Київський національний університет імені Тараса Шевченка
Факультет комп’ютерних наук та кібернетики

Лабораторна робота №4

Чисельні методи в інформатиці

“Розв’язок систем рівнянь за допомогою
прямих та ітераційних методів”

Варіант №4

Виконав студент групи ІПС-31
Міцкевич Костянтин Олександрович

Київ — 2025

Постановка задачі

1. Побудувати інтерполяційний поліном Лагранжа за п’ятьма вузлами для функції $x^6 + 2 * x^5 + 3 * x^4 - 3$ на проміжку [3..7]. Вузли обрати як нулі полінома Чебишова того ж порядку для відповідного проміжку. Оцінити похибку інтерполяції.
2. Знайти деякий розв’язок рівняння з попереднього пункту, використовуючи пряму та обернену інтерполяцію. Порядок полінома взяти той же що і у попередньому пункті. Якщо у дослідженому інтервалі є розв’язок – для прямої інтерполяції можна використати вже побудований поліном.

Теоретичні відомості та обгрунтування

<u>Інтерполяційний поліном Лагранжа:</u>

Формула для побудови поліному у формі Лагранжа:

$L_n(x) = \sum_{k=0}^{n} \frac{\prod_{j=0, j\neq k}^{n} x - x_j}{\prod_{j=0, j\neq k}^{n} x_k - x_j} f(x_k)$ для зручності її можна переписати в іншому вигляді:

$L_n(x) = \sum_{k=0}^{n} \frac{\omega(x)}{(x - x_k)\omega'(x_k)} f(x_k)$, де $\omega(x) = (x - x_0)(x - x_1)...(x - x_n)$

<u>Похибка інтерполяції.</u>

Для оцінки похибки інтерполяції можна використати оцінку залишкового члену у формі Лагранжа:

$|f(x) - P_n(x)| \leq \frac{M_{n+1}}{(n+1)!} |\omega(x)|$, де $M_{n+1} = \max_{x \epsilon [a,b]} |f^{(n+1)}(x)|$

<u>Оптимальний вибір вузлів:</u>

Для зменшення похибки інтерполяції необхідно в якості вузлів взяти нулі поліному Чебишова 1 роду. Для їх визначення використовують рекурентні співвідношення:

$T_{n+1}(x) - 2xT_n(x) + T_{n-1}(x) = 0$

$T_0(x) = 0$, $T_1(x) = x$

чи в явному вигляді:

$T_n(x)=\frac{(x+\sqrt{x^2-1})^n+(x-\sqrt{x^2-1})^n}{2}$, |x| >= 1

$T_n(x) = \cos(n*\arccos(x))$, x <= 1.

Позначимо: $\overline{T_n}(x)=2^{1-n}T_n(x),\ |x|\leq 1$

Лема 1. Серед усіх поліномів $P_n(x)$ степеня n із старшим коефіцієнтом, що дорівнює 1 багаточлен $\overline{T_n}(x)$ найменш відхиляється від нуля на проміжку [−1, 1] та $\|P_n(x)\|_{C[-1,1]}\geq\|\overline{T_n}(x)\|_{C[-1,1]}=2^{1-n}$

Лема 2. Система багаточленів Чебишова $T_n(x)$ є ортогональною на проміжку [−1; 1] з ваговою функцією $\rho(x)=\frac{1}{\sqrt{1-x^2}}$

$$(T_k(x),T_m(x))_{L_{2,\rho}[a,b]}=\int_{-1}^{1}\frac{1}{\sqrt{1-x^2}}T_k(x)T_m(x)dx=\begin{matrix}=0,\ якщо\ k\neq m\\ \neq 0,\ якщо\ k=m\end{matrix}$$

Нулі полінома Чебишова x ∈ [−1, 1]: $x_k=\cos\frac{(2k+1)\pi}{n},\quad k=\overline{0,n-1}$

Екстремуми: $x_k=\cos\frac{\pi k}{n},\quad k=\overline{0,n}$

<u>Поліноми Чебишова 1 роду на проміжку [a;b]:</u>

За допомогою заміни $x=\frac{1}{2}((b-a)z+(b+a))$ переведемо проміжок [−1, 1] в [a, b]. Тоді поліноми Чебишова запишуться таким чином:

$$T_n^{[a;b]}(x)=T_n^{[-1;1]}\left(\frac{2x-(b+a)}{b-a}\right)$$

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

Відповідно поліном із старшим коефіцієнтом 1 набуває вигляду:

$$\overline{T_n}^{[a;b]}(x)=(b-a)^n*2^{1-2n}T_n^{[-1;1]}\left(\frac{2x-(b+a)}{b-a}\right),$$

а його відхилення від нуля подамо у вигляді: $\|\overline{T_n}^{[a;b]}\|_{C[a,b]}=(b-a)^n*2^{1-2n}$

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:

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

Пряма інтерполяція:

Нехай функція y = f(x) ∈ C[a, b], що задана таблично (x_i, y_i), $i=\overline{0,n}$, немонотонна. Для знаходження x^* застосовуємо такий алгоритм:

За заданою таблицею (x_i , y_i), i = 0, n будуємо інтерполяційний поліном $L_n=\sum_{i=0}^{n} f(x_i)\frac{w_{n+1}(x)}{(x-x_i)w'_{n+1}(x_i)}$ та розв’язують нелінійне рівняння

$L_n = y^*$

На підставі теореми про середнє: $f(x) - f(x^*) = f_0(\xi)(x - x^*)$ одержимо

$$x-x^*=\frac{f(x)-f(x^*)}{f'(\xi)}$$

Отже, похибка інтерполяції має вигляд: $|x-x^*|\le\frac{M_{n+1}|\omega(x^*)|}{m_1*(n+1)!}$, де

$$m_1=\min_x|f'(x)|,\quad M_2=\max_x|f^{n+1}(x)|$$

Обернена інтерполяція:

Нехай функція y = f(x) ∈ C[a, b], що задана таблично (x_i, y_i), $i=\overline{0,n}$, монотонна. Для знаходження x^* застосовуємо такий алгоритм:

Будуємо за таблицею (x_i, y_i), i = 0, n таку таблицю:

(y_i, x_i), $i=\overline{0,n}$. На підставі останньої таблиці інтерполянт набуває вигляду: $L_n(y)=\sum_{i=0}^{n} x_i\frac{w_{n+1}(y)}{(y-y_i)\omega'_{n+1}(y)}$,

$\omega_{n+1}(y) = (y - y_0)(y - y_1)\dots(y - y_n)$ та $L(y^*) \approx x^*$. Залишковий член в цьому випадку утворюється із залишкового члена формули Ньютона, якщо в останньому поміняти місцями x та y, а похідну $f_0(x)$ замінити на похідну від оберненої функції. Похибка інтерполяції має вигляд:

$$|x-x^*| \le \frac{\widetilde{M}_{n+1}}{(n+1)!}|\omega(y^*)|; \quad \widetilde{M}_{n+1} = \max_y \left| \frac{d^{n+1}}{dy^{n+1}} x(y) \right|$$

Зауваження. За допомогою інтерполяції можна знаходити корені нелінійних рівнянь, для цього знаходять x^* при $y^* = 0$.

Хід роботи

Виконання лабораторної розпочинаємо з пошуку вузлів Чебешева на інтервалі [3, 7] та відсортуємо їх

```
=== КРОК 1: Обчислення вузлів Чебишева ===
k=0: t_k=0.951057,  x_k=6.902113
k=1: t_k=0.587785,  x_k=6.175571
k=2: t_k=0.000000,  x_k=5.000000
k=3: t_k=-0.587785,  x_k=3.824429
k=4: t_k=-0.951057,  x_k=3.097887

Відсортовані вузли: [3.09788697 3.8244295  5.         6.1755705  6.90211303]
```

Наступним етапом шукаємо значення функціх f(x) у знайдених вузлах:

```
=== КРОК 2: Значення функції у вузлах ===
     x_k        f(x_k)
3.097887   1727.816129
3.824429   5404.043473
5.000000  23747.000000
6.175571  77795.511873
6.902113 146250.628525
```

Починаємо будувати інтерполяційний поліном за допомогою $L_n(x)=\sum_{k=0}^{n}\frac{\prod_{j=0,j\neq k}^{n} x-x_j}{\prod_{j=0,j\neq k}^{n} x_k-x_j} f(x_k)$. Програма обчислює L_k, а потім з’єднує їх в кінцевий поліном:

```
=== КРОК 3: Побудова інтерполяційного полінома Лагранжа ===

--- Обчислення L_0(x) ---
 L_0(x) = (0.262865556059567*x - 1.81432778029783)*(0.324919696232906*x - 2.00656449241464)*(0.525731112
119134*x - 2.62865556059567)*(1.37638192047117*x - 5.26387561360597)

--- Обчислення L_1(x) ---
 L_1(x) = -0.161803398874989*x**4 + 3.42627928075882*x**3 - 26.5382691835089*x**2 + 88.6252542554367*x -
 106.827327487445

--- Обчислення L_2(x) ---
 L_2(x) = (0.525731112119134*x - 3.62865556059567)*(0.525731112119134*x - 1.62865556059567)*(0.850650808
35204*x - 5.2532540417602)*(0.85065080835204*x - 3.2532540417602)

--- Обчислення L_3(x) ---
 L_3(x) = -0.161803398874989*x**4 + 3.04585667424076*x**3 - 20.831930085738*x**2 + 61.4699406870533*x -
66.1564112750433

--- Обчислення L_4(x) ---
 L_4(x) = (0.262865556059567*x - 0.814327780297834)*(0.324919696232906*x - 1.24263246991443)*(0.52573111
2119134*x - 2.62865556059567)*(1.37638192047117*x - 8.49994359110577)

=== Кінцевий поліном Лагранжа ===
P(x) =  433.0*x**4 - 5439.99999999997*x**3 + 28970.0*x**2 - 71860.0*x + 68171.9999999991
```

За допомогою бібліотек ми спрости вираз і привели його до нормального вигляду. Тепер знайдемо похибку інтерполяцій:

```
=== КРОК 4: Оцінка похибки інтерполяції ===
M_5 = 5280.000000
Оцінка максимальної похибки на [3,7]: 88.000000
```

Перейдемо до прямої та оберненої інтерполяції. Але перед цим глянемо чи є в проміжку [3, 7] якийсь корінь. Якщо його не виявиться змінимо інтервал так, щоб в ньому опинився лише один корінь.

```
=== КРОК 5: Пошук кореня функції f(x) ===
Символьні корені (усі): [CRootOf(x**6 + 2*x**5 + 3*x**4 - 3, 0), CRootOf(x**6 + 2*x**5 + 3*x**4 - 3, 1),
 CRootOf(x**6 + 2*x**5 + 3*x**4 - 3, 2), CRootOf(x**6 + 2*x**5 + 3*x**4 - 3, 3), CRootOf(x**6 + 2*x**5 +
 3*x**4 - 3, 4), CRootOf(x**6 + 2*x**5 + 3*x**4 - 3, 5)]
Знайдені дійсні корені функції f(x):
  x = -1.105157460337
  x = 0.860849560196
```

Як бачимо жоден з коренів не належить проміжку, тому змінимо його та оберемо корінь який шукатимемо:

```
Обраний корінь для побудови локального інтервалу: x = -1.105157460337
Пояснення: вибираємо перший дійсний корінь зі списку знайдених. Якщо потрібно інший корінь — змініть інд
екс у коді.
Інтервал навколо кореня: [-1.605157460337, -0.605157460337]
```

Тепер повторимо процес побудови інтерполяційного полінома Лангранжа, але вже так, щоб можна було застосувати пряму інтерполяцію.

```
=== КРОК 1: Обчислення вузлів Чебишева ===
k=0: t_k=0.951057,  x_k=-0.629629
k=1: t_k=0.587785,  x_k=-0.811265
k=2: t_k=0.000000,  x_k=-1.105157
k=3: t_k=-0.587785,  x_k=-1.399050
k=4: t_k=-0.951057,  x_k=-1.580686

Відсортовані вузли: [-1.58068572 -1.39905009 -1.10515746 -0.81126483 -0.6296292 ]


=== КРОК 2: Значення функції у вузлах ===
      x_k    f(x_k)
-1.580686 11.590737
-1.399050  5.272450
-1.105157  0.000000
-0.811265 -2.118246
-0.629629 -2.664124
```

```
=== КРОК 3: Побудова інтерполяційного полінома Лагранжа ===

--- Обчислення L_0(x) ---
 L_0(x) = (1.05146222423827*x + 0.662031321379111)*(1.29967878493163*x + 1.05438369395838)*(2.1029244484
7653*x + 2.32406264275822)*(5.50552768188469*x + 7.70250897947482)

--- Обчислення L_1(x) ---
 L_1(x) = -(1.29967878493163*x + 0.818315716458591)*(1.70130161670408*x + 1.38020617398335)*(3.402603233
40816*x + 3.76041234796671)*(5.50552768188469*x + 8.70250897947482)

--- Обчислення L_2(x) ---
 L_2(x) = (2.10292444847653*x + 1.32406264275822)*(2.10292444847653*x + 3.32406264275822)*(3.40260323340
816*x + 2.76041234796671)*(3.40260323340816*x + 4.76041234796671)

--- Обчислення L_3(x) ---
 L_3(x) = -(1.29967878493163*x + 2.05438369395838)*(1.70130161670408*x + 2.38020617398335)*(3.4026032334
0816*x + 3.76041234796671)*(5.50552768188469*x + 3.46644100197503)

--- Обчислення L_4(x) ---
 L_4(x) = (1.05146222423827*x + 1.66203132137911)*(1.29967878493163*x + 1.81831571645859)*(2.10292444847
653*x + 2.32406264275822)*(5.50552768188469*x + 4.46644100197503)

=== Кінцевий поліном Лагранжа ===
P(x) =  10.5815205787002*x**4 + 29.4991961932518*x**3 + 37.6051418332908*x**2 + 21.0813167955984*x + 1.4
0155947570187
```

Опісля знаходження полінома застосуємо пряму інтерполяцію:

```
=== ПРЯМА ІНТЕРПОЛЯЦІЯ (перевірка коренів P(x)=0)) ===
Усі корені, які повернув sympy.solve(): [-1.10515746033664, -0.0762517519640395, -0.803196959028841 - 0.
962624716328455*I, -0.803196959028841 + 0.962624716328455*I]
Корінь (дійсна): -1.105157460337
 -> Приймаємо: -1.105157460337 знаходиться в інтервалі [-1.605157460337, -0.605157460337]
Корінь (дійсна): -0.076251751964
 -> Відкидаємо: -0.076251751964 НЕ належить інтервалу [-1.605157460337, -0.605157460337]
Корінь -0.803196959028841 - 0.962624716328455*I — комплексний (ім). Відкидаємо: Im=-9.626e-1
Корінь -0.803196959028841 + 0.962624716328455*I — комплексний (ім). Відкидаємо: Im=9.626e-1

Результат: знайдені корені інтерполяційного полінома в інтервалі:
  x = -1.105157460337  (взятий як наближений корінь f(x)=0 через пряму інтерполяцію)
```

Бачимо, що пряма інтерполяція дала коректний корінь, а отже і коректний результат. Тепер перейдемо до оберненої інтерполяції. Для цього побудуємо новий поліном за формулою $L_n(y)=\sum_{i=0}^{n} x_i \frac{w_{n+1}(y)}{(y-y_i)\omega'_{n+1}(y)}$

Проміжок і корені Чебишева залишуються ті ж, що і при прямій інтерполяції.

```
=== ОБЕРНЕНА ІНТЕРПОЛЯЦІЯ — Побудова x = P_inv(y) ===
Базисний L_0(y) (спрощено) = y*(6.98749288027326e-5*y**3 - 3.42443578397899e-5*y**2 - 0.0013675597013214
2*y - 0.00207904842472599)
Базисний L_1(y) (спрощено) = y*(-0.00051176403335355*y**3 + 0.00348427770995538*y**2 + 0.025479666149565
9*y + 0.0334743040064964)
Базисний L_2(y) (спрощено) = (0.0862757895042594*y - 1.0)*(0.189665137884604*y - 1.0)*(0.375357954587201
*y + 1.0)*(0.472088736173766*y + 1.0)
Базисний L_3(y) (спрощено) = y*(-0.00853567013220109*y**3 + 0.121198524426434*y**2 - 0.138158089339672*y
 - 1.38968236360604)
Базисний L_4(y) (спрощено) = y*(0.00607790833044644*y**3 - 0.0896184031437311*y**2 + 0.154325436746362*y
 + 0.786781344652167)

=== Кінцевий обернений поліном (спрощений) ===
x = P_inv(y) =  0.000498822823034985*y**4 - 0.00800444988273898*y**3 + 0.0259444897091359*y**2 - 0.04312
98418207077*y - 1.10515746033667
Пояснення: цей поліном апроксимує x як функцію від y = f(x).
```

Тепер можемо знайти x для якого y = 0:

```
=== Кінцевий обернений поліном (спрощений) ===
x = P_inv(y) =  0.000498822823034985*y**4 - 0.00800444988273898*y**3 + 0.0259444897091359*y**2 - 0.04312
98418207077*y - 1.10515746033667
Пояснення: цей поліном апроксимує x як функцію від y = f(x).
Підставляємо y=0 у P_inv(y): P_inv(0) = -1.10515746033667  -> чисельно: -1.10515746033667
Пояснення: оскільки y = f(x), при f(x)=0 отримаємо y=0, отже корінь x приблизно P_inv(0).
Результат оберненої інтерполяції: x ≈ -1.105157460337
```

Обернена інтерполяцій дала ідентичний результат до прямої, а отже обидва метода працюють коректно. Знайдений x ≈ -1.105.