

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА

Факультет комп'ютерних наук та кібернетики

ЛАБОРАТОРНА РОБОТА №3

«Інтерполяція. Сплайни»

Варіант 22

Виконала:

студентка 2 курсу, групи К-24

спеціальності «Комп'ютерні науки.
Інформатика»

Баклан Аліса

Завдання:

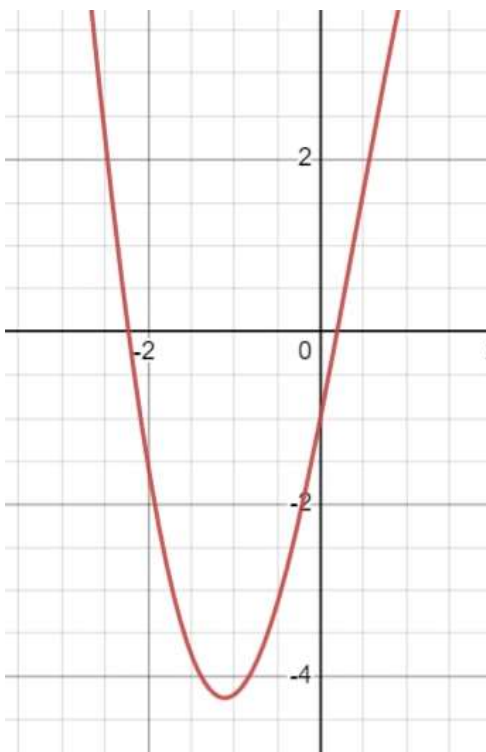
Ця робота складається з двох завдань:

Інтерполяція.

22. Знайти найменший по модулю від'ємний корінь нелінійного рівняння $x^2 + 5 \sin x - 1 = 0$ за допомогою інтерполяції (використати інтерполяційний поліноми Лагранжа та Ньютона, побудованого за 10 вузлами, які є нулями поліному Чебишова).

Сплайни. За своїми вузлами в першій частині завдання побудувати природний кубічний інтерполяційний сплайн. Вивести графік із функцією, яка інтерполюється, вивести інтерполяційні поліноми (з першої частини завдання) та вивести кубічний сплайн.

Аналіз кореня



Ми шукаємо найменший по модулю від'ємний корінь. На графіку він знаходиться на проміжку $[-3; -2]$.

Перевіримо, чи дійсно це так:

$$F(-3) = 9 + 5 \sin(-3) - 1 = 7.2943 > 0;$$

$$F(-2) = 4 + 5 \sin(-2) - 1 = -1.546 < 0;$$

$$F(-2) \cdot F(-3) < 0;$$

Маємо корінь на $[-3; -2]$.

Використана теорія

Поліном Чебишова:

Для побудови інтерполяційного поліному n -го степеня P_n необхідно в якості вузлів взяти $(n + 1)$ корінь поліному Чебишова $(n + 1)$ -го степеня:

$$x_k = \frac{a+b}{2} + \frac{b-a}{2} \cos \frac{(2k+1)\pi}{2(n+1)}, \quad k = \overline{0, n}.$$

При використанні цих вузлів похибка має вигляд:

$$|f(x) - L_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \frac{(b-a)^{n+1}}{2^{2n+1}}.$$

Інтерполяційний поліном Лагранжа:

Інтерполяційний поліном Лагранжа

Формула для побудови поліному у формі Лагранжа:

$$L_n(x) = \sum_{k=0}^n \frac{\prod_{\substack{j=0 \\ j \neq k}}^n (x - x_j)}{\prod_{\substack{j=0 \\ j \neq k}}^n (x_k - x_j)} f(x_k),$$

для зручності її можна переписати в іншому вигляді:

$$L_n(x) = \sum_{k=0}^n \frac{\omega(x)}{(x - x_k)\omega'(x_k)} f(x_k), \quad (25)$$

де $\omega(x) = (x - x_0)(x - x_1)\dots(x - x_n)$.

Інтерполяційний поліном Ньютона:

Інтерполяційний поліном Ньютона

Розділеною різницею першого порядку називається величина:

$$f(x_i, x_j) = \frac{f(x_j) - f(x_i)}{x_j - x_i};$$

другого порядку:

$$f(x_{i-1}, x_i, x_{i+1}) = \frac{f(x_i, x_{i+1}) - f(x_{i-1}, x_i)}{x_{i+1} - x_{i-1}};$$

$(k+1)$ порядку:

$$f(x_i, \dots, x_{i+k+1}) = \frac{f(x_{i+1}, \dots, x_{i+k+1}) - f(x_i, \dots, x_{i+k})}{x_{i+k+1} - x_i}.$$

На підставі цієї таблиці, використовувачи перший її рядок, можемо записати *інтерполянт Ньютона вперед*:

$$P_n(x) = f(x_0) + f(x_0, x_1)(x - x_0) + f(x_0, x_1, x_2)(x - x_0)(x - x_1) + \dots + f(x_0, x_1, \dots, x_n)(x - x_0)(x - x_1) \cdot \dots \cdot (x - x_{n-1}),$$

Сплайни:

Інтерполяційний природний кубічний сплайн

Інтерполяційним природним кубічним сплайном називається поліном, для якого виконуються умови:

- 1) $s(x)$ – поліном степеня 3 для $x \in [x_{i-1}, x_i]$, $i = \overline{1, n}$;
- 2) $s(x) \in C_{[a; b]}^2$;
- 3) $s(x_i) = f(x_i)$, $i = \overline{0, n}$;

4) $s''(a) = s''(b) = 0$ – умова природності.

Зауваження. Для побудови інтерполяційного кубічного сплайну можна замість умови 4) використовувати інші умови, але тоді сплайн не буде природним: $s''(a) = A$; $s''(b) = B$ або $s'(a) = A$; $s'(b) = B$, або умови періодичності: $s(a) = s(b)$, $s'(a) = s'(b)$, $s''(a) = s''(b)$.

Розглянемо формули для побудови інтерполяційного природного кубічного сплайну s_i на проміжку $[x_{i-1}, x_i]$:

$$s_i = a_i + b_i(x - x_i) + \frac{c_i}{2}(x - x_i)^2 + \frac{d_i}{6}(x - x_i)^3,$$

де c_i знаходяться з тридіагональної системи лінійних алгебраїчних рівнянь:

$$h_i c_{i-1} + 2c_i(h_i + h_{i+1}) + h_{i+1} c_{i+1} = 6 \left(\frac{f_{i+1} - f_i}{h_{i+1}} - \frac{f_i - f_{i-1}}{h_i} \right),$$

$$c_0 = c_n = 0;$$

решта коефіцієнтів знаходяться за формулами:

$$a_i = f_i; \quad b_i = \frac{h_i}{2}c_i - \frac{h_i^2}{6}d_i + \frac{f_i - f_{i-1}}{h_i}; \quad d_i = \frac{c_i - c_{i-1}}{h_i}.$$

Реалізація в програмі

Створюємо функцію що описує наш приклад, а також функцію, яка знаходить нулі поліному Чебишова:

```
float f(float x) {  
    float y = x * x + 5 * sin(x) - 1;  
    return y;  
}  
  
void Cheb(float x[10], float a, float b)  
{  
    for (int i = 0; i < 10; i++)  
    {  
        x[i] = ((a + b) / 2) + ((b - a) / 2) * cos((2 * i + 3) * pi / (2 * 10));  
    }  
}
```

Створюємо функції для знаходження поліномів Лагранжа і Ньютона.

```
float Lagrange(float x, int n, float x_arr[], float y_arr[]) {  
    float sum = 0;  
    for (int i = 0; i < n; ++i) {  
        float l = 1;  
        for (int j = 0; j < n; ++j)  
            if (j != i)  
                l *= (x - x_arr[j]) / (x_arr[i] - x_arr[j]);  
        sum += y_arr[i] * l;  
    }  
    return sum;  
}
```



```

float Newton(float x, int n, float x_arr[], float y_arr[]) {
    float sum = y_arr[0];
    for (int i = 1; i < n; ++i) {
        float F = 0;
        for (int j = 0; j <= i; ++j) {
            float den = 1;
            for (int k = 0; k <= i; ++k)
                if (k != j)
                    den *= (x_arr[j] - x_arr[k]);

            F += y_arr[j] / den;
        }
        for (int k = 0; k < i; ++k)
            F *= (x - x_arr[k]);
        sum += F;
    }
    return sum;
}

```

Ініціалізуємо змінні, знайдемо нулі Чебишова, значення функції у цих точках:

```

float x_arr[10];
float y_arr[10];
int N = 10;

float a = -3;
float b = -2;

Cheb(x_arr, a, b);

for (int i = 0; i < 10; i++)
{
    y_arr[i] = f(x_arr[i]);
}

```

Організуємо виведення у вигляді таблиць:

```

for (int i=0; i<10; i++) {
    cout << setw(3) << fixed << setprecision(8) << x_arr[i];
    cout << setw(12) << fixed << setprecision(6) << f(x_arr[i]);
    cout << setw(12) << fixed << setprecision(6) << Lagrange(x_arr[i], N, x_arr, y_arr);
    cout << setw(12) << fixed << setprecision(6) << Newton(x_arr[i], N, x_arr, y_arr);
    cout << endl;
}
cout << "======" << endl;

```

Запустимо програму:

x	f(x)	Lagrange(x)	Newton(x)
-2.05449677	-1.205443	-1.205443	-1.205443
-2.14644670	-0.586958	-0.586958	-0.586958
-2.27300477	0.349463	0.349463	0.349463
-2.42178273	1.568823	1.568823	1.568823
-2.57821727	2.976990	2.976990	2.976990
-2.72699523	4.422396	4.422396	4.422394
-2.85355330	5.722402	5.722402	5.722398
-2.94550323	6.701813	6.701813	6.701814
-2.99384403	7.227044	7.227044	7.227053
-2.99384427	7.227046	7.227046	7.227055
=====			

Використавши ці дані за допомогою онлайн калькулятора можемо знайти x :

$X = -2.227$

Перейдемо до реалізації сплайнів:

```

void Spline(float x[N + 1], float y[N + 1], float A[N], float B[N], float C[N], float D[N])
{
    float h[N];           // delta x
    float b[N];           // delta y / delta x
    float ftt[N + 1];     // f''(xi)

    // calc h & b
    for (int i = 0; i < N; i++)
    {
        h[i] = (x[i + 1] - x[i]);
        b[i] = (y[i + 1] - y[i]) / h[i];
    }

    // calc f''(xi)
    ftt[0] = 0; // f''(x0) = 0
    for (int i = 0; i < N - 1; i++)
        ftt[i + 1] = 3 * (b[i + 1] - b[i]) / (h[i + 1] + h[i]);
    ftt[N] = 0; // f''(xn) = 0

    // calc spline coefficients
    for (int i = 0; i < N; i++)
    {
        A[i] = (ftt[i + 1] - ftt[i]) / (6 * h[i]);
        B[i] = ftt[i] / 2;
        C[i] = b[i] - h[i] * (ftt[i + 1] + 2 * ftt[i]) / 6;
        D[i] = y[i];
    }
}

```

Функція виведення:

```

// receives points and Spline coefficients and print the functions
void PrintSpline(float x[N + 1], float A[N], float B[N], float C[N], float D[N])
{
    cout << " f(x) = d(x - xi)^3 + c(x - xi)^2 + d (x - xi) + a" << endl;
    cout << setw(12) << " a ";
    cout << setw(12) << " b ";
    cout << setw(12) << " c ";
    cout << setw(12) << " d ";
    cout << endl;
    for (int i = 0; i < N; i++)
    {
        cout << setw(12) << fixed << setprecision(5) << A[i];
        cout << setw(12) << fixed << setprecision(5) << B[i];
        cout << setw(12) << fixed << setprecision(5) << C[i];
        cout << setw(12) << fixed << setprecision(5) << D[i];
        cout << endl;
    }
}

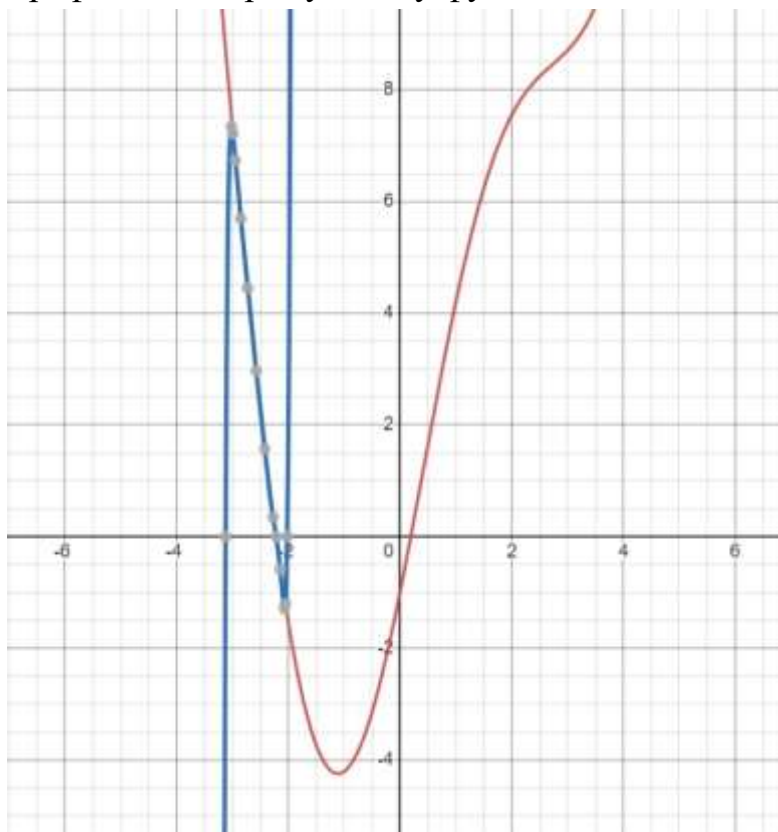
```

Результат:

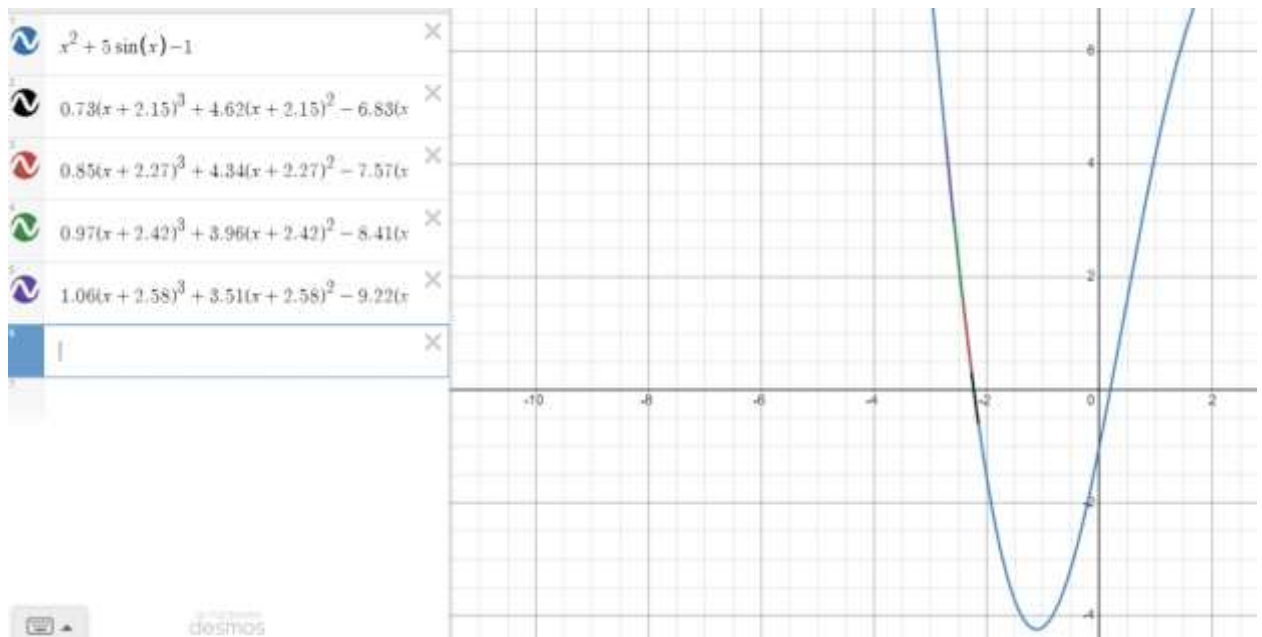

```
=====
f(x) = d(x - xi)^3 + c(x - xi)^2 + d (x - xi) + a
      a          b          c          d
-16.74359    0.00000    -6.58476    -1.20544
 0.73320     4.61872    -6.82635    -0.58696
 0.85178     4.34034    -7.56895     0.34946
 0.96595     3.96016    -8.40577     1.56882
 1.06048     3.50684    -9.21692     2.97699
 1.12705     3.03351    -9.90615     4.42240
 1.16671     2.60560   -10.42185     5.72240
 628.78967    2.28376   -12.22414     6.70181
-124297952.00000 -88.90482    -8.00001     7.22704
=====
```

Графіки

Графік, що зображує нашу функцію і поліном Ньютона:



Зверніть увагу, що поліном і функція дуже щільно розташовані один до одного саме на обраному інтервалі $[-2; -3]$. Наш метод працює! А так виглядають наші сплайни:



Висновок

У цій лабораторній роботі було досліджено інтерполяційні поліноми Ньютона і Лагранжа, а також природній кубічний сплайн. Всі методи справно працюють. У цій роботі я зверталася до багатьох онлайн-ресурсів, зокрема Desmos для побудови графіків і онлайн-калькуляторів для побудови інтерполяційних поліномів (для перевірки значень власних значень на правильність). Всі методи у лабораторній роботі були корисними для реалізування. Наявність графіків допомагає візуалізувати інтерполяцію.