np.ndarray(shape=(len(list_X) + 1, len(list_X)))
for j in range(len(list_X)):
    list_Y[0, j] = polinom(list_X[j])
    list_Y[1, j] = lagrange(X, Y, list_X[j])
    list_Y[2, j] = newton1(X, ntable(Y), list_X[j], h)
    list_Y[3, j] = newton2(X, ntable(Y), list_X[j], h)
names_r = ["Original f(x)", "Lagrange", "Newton 1", "Newton2"]
names_c = ["x" + str(i + 1) for i in range(len(list_X))]
t = pd.DataFrame(list_Y, index=names_r, columns=names_c)
print(t)
```

```
Введіть невузлові точки (x1, x2, x3) в інтервалі ( -1.0 , 6.0 ):
1 3 5
                     x1        x2         x3
Original f(x)  0.802775  8.562248  23.324539
Lagrange       0.809772  8.555731  23.338548
Newton 1       0.809772  8.555731  23.338548
Newton2        0.809772  8.555731  23.338548
```

Завдання 5: Обрати три вузли з тих, які були використані в попередніх пунктах та побудувати за обраними вузлами інтерполяційний кубічний сплайн дефекту 1

Побудова за обраними вузлами інтерполяційного кубічного сплайну дефекту 1

Маємо табл. ф-ю $f(x)$

x	1	3	5
y	0,8098	8,5557	23,3385

Побудуємо ф-ю $g(x)$:

$$g(x)=\begin{cases} a_1+b_1(x-3)+c_1(x-3)^2+d_1(x-3)^3, & x\in[1;3] \\ a_2+b_2(x-5)+c_2(x-5)^2+d_2(x-5)^3, & x\in[3;5] \end{cases}$$

З умов інтерполяції, у вузлі $x=1$:

$$a_1+b_1(1-3)+c_1(1-3)^2+d_1(1-3)^3=0,8098 \Rightarrow$$

$$a_1+b_1(-2)+4c_1+d_1(-8)=0,8098$$

① $\boxed{a_1-2b_1+4c_1-8d_1=0,8098}$

у вузлі $x=3$:

$$a_1+b_1(3-3)+c_1(3-3)^2+d_1(3-3)^3=8,5557$$

② $\boxed{a_1=8,5557}$

$$a_2+b_2(3-5)+c_2(3-5)^2+d_2(3-5)^3=8,5557$$

③ $\boxed{a_2-2b_2+4c_2-8d_2=8,5557}$

у вузлі $x=5$:

$$a_2+b_2(5-5)+c_2(5-5)^2+d_2(5-5)^3=23,3385$$

④ $\boxed{a_2=23,3385}$

З умов неперервності сплайну (в точках стику g_1 і g_2 та їх похідні мають однакові значення)

$g_1(3)=g_2(3)$

$g_1'(3)=g_2'(3)$

$g_1''(3)=g_2''(3)$

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

$$g_2'(x)=b_2+2c_1(x-5)+3d_2(x-5)^2$$

$$g_1''(x)=2c_1+6d_1(x-3)$$

$$g_2''(x)=2c_2+6d_2(x-5)$$

Умову $g_1(3)=g_2(3)=8,5557$ – вже використали:

$a_1=a_2-2b_2+4c_2-8d_2$

З умови $g_1'(3)=g_2'(3)$:

$b_1+2c_1(3-3)+3d_1(3-3)^2=b_2+c_2(3-5)+3d_2(3-5)^2$

(5) $\boxed{b_1=b_2-4c_2+12d_2}$

З умови $g_1''(3)=g_2''(3)$:

$2c_1+6d_1(3-3)=2c_2+6d_2(3-5)$

$2c_1=2c_2-12d_2 \quad |:2$

(6) $\boxed{c_1=c_2-6d_2}$

З умов $g_1''(1)=0$ та $g_2''(5)=0$ (умови рівності 0 2-ої похідної на кінцях)

$2c_1+6d_1(1-3)=0 \Rightarrow 2c_1-12d_1=0 \Rightarrow$ (7) $\boxed{c_1-6d=0}$

$2c_2+6d_2(5-5)=0 \Rightarrow$ (8) $\boxed{c_2=0}$

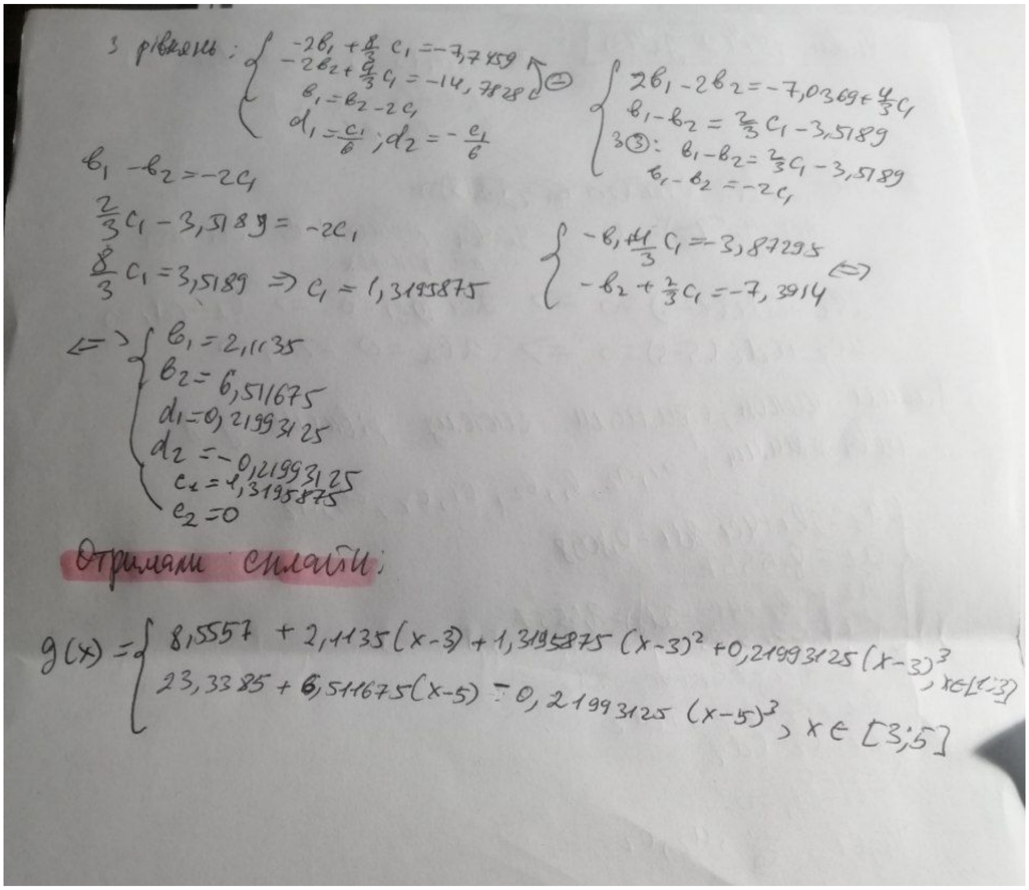
Отримали сист рівн з 8 невідомими: $a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2$:

$$\begin{cases} a_1-2b_1+4c_1-8d_1=0,8098 \\ a_1=8,5557 \\ a_2-2b_2+4c_2-8d_2=8,5557 \\ a_2=23,3385 \\ b_1=b_2-4c_2+12d_2 \\ c_1=c_2-6d_2 \\ c_2=0 \\ c_1=6d_1 \end{cases} \Leftrightarrow$$

Підставимо відомі знач a_1, a_2, c_2:

$$\begin{cases} 8,5557-2b_1+4c_1-8d_1=0,8098 \\ 23,3385-2b_2+4\cdot 0-8d_2=8,5557 \\ b_1=b_2-4\cdot 0+12d_2 \\ c_1=0-6d_2 \\ c_1=6d_1 \end{cases} \Leftrightarrow$$

$$\Leftrightarrow \begin{cases} -2b_1+4c_1-8d_1=-7,7459 \\ -2b_2-8d_2=-14,7828 \\ b_1=b_2+12d_2 \\ d_1=\frac{c_1}{6};\ d_2=-\frac{c_1}{6} \end{cases} \Leftrightarrow \begin{cases} -2b_1+4c_1-\frac{4}{3}c_1=-7,7459 \\ -2b_2+\frac{4}{3}c_1=-14,7828 \\ b_1-b_2=-2c_1 \\ d_1=\frac{c_1}{6};\ d_2=-\frac{c_1}{6} \end{cases}$$



з рівнянь: $\begin{cases} -2b_1 + \frac{8}{3}c_1 = -3,7459 \\ -2b_2 + \frac{4}{3}c_1 = -14,7828 \\ b_1 = b_2 - 2c_1 \\ d_1 = \frac{c_1}{6};\ d_2 = -\frac{c_1}{6} \end{cases}$ ⊖ $\begin{cases} 2b_1 - 2b_2 = -7,0369 + \frac{4}{3}c_1 \\ b_1 - b_2 = \frac{2}{3}c_1 - 3,5189 \\ 3③:\ b_1 - b_2 = \frac{2}{3}c_1 - 3,5189 \\ b_1 - b_2 = -2c_1 \end{cases}$

$b_1 - b_2 = -2c_1$

$\frac{2}{3}c_1 - 3,5189 = -2c_1$

$\frac{8}{3}c_1 = 3,5189 \Rightarrow c_1 = 1,3195875$

$\begin{cases} -b_1 + \frac{4}{3}c_1 = -3,87295 \\ -b_2 + \frac{2}{3}c_1 = -7,3914 \end{cases} \Leftrightarrow$

$\Leftrightarrow \begin{cases} b_1 = 2,1135 \\ b_2 = 6,511675 \\ d_1 = 0,21993125 \\ d_2 = -0,21993125 \\ c_1 = 1,3195875 \\ c_2 = 0 \end{cases}$

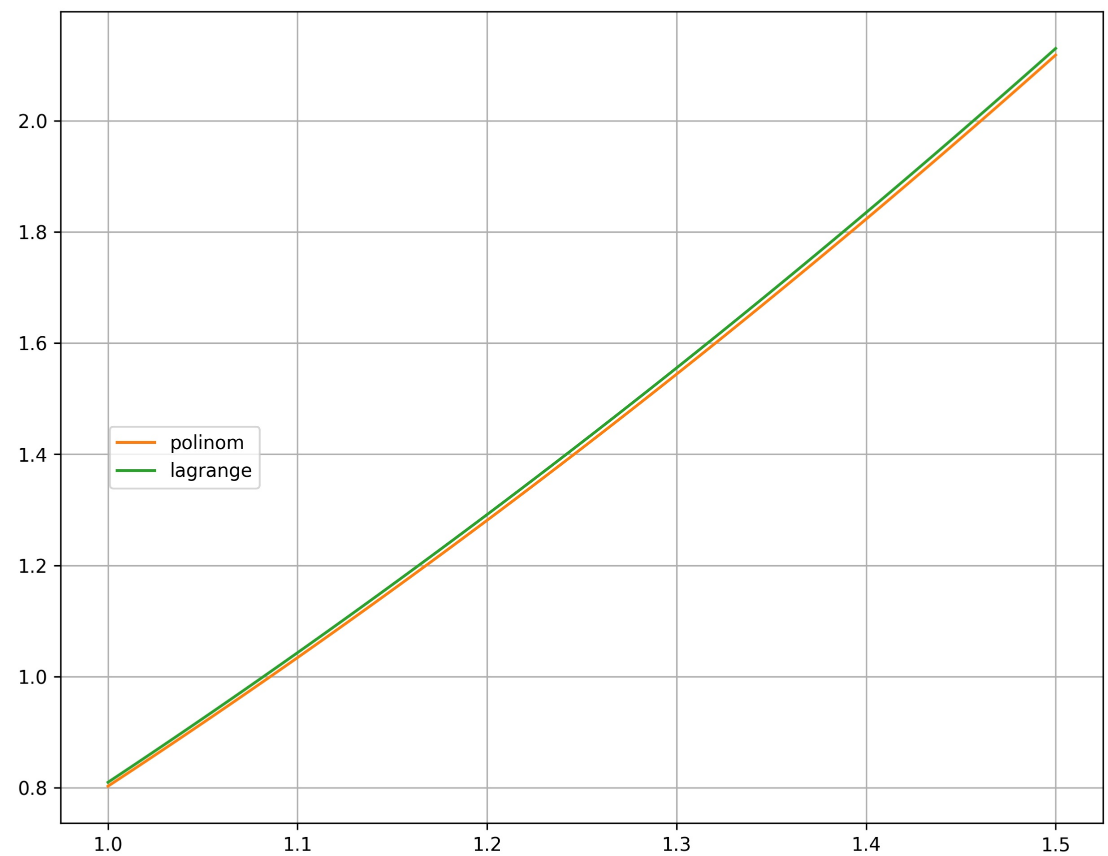
Отримали сплайни:

$$g(x) = \begin{cases} 8,5557 + 2,1135(x-3) + 1,3195875(x-3)^2 + 0,21993125(x-3)^3, & x \in [1;3] \\ 23,3385 + 6,511675(x-5) - 0,21993125(x-5)^3, & x \in [3;5] \end{cases}$$

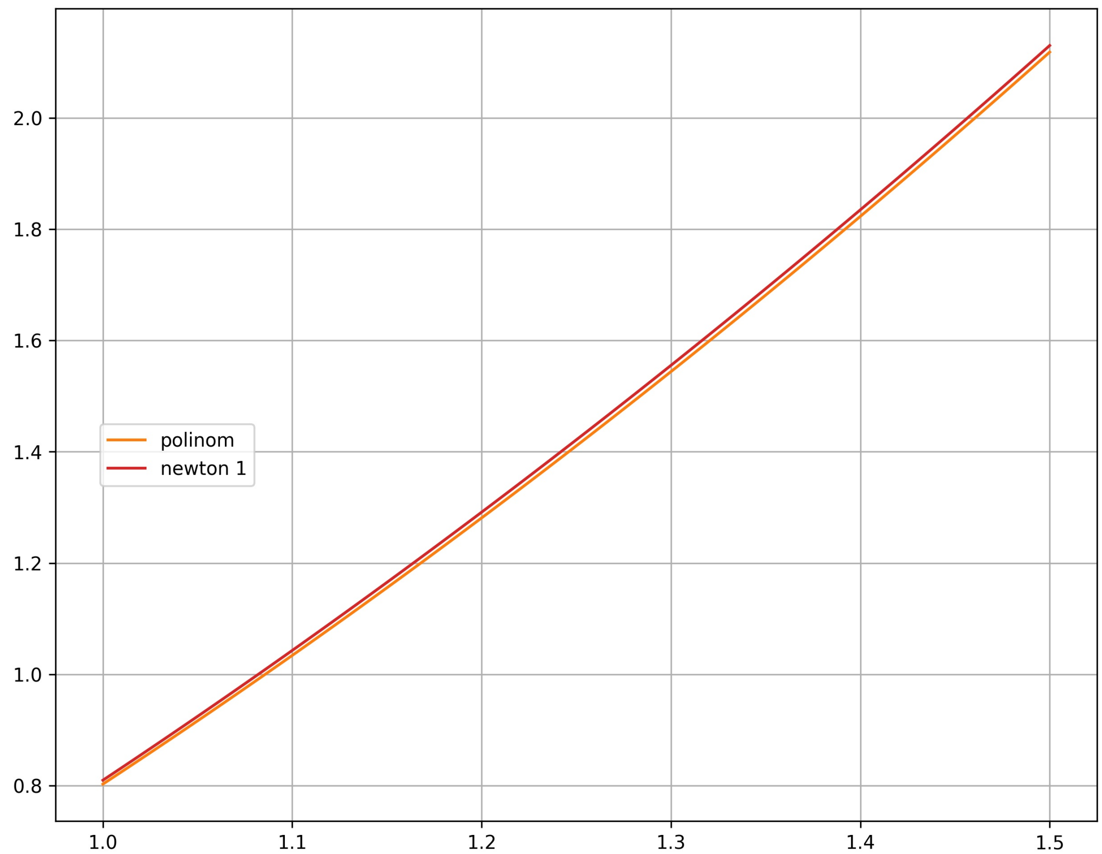
Завдання 6: Побудувати графіки отриманих поліномів та графік функції f(x) на одному рисунку. Побудову графіків виконати різним кольором, вказавши яким кольором зображено функцію, інтерполяційні поліноми Лагранжа, Ньютона та інтерполяційний кубічний сплайн дефекту 1

Зауваження: Графіки розміщенно на різних рисунках, щоб не було абсоютного перекривання одним з графіків інших.

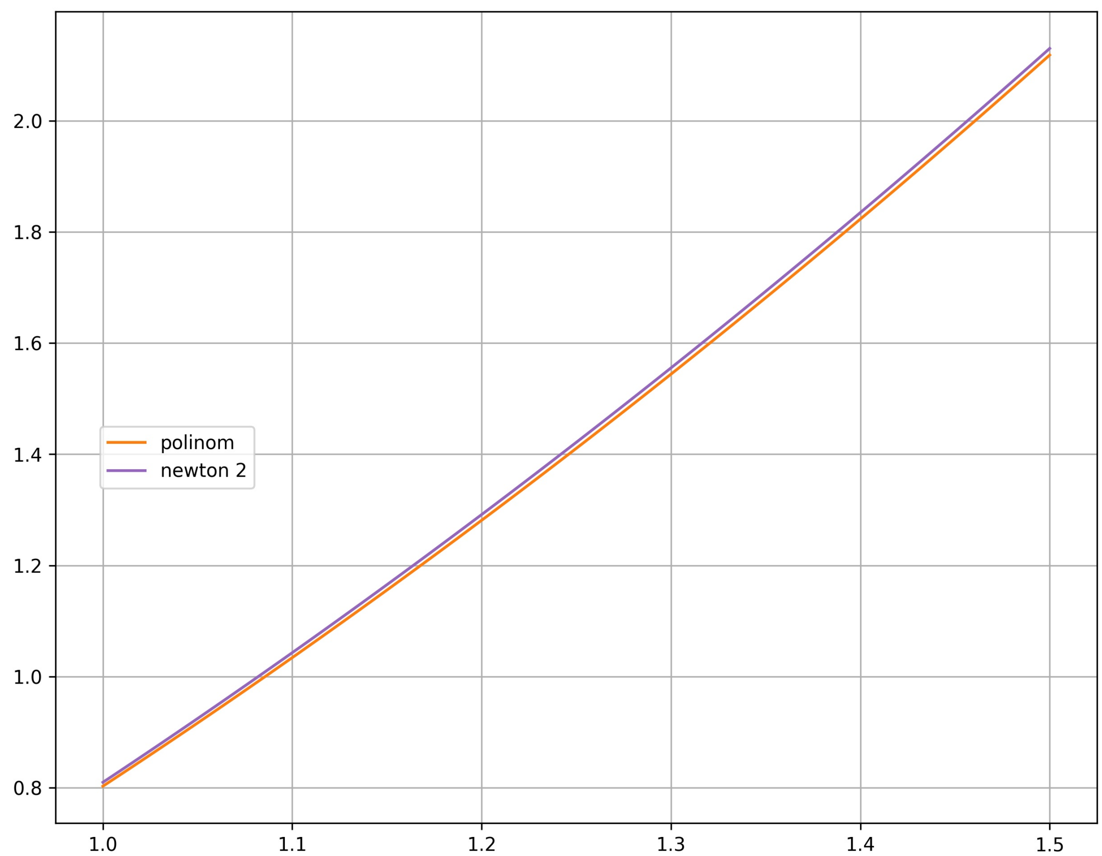
Графік за поліномом Лагранжа :



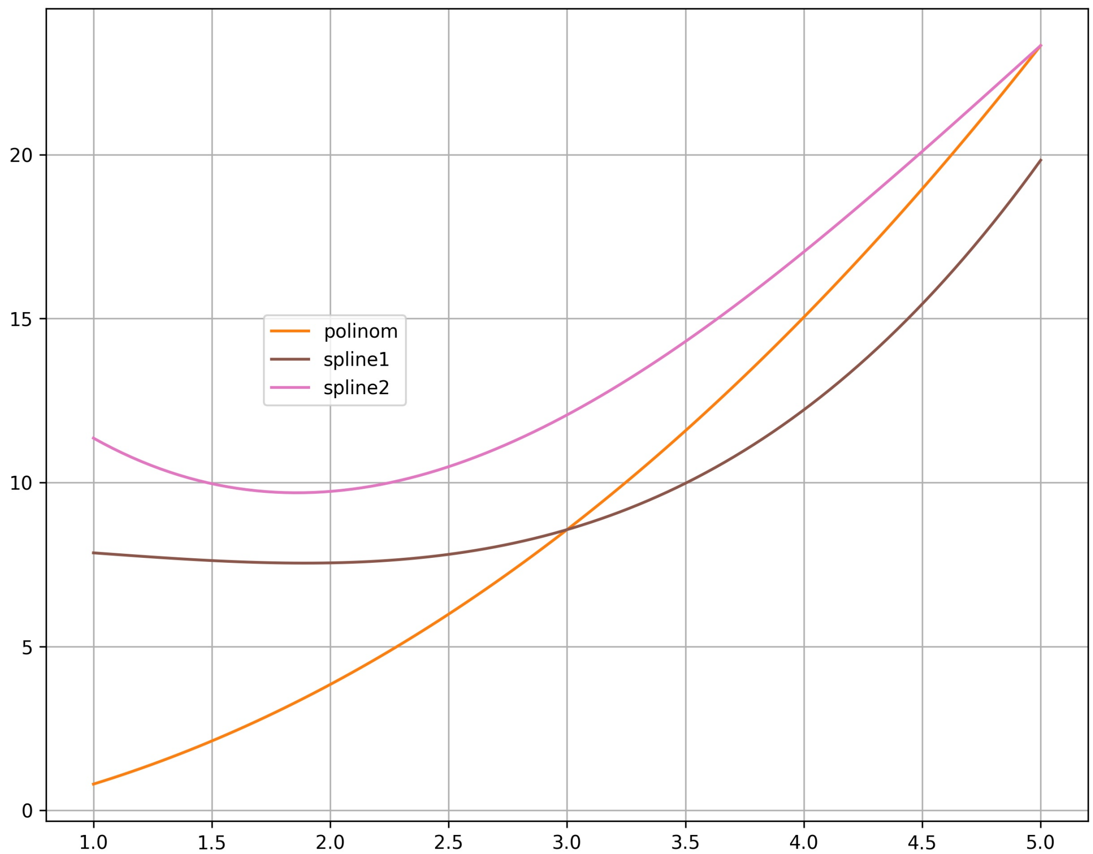
Графік за методом Ньютона 1 :



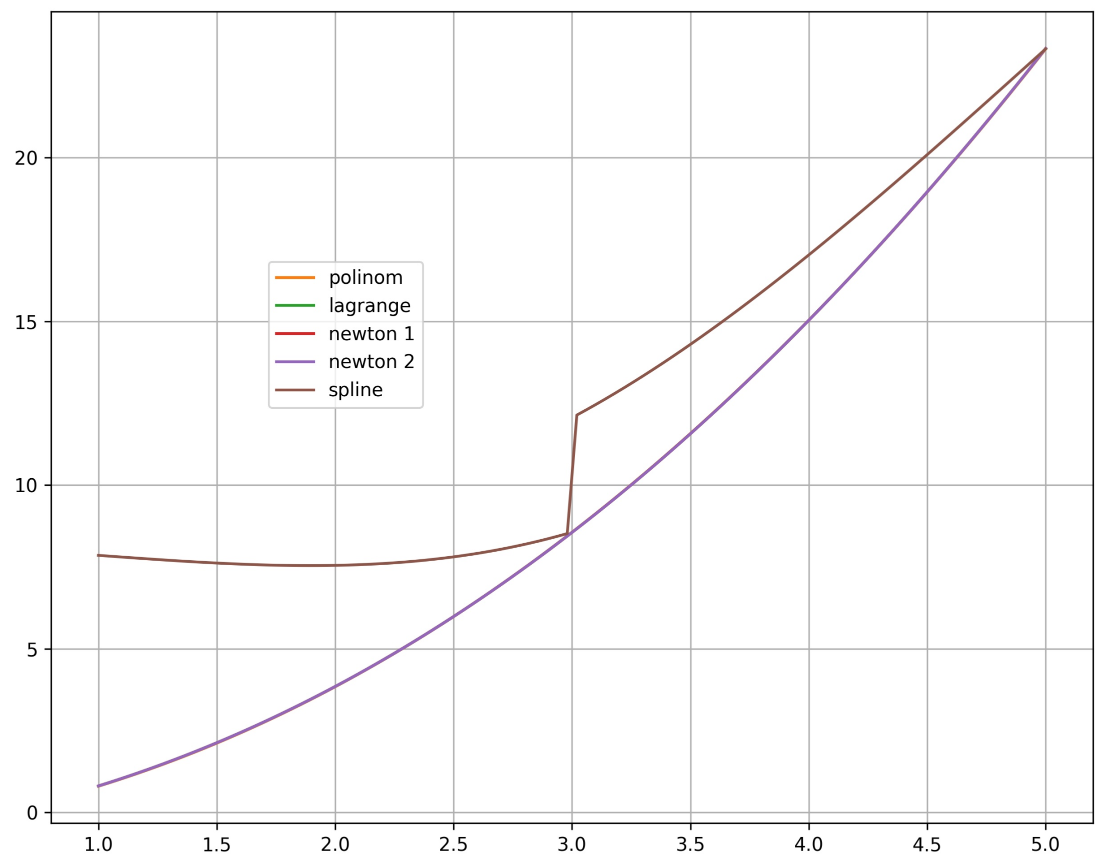
Графік за методом Ньютона 2 :



Графік за кубічним сплайном дефекту 1:

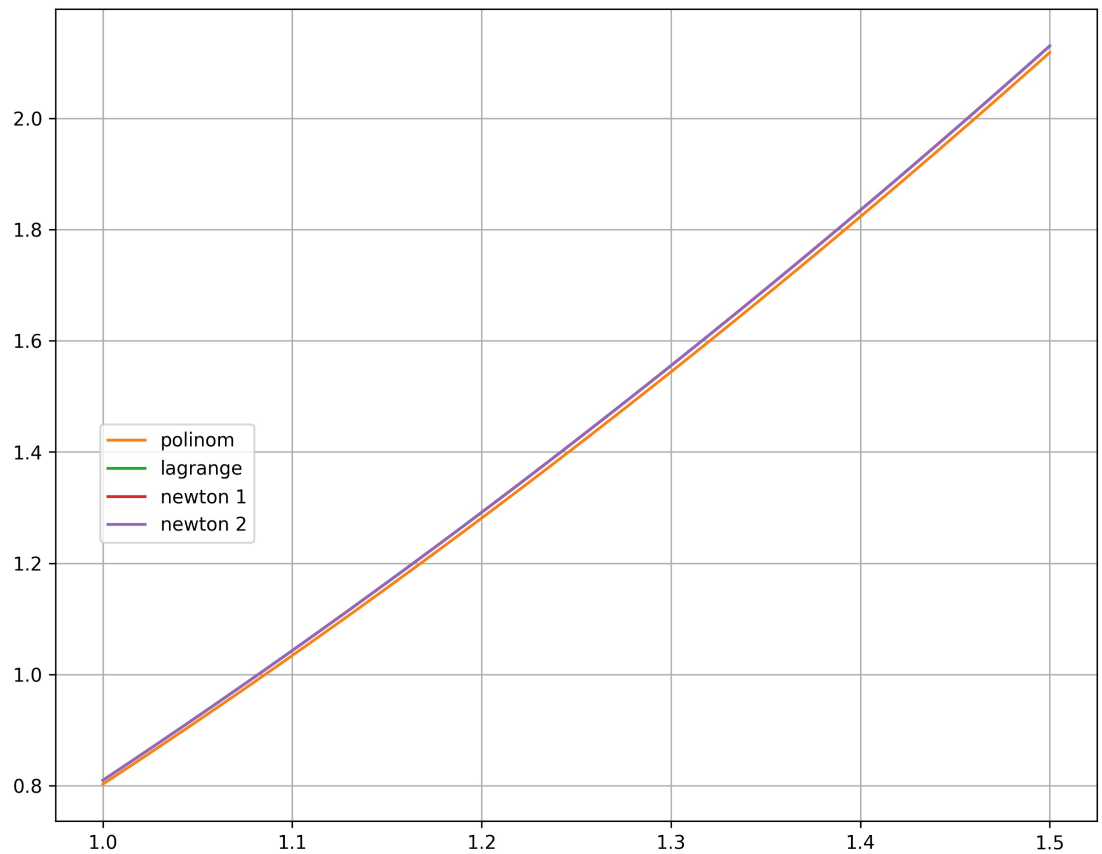


Графік за кубічним сплайном дефекту 1:

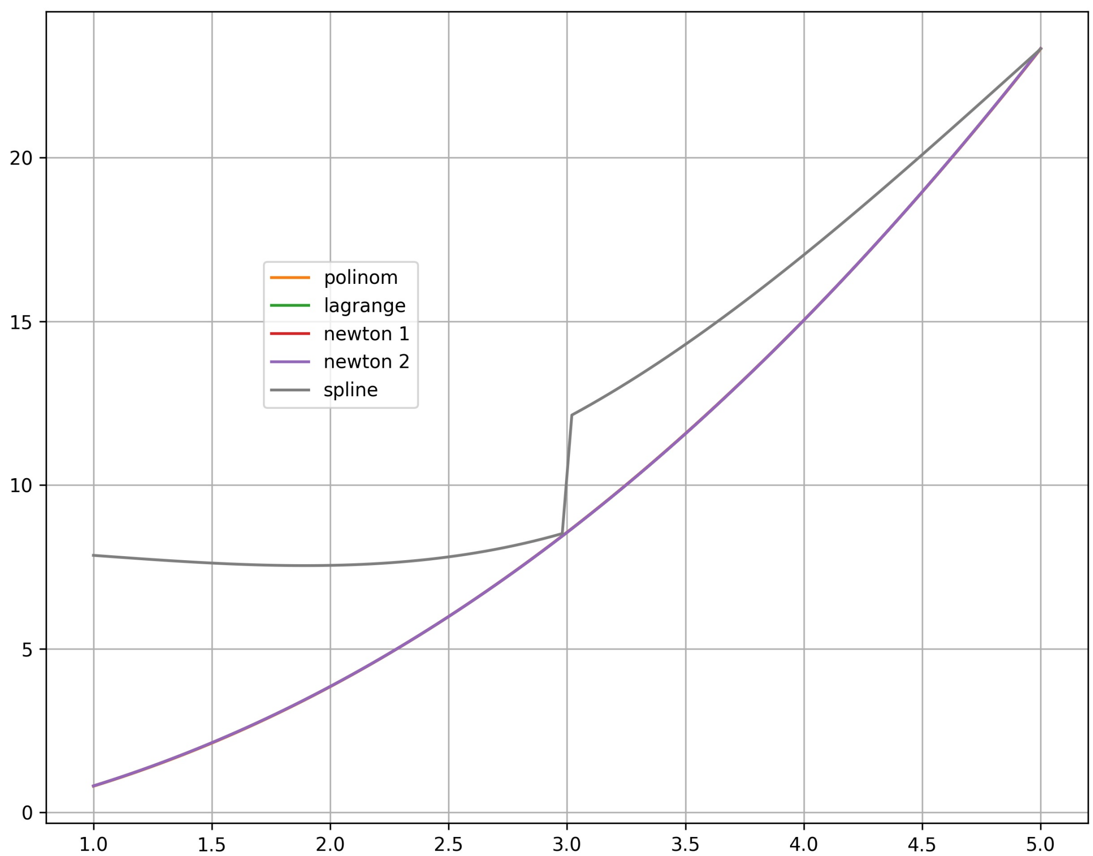


Графік з усіма поліномами :

Бачимо що накладаються графіки з поліномами Ньютона 1, Ньютона 2, Лагранжа.



Графік з усіма поліномами та сплайном:



Завдання 7: Зробити висновок, проаналізувавши отримані графіки

Бачимо, що методи Ньютона 1, 2 та Лагранжа дають хороше наближення. При малій к-ті вузлів зручно обирати метод Лагранжа, в іншому випадку - Ньютона (1 - коли вузли знаходяться ближче до початку відрізку, 2 - коли ближче до кінця). Недоліком формул Ньютона є те, що вони призначені для інтерполяції лише на кінцях відрізків, але інтерполяцію, у більшості випадків, треба виконувати всередині, через великі розміри вибірки. Інтерполяційний кубічний сплайн дефекту 1 вигідно використовувати коли проміжки між вузлами малі, у інших випадках даний метод показує гірші результати, ніж інші методи. Зауважимо, що метод побудови кубічного сплайну є більш ресурсозатратним, це обумовлюється розв'язанням СЛАР.

2 Завдання

1. Нанести точки, задані в таблиці на графік.
2. Обрати клас апроксимуючої функції, наприклад, одну з функцій

$$y = ax + b, \quad y = a + b\ln x, \quad y = ax^b, \quad y = ae^{bx},$$
$$y = a + \frac{b}{x}, \quad y = \frac{1}{ax+b}, \quad y = \frac{x}{ax+b}$$
$$y = ax^2 + bx + c, \quad y = ax^b + c, \quad y = ae^{bx} + c.$$

3. Скласти та розв'язати систему рівнянь для обчислення параметрів. Побудувати на одному графіку точки, задані в таблиці та графік отриманої функції.
4. Зробити висновок, проаналізувавши отримані графіки.

Варіант 13

x	1.00	1.33	1.67	2.00	2.33	2.67	3.00	3.33	3.67	4.00
y	1.28	1.55	1.11	1.48	1.52	1.17	1.44	1.42	1.01	1.01

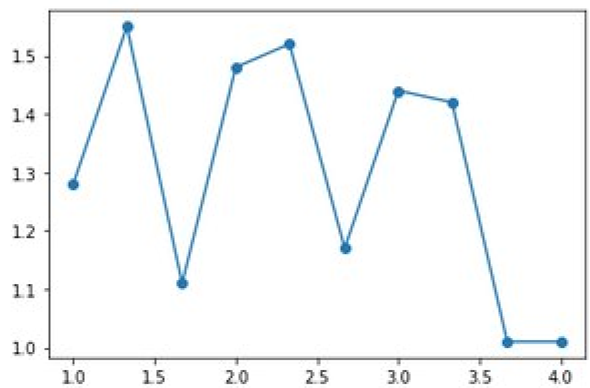
2.1 Нанести точки, задані в таблиці на графік

Нанести точки, задані в таблиці на графік:

x	1.00	1.33	1.67	2.00	2.33	2.67	3.00	3.33	3.67	4.00
y	1.28	1.55	1.11	1.48	1.52	1.17	1.44	1.42	1.01	1.01

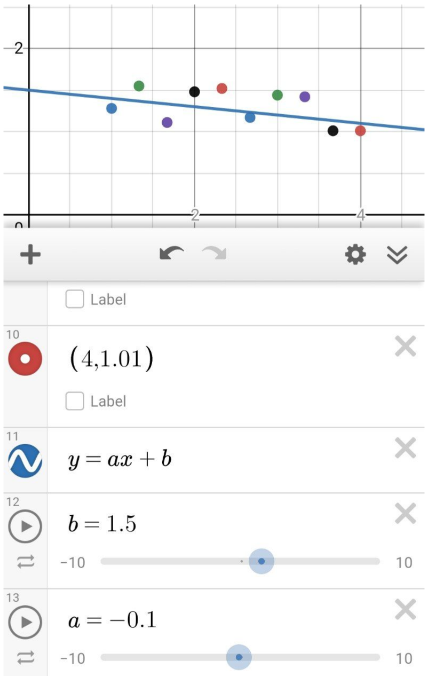
```
x = np.array([1.00,1.33,1.67,2.00,2.33,2.67,3.00,3.33,3.67,4.00])
y = np.array([1.28,1.55,1.11,1.48,1.52,1.17,1.44,1.42,1.01,1.01])
```

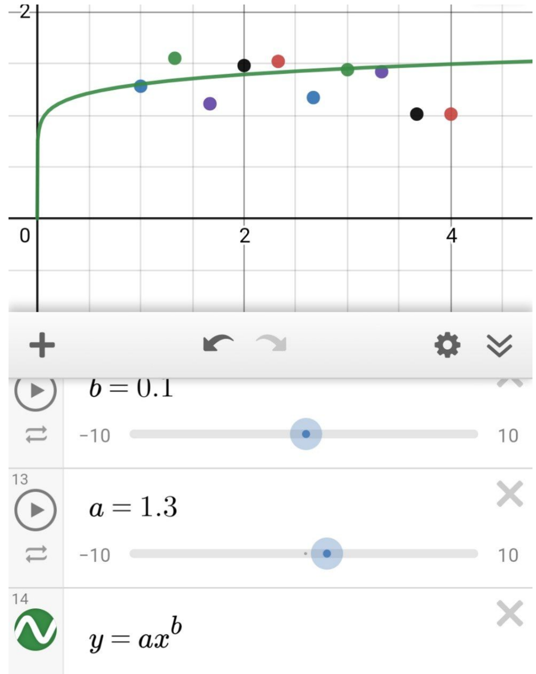
```
fig = plt.subplots()
plt.plot(x,y,marker='o')
plt.show()
```

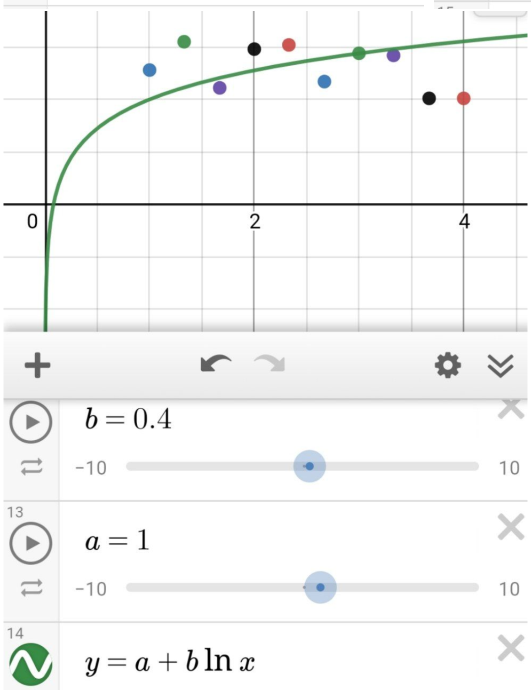


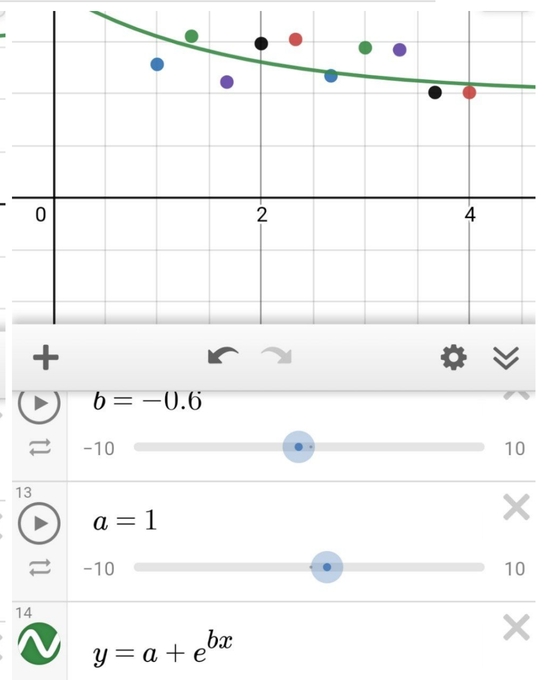
2.2 Обрати клас апроксимуючої функції

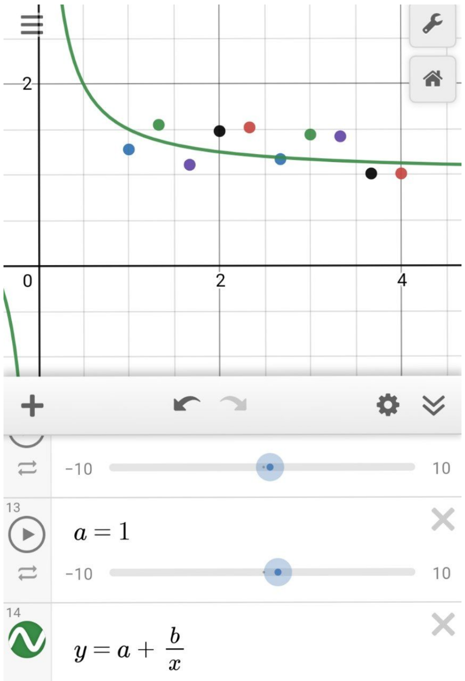
Оберемо клас апроксимуючої функції, для цього приблизно відобразимо графіки навколо точок та подивимося, які функції краще підходять. Тут коефіцієнти будемо обирати методом підбору:


Label
10
(4,1.01)
Label
11
y = ax + b
12
b = 1.5
−10
10
13
a = −0.1
−10
10

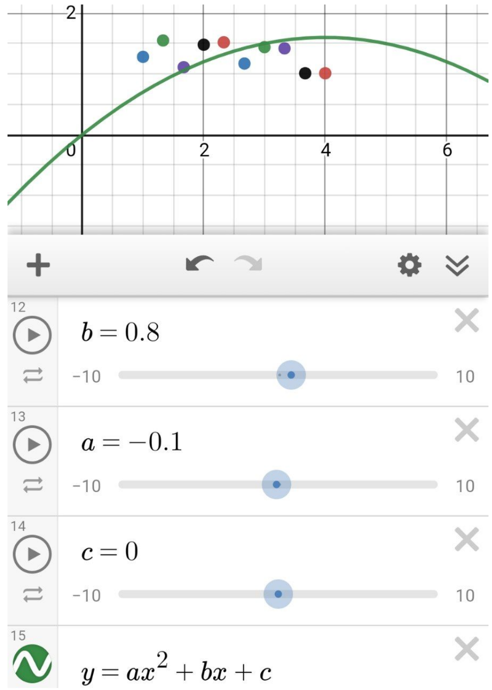

2
0
2
4
b = 0.1
−10
10
13
a = 1.3
−10
10
14
$y = ax^b$


0
2
4
b = 0.4
−10
10
13
a = 1
−10
10
14
y = a + b ln x

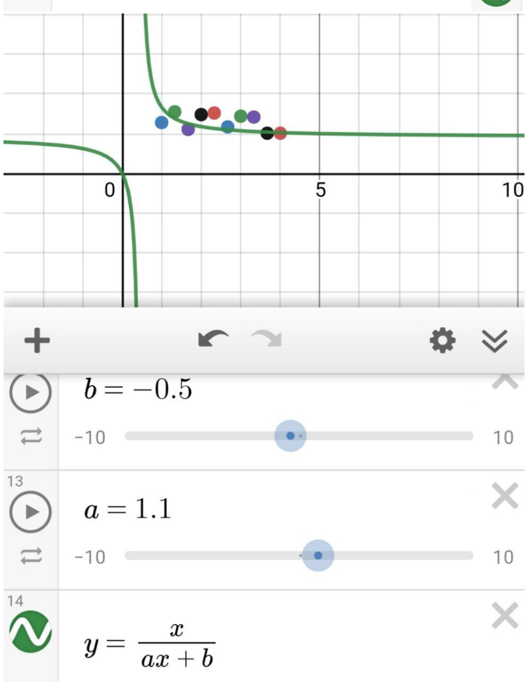

0
2
4
b = −0.6
−10
10
13
a = 1
−10
10
14
$y = a + e^{bx}$



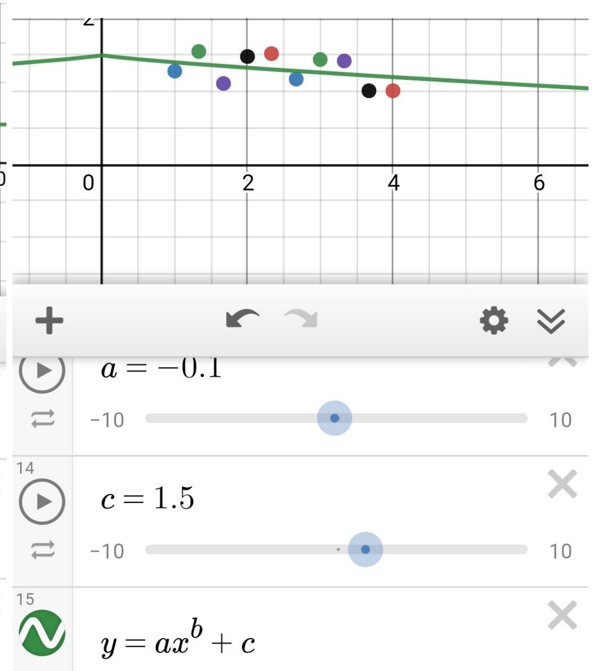
2
0
2
4
−10
10
13
a = 1
−10
10
14
y = a + b/x



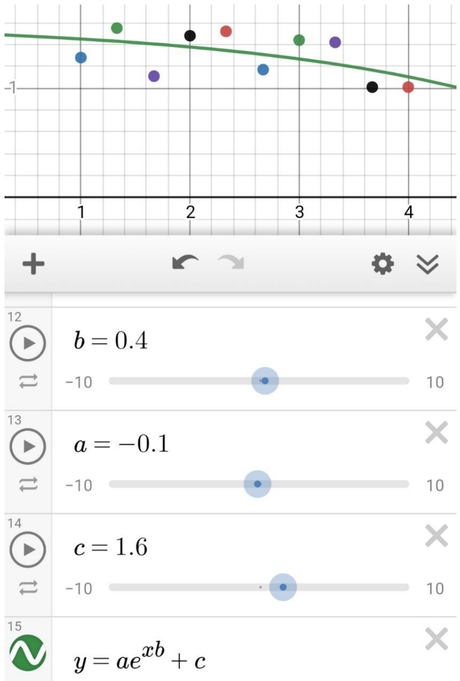
2
0
2
4
6
12
b = 0.8
−10
10
13
a = −0.1
−10
10
14
c = 0
−10
10
15
y = ax^2 + bx + c



0
5
10
b = −0.5
−10
10
13
a = 1.1
−10
10
14
y = x/(ax + b)



0
2
4
6
a = −0.1
−10
10
14
c = 1.5
−10
10
15
y = ax^b + c



Оберемо 2 класи апроксимуючих функцій: $y = ax + b$ та $y = ae^{xb} + c$, знайдемо оптимальні параметри та порівняємо результати.

2.3 Скласти та розв'язати систему рівнянь для обчислення параметрів. Побудувати на одному графіку точки, задані в таблиці та графік отриманої функції

Спочатку знайдемо коефіцієнти для апроксимуючого полінома першого степеня $y = ax + b$ з допомогою МНК:

Використовуючи метод найменших квадратів, підберемо апроксимуючий поліном 1го степеня $ax+b$ для даних

$\varphi(x) = ax+b$

Коефіцієнти a і в знайдемо з умови $\sum_{i=0}^{9}(ax_i+b-y_i)^2 \to \min$

прирівнюємо до нуля частинні похідні по a та в

$$\begin{cases} \sum_{i=0}^{9} 2(ax_i+b-y_i)\cdot x_i = 0 \\ \sum_{i=0}^{9} 2(ax_i+b-y_i) = 0 \end{cases}$$

Підставимо з табл. значення x_i та y_i

$$\begin{cases} (ax_0+b-y_0)\cdot x_0+(ax_1+b-y_1)\cdot x_1+(ax_2+b-y_2)x_2+(ax_3+b-y_3)x_3+ \\ +(ax_4+b-y_4)\cdot x_4+(ax_5+b-y_5)x_5+(ax_6+b-y_6)x_6+(ax_7+b-y_7)x_7+ \\ +(ax_8+b-y_8)\cdot x_8+(ax_9+b-y_9)x_9=0 \\ \\ (ax_0+b-y_0)+(ax_1+b-y_1)+(ax_2+b-y_2)+(ax_3+b-y_3)+(ax_4+b-y_4)+ \\ +(ax_5+b-y_5)+(ax_6+b-y_6)+(ax_7+b-y_7)+(ax_8+b-y_8)+(ax_9+b-y_9)=0 \end{cases}$$

Розв'яжемо систему:

$$\begin{cases} (a+b-1{,}28)+(1{,}33a+b-1{,}55)\cdot 1{,}33+(1{,}67a+b-1{,}11)\cdot 1{,}67+(2a+b-1{,}48)\cdot 2+ \\ +(2{,}33a+b-1{,}52)\cdot 2{,}33+(2{,}67a+b-1{,}17)\cdot 2{,}67+(3a+b-1{,}44)\cdot 3+(3{,}33a+b-1{,}42)\cdot 3{,}33+ \\ +(3{,}67a+b-1{,}01)\cdot 3{,}67+(4a-1{,}01+b)\cdot 4=0 \\ \\ (a+b-1{,}28)+(1{,}33a+b-1{,}55)+(1{,}67a+b-1{,}11)+(2a+b-1{,}48)+ \\ +(2{,}33a+b-1{,}52)+(2{,}67a+b-1{,}17)+(3a+b-1{,}44)+(3{,}33a+b-1{,}42)+ \\ +(3{,}67a+b-1{,}01)+(4a-1{,}01+b) \end{cases}$$

$$\begin{cases} 71{,}6734a+25b-31{,}616=0 \\ 25a+10b-12{,}99=0 \end{cases}$$

Маємо: $\begin{cases} a^* = 0{,}01537053 \\ b^* = 1{,}26057368 \end{cases}$

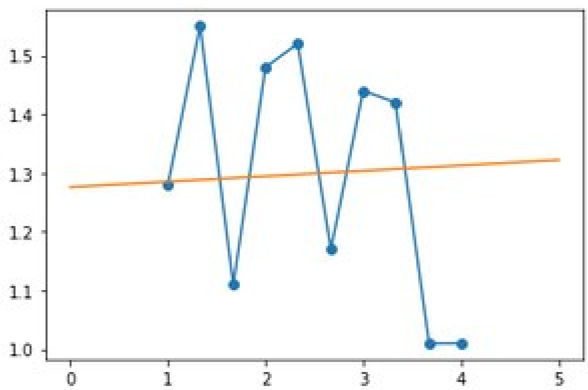
Знайдемо коефіцієнти апроксимуючого многочлену

```
A = np.array([[71.6734,25], [25,10]])
b = np.array([32.616, 12.99])
syst_x = np.linalg.solve(A, b)
syst_x
```

```
array([0.01537053, 1.26057368])
```

```
lin_regr = lambda x: syst_x[0]*x+syst_x[1]
```

```
fig = plt.subplots()
plt.plot(x,y,marker='o')
xi = np.linspace(0, 5, num=10)
plt.plot(xi, lin_regr(x),color='C1', label='y=ax+b')
plt.show()
```



Тепер візьмемо експоненціальну функцію $y = ae^{xb} + c$ за апроксимуючу та знайдемо оптимальні параметри з допомогою МНК:

Нехай для апроксимуючу ф-ції $f(x)$ вибрано експоненціальну

$\phi(x,a,b) = a \cdot e^{bx} + C$

Частинні похідні $\frac{\partial \phi}{\partial a} = e^{bx}$; $\frac{\partial \phi}{\partial b} = ab + e^{bx}$ $\Rightarrow$ сист. буде нелінійною

Прологарифмуємо рівність: $ae^{bx} = y$: $\ln a + bx = \ln y$

Уведемо позначення $A = \ln a$.

$y_i = f(x_i)$, $i = \overline{0,9}$ см. таблицю значень

X	1	1,33	1,67	2	2,33	2,67	3	3,33	3,67	4
Y	0,2468	0,4383	0,1044	0,392	0,4187	0,157	0,3646	0,3506	~~[illegible]~~ 0,00995	~~[illegible]~~ 0,00995

Методом найм квадратів з-мо оптимальні параметри

A^* і b^* для ф-ції $Y = A + bx$

Після цього обчислимо значення $a^* = e^{A^*}$ й результат $\boxed{f(x) \approx a^* e^{b^* x}}$

$$\begin{cases} (A+b-0,2468)+(1,33A+b-0,4383)\cdot 1,33+(1,67A+b-0,1044)\cdot 1,67+ \\ +(2A+b-0,392)\cdot 2+(2,33A+b-0,4187)\cdot 2,33+(2,67A+b-0,157)\cdot 2,67+ \\ +(3A+b-0,3646)\cdot 3+(3,33A+b-0,3506)\cdot 3,33+(3,67A+b-0,00995)\cdot 3,67+ \\ +(4A+b-0,00995)\cdot 4 \\ (A+b-0,2468)+(1,33A+b-0,4383)+(1,67A+b-0,1044)+(2A+b-0,392)+ \\ +(2,33A+b-0,4187)+(2,67A+b-0,157)+(3A+b-0,3646)+(3,33A+b-0,3506)+ \\ +(3,67A+b-0,00995)+(4A+b-0,00995) \end{cases}$$

$$\begin{cases} 71,6734A + 25b = 5,5204625 \\ 25A + 10b = 2,4923 \end{cases} \Rightarrow \begin{cases} A^* = -0,07742903 \\ b^* = 0,44280258 \\ C^* = 1,6 \end{cases}$$

Нехай за апроксимуючу функцію вибрано експоненціальну: $y = a \cdot e^{bx} + c$

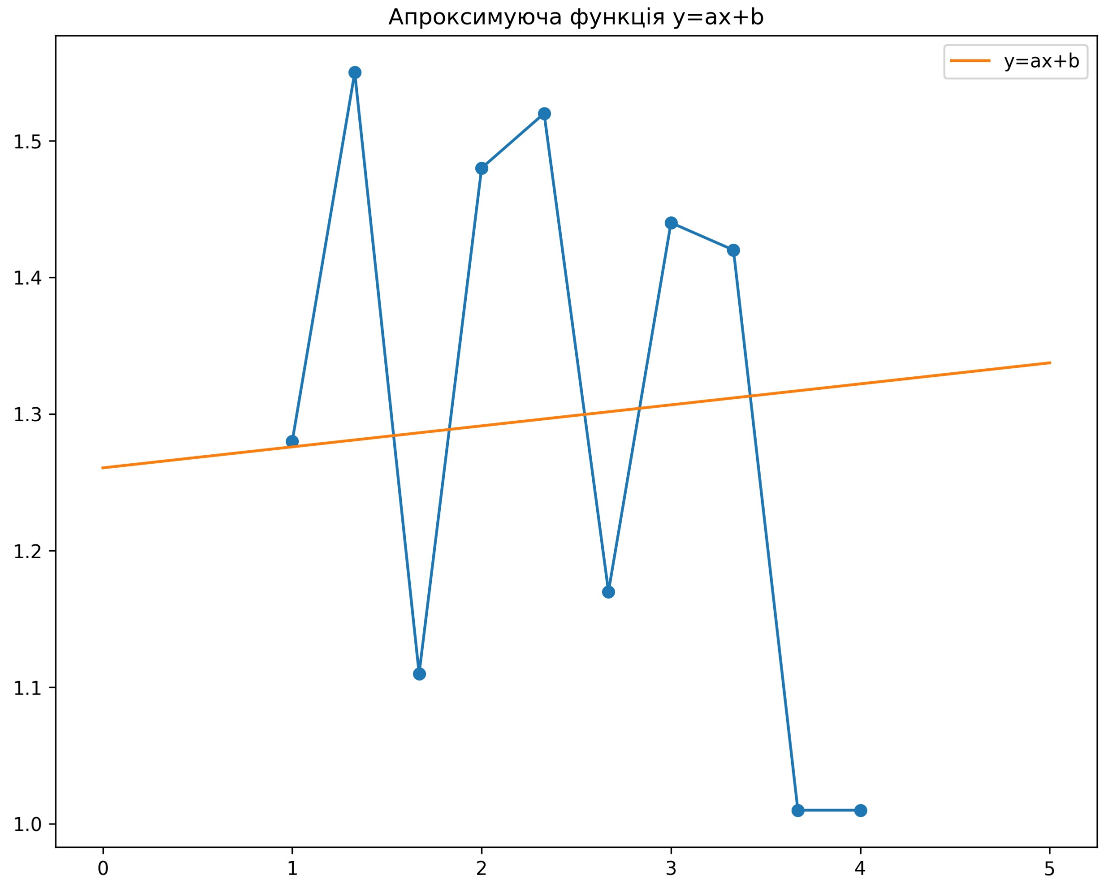
```
d = {'x':x,'y':y, 'ln(y)': np.log(y)}
df = pd.DataFrame(d, columns=['x','y', 'ln(y)'])
df
```

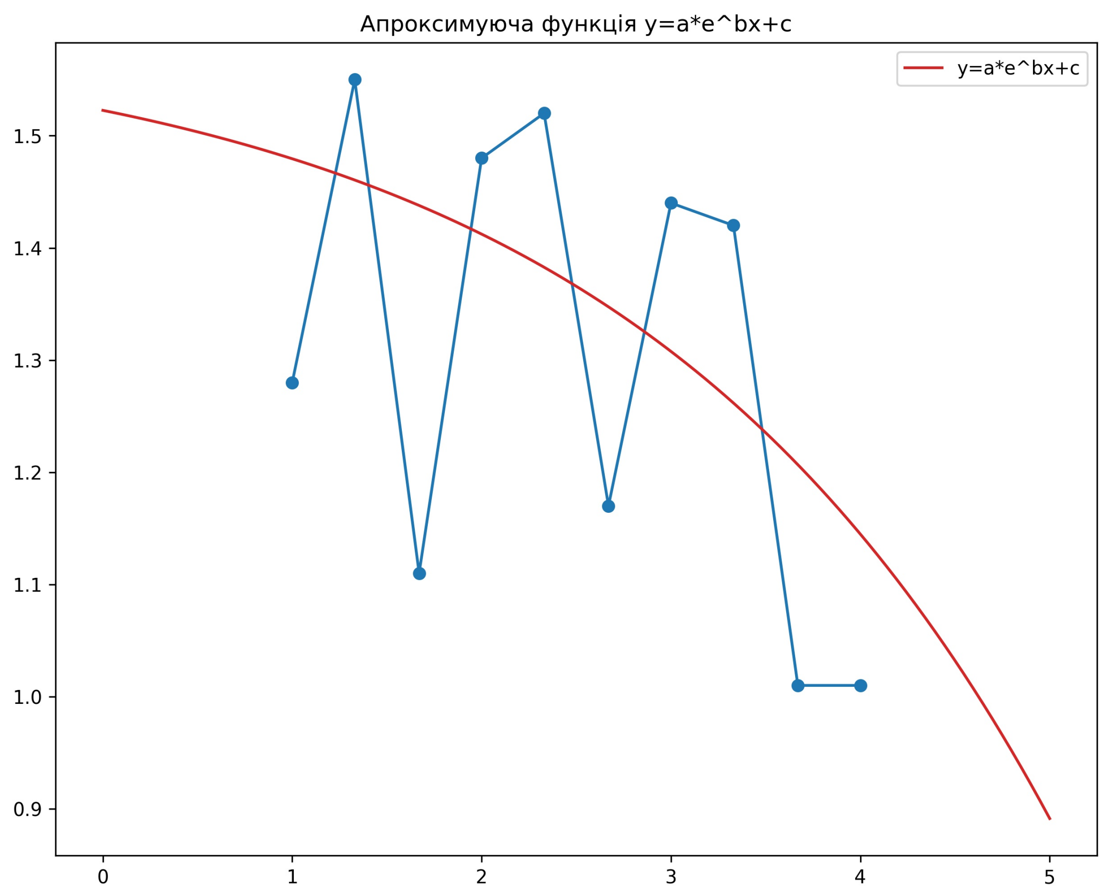
	x	y	ln(y)
0	1.00	1.28	0.246860
1	1.33	1.55	0.438255
2	1.67	1.11	0.104360
3	2.00	1.48	0.392042
4	2.33	1.52	0.418710
5	2.67	1.17	0.157004
6	3.00	1.44	0.364643
7	3.33	1.42	0.350657
8	3.67	1.01	0.009950
9	4.00	1.01	0.009950

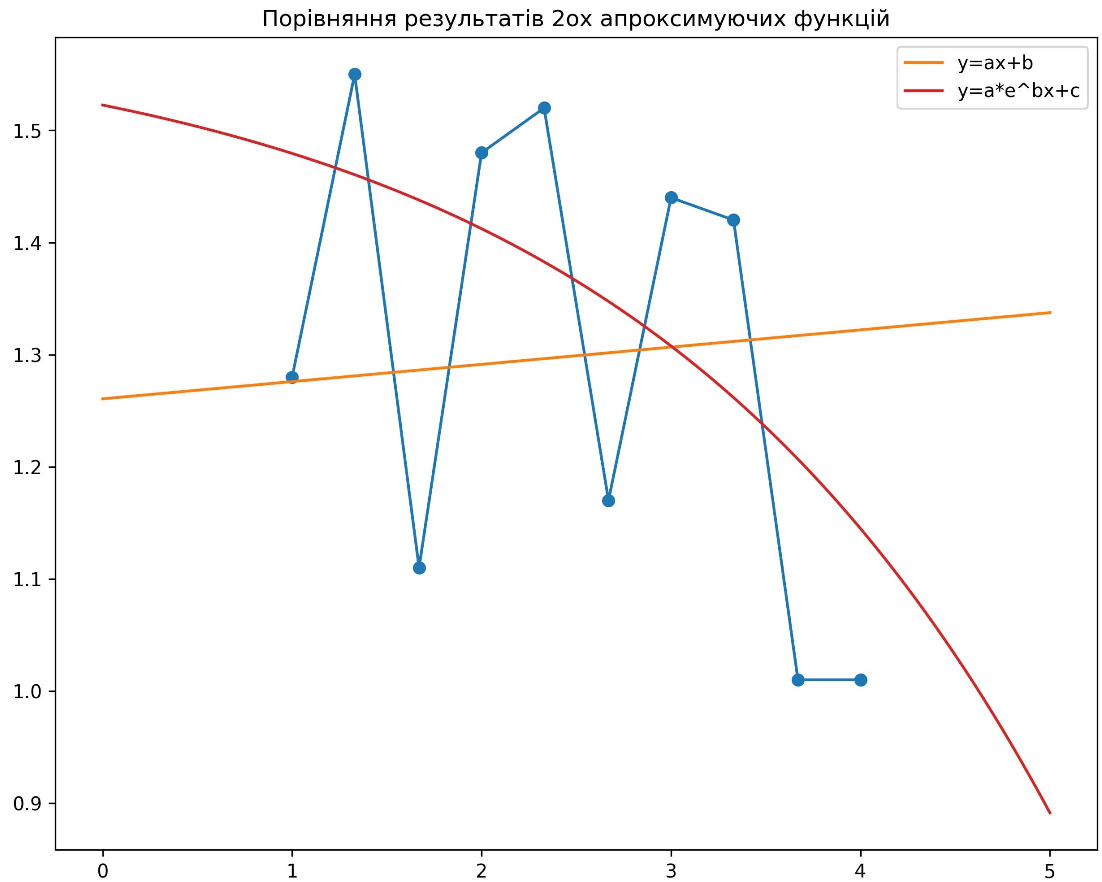
```
A2 = np.array([[71.6734,25], [25,10]])
b2 = np.array([5.5204625, 2.4923])
syst_x2 = np.linalg.solve(A2, b2)
syst_x2
```

```
array([-0.07742903,  0.44280258])
```

Порівняємо отримані апроксимуючі функції з допомогою графіків:




Апроксимуюча функція y=a*e^bx+c
y=a*e^bx+c
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0
1
2
3
4
5



2.4 Зробити висновок, проаналізувавши отримані графіки

Як бачимо з графіків, апроксимуюча функція $y = ae^{xb} + c$ краще наближує нашу таблично задану функцію.