

# ОСНОВИ МЕТОДІВ ОБЧИСЛЕНЬ

Нікіта Скибицький

3 січня 2019 р.

У ваших руках конспект лекцій з нормативного курсу “Основи методів обчислень”, прочитаного доц., к.ф.-м.н. Риженком Андрієм Івановичем на третьому курсі спеціальності прикладна математика факультету комп’ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка восени 2018-го року.

Конспект у компактній формі відображає матеріал курсу, допомагає сформулювати загальне уявлення про предмет вивчення, правильно зорієнтуватися в даній галузі знань. Конспект лекцій з названої дисципліни сприятиме більш успішному вивченню дисципліни, причому більшою мірою для студентів заочної форми, екстернату, дистанційного та індивідуального навчання.

Комп’ютерний набір та верстка – Скибицький Нікіта Максимович.

## Зміст

<b>1</b>	<b>Аналіз похибок заокруглення</b>	<b>4</b>
1.1	Види похибок	4
1.2	Підрахунок похибок в ЕОМ	5
1.3	Підрахунок похибок обчислення значення функції	6
1.3.1	Похибки арифметичних операцій	7
1.4	Обернена задача аналізу похибок	7
<b>2</b>	<b>Нелінійні рівняння</b>	<b>8</b>
2.1	Метод ділення навпіл	8
2.2	Метод простої ітерації	9
2.3	Метод релаксації	11
2.4	Метод Ньютона (метод дотичних)	13
2.5	Збіжність методу Ньютона	14
<b>3</b>	<b>Методи розв'язання СЛАР</b>	<b>18</b>
3.1	Метод Гаусса	18
3.2	Метод квадратних коренів	21
3.3	Обчислення визначника та оберненої матриці	23
3.4	Метод прогонки	23
3.5	Обумовленість систем лінійних алгебраїчних рівнянь	25
<b>4</b>	<b>Ітераційні методи для систем</b>	<b>28</b>
4.1	Ітераційні методи розв'язання СЛАР	28
4.1.1	Метод простої ітерації	28
4.1.2	Метод Якобі	29
4.1.3	Метод Зейделя	29
4.1.4	Матрична інтерпретація методів Якобі і Зейделя	30
4.1.5	Однокрокові (двошарові) ітераційні методи	30
4.1.6	Збіжності стаціонарних ітераційних процесів у випадку симетричних матриць	31
4.1.7	Метод верхньої релаксації	32
4.1.8	Методи варіаційного типу	32
4.2	Методи розв'язання нелінійних систем	33
4.2.1	Метод простої ітерації	33
4.2.2	Метод Ньютона	34
4.2.3	Модифікований метод Ньютона	35

<b>5</b>	<b>Проблема власних значень</b>	<b>35</b>
5.1	Степеневий метод . . . . .	36
5.2	Ітераційний метод обертання . . . . .	38
<b>6</b>	<b>Інтерполювання функцій</b>	<b>40</b>
6.1	Постановка задачі інтерполювання . . . . .	40
6.2	Інтерполяційна формула Лагранжа . . . . .	41
6.3	Залишковий член інтерполяційного полінома . . . . .	42
6.4	Багаточлени Чебишова. Мінімізація залишкового члена ін- терполяційного полінома . . . . .	43
6.5	Розділені різниці . . . . .	46
6.6	Інтерполяційна формула Ньютона . . . . .	47
6.7	Інтерполювання з кратними вузлами . . . . .	49
6.8	Збіжність процесу інтерполювання . . . . .	50
6.9	Кусково-лінійна інтерполяція . . . . .	53
6.10	Кусково-кубічна ермітова інтерполяція . . . . .	55

# 1 Аналіз похибок заокруглення

## 1.1 Види похибок

Нехай необхідно розв'язати рівняння

$$Au = f. \quad (1.1)$$

За рахунок неточно заданих вхідних даних насправді ми маємо рівняння

$$\tilde{A}\tilde{u} = \tilde{f}. \quad (1.2)$$

Назвемо  $\delta_1 = u - \tilde{u}$  – *неусувною похибкою*.

Застосування методу розв'язання (1.2) приводить до рівняння

$$\tilde{A}_h \tilde{u}_h = \tilde{f}_h, \quad (1.3)$$

де  $h > 0$  – малий параметр. Назвемо  $\delta_2 \tilde{u} - \tilde{u}_h$  – *похибкою методу*.

Реалізація методу на ЕОМ приводить до рівняння

$$\tilde{A}_h^* \tilde{u}_h^* = \tilde{f}_h^*. \quad (1.4)$$

Назвемо  $\delta_3 = \tilde{u}_h^* - \tilde{u}_h$  – *похибкою заокруглення*.

Тоді *повна похибка*  $\delta = u - \tilde{u}_h^* = \delta_1 + \delta_2 + \delta_3$ .

**Визначення 1.** Кажуть, що задача (1.1) *коректна*, якщо

1.  $\forall f \in F \exists! u \in U$ ;
2. задача (1.1) *стійка*, тобто

$$\forall \varepsilon > 0 \quad \exists \delta > 0 : \|A - \tilde{A}\| < \delta, \|f - \tilde{f}\| < \delta \Rightarrow \|u - \tilde{u}\| < \varepsilon. \quad (1.5)$$

Якщо задача (1.1) *некоректна*, то або розв'язок її не існує, або він не-єдиний, або він нестійкий, тобто

$$\exists \varepsilon > 0 : \forall \delta > 0 : \exists A, f : \|A - \tilde{A}\| < \delta, \|f - \tilde{f}\| < \delta, \|u - \tilde{u}\| > \varepsilon. \quad (1.6)$$

*Абсолютна похибка*  $\Delta(x^*) \geq \max_x |x - x^*|$ .

Відносна похибка  $\delta(x^*) \geq \max_x \Delta(x^*)/|x^*|$ .

Значущими цифрами називаються всі цифри, починаючи з першої ненульової зліва.

Вірна цифра – це значуща, якщо абсолютна похибка за рахунок відкидання всіх молодших розрядів не перевищує одиниці розряду цієї цифри. Тобто, якщо

$$x^* = \overline{\alpha_n \dots \alpha_0 \alpha_{-1} \dots \alpha_{-p} \dots}, \quad (1.7)$$

то  $\alpha_{-p}$  – вірна, якщо  $\Delta(x^*) \leq 10^{-p}$ .

Інколи  $\Delta(x^*) \leq w \cdot 10^{-p}$ ,  $1/2 \leq w < 1$ , наприклад,  $w = 0.55$ .

## 1.2 Підрахунок похибок в ЕОМ

Підрахуємо відносну похибку заокруглення числа  $x$  на ЕОМ з плаваючою комою. В  $\beta$ -ічній системі числення число представляється у вигляді

$$x = \pm(\alpha_1\beta^{-1} + \alpha_2\beta^{-2} + \dots + \alpha_t\beta^{-t} + \dots)\beta^p, \quad (1.8)$$

де  $0 \leq \alpha_k < \beta$ ,  $\alpha_1 \neq 0$ ,  $k = 1, 2, \dots$

Якщо в ЕОМ  $t$  розрядів, то при відкиданні молодших розрядів ми оперуємо з наближеним значенням

$$x^* = \pm(\alpha_1\beta^{-1} + \alpha_2\beta^{-2} + \dots + \alpha_t\beta^{-t})\beta^p, \quad (1.9)$$

і відповідно похибка заокруглення

$$x - x^* = \pm\beta^p(\alpha_{t+1}\beta^{-t-1} + \dots) \quad (1.10)$$

Тоді її можна оцінити так

$$\begin{aligned} |x - x^*| &\leq \beta^{p-t-1} \cdot (\beta - 1) \cdot (1 + \beta^{-1} + \dots) \leq \\ &\leq \beta^{p-t-1} \cdot (\beta - 1) \cdot \frac{1}{1 - \beta^{-1}} = \beta^{p-t}. \end{aligned} \quad (1.11)$$

Якщо в представленні (1.8) взяти  $\alpha_1 = 1$ , то  $|x| \geq \beta^p \cdot \beta^{-1}$ . Звідси остаточно

$$\delta(x^*) \leq \frac{\beta^{p-t}}{\beta^{p-1}} = \beta^{1-t}. \quad (1.12)$$

При більш точних способах заокруглення можна отримати оцінку  $\delta(x^*) \leq \beta^{1-t}/2 = \varepsilon$ . Число  $\varepsilon$  називається “машинним іпсилон”. Наприклад, для  $\beta = 2$ ,  $t = 24$ ,  $\varepsilon = 2^{-24} \approx 10^{-7}$ .

### 1.3 Підрахунок похибок обчислення значення функції

Нехай задана функція

$$y = f(x_1, \dots, x_n) \in C^1(\Omega) \quad (1.13)$$

Необхідно обчислити її значення при наближеному значенні аргументів

$$\vec{x}^* = (x_1^*, \dots, x_n^*), \quad (1.14)$$

де  $|x_i - x_i^*| \leq \Delta(x_i^*)$  та оцінити похибку обчислення значення функції

$$y^* = f(x_1^*, \dots, x_n^*). \quad (1.15)$$

Маємо

$$|y - y^*| = |f(\vec{x}) - f(\vec{x}^*)| = \left| \sum_{i=1}^n \frac{\partial f}{\partial x_i}(\vec{\xi})(x_i - x_i^*) \right| \leq \sum_{i=1}^n B_i \cdot \Delta(x_i^*), \quad (1.16)$$

де

$$B_i = \max_{\vec{x} \in U} \left| \frac{\partial f}{\partial x_i}(\vec{x}) \right|. \quad (1.17)$$

Тут

$$U = \{\vec{x} = (x_1, \dots, x_n) : |x_i - x_i^*| \leq \Delta(x_i^*)\} \in \Omega, \quad i = \overline{1, n}. \quad (1.18)$$

Отже

$$\Delta(y^*) = |y - y^*| \prec \sum_{i=1}^n n_i \cdot \Delta(x_i^*), \quad (1.19)$$

з точністю до величин першого порядку малості по

$$\Delta(x^*) = \max_{i=\overline{1, n}} \Delta(x_i^*), \quad (1.20)$$

де

$$b_i = \left| \frac{\partial f}{\partial x_i}(\vec{x}^*) \right| \quad (1.21)$$

та “ $\prec$ ” означає приблизно менше.

### 1.3.1 Похибки арифметичних операцій

1. Сума:  $y = x_1 + x_2$ ,  $x_1, x_2 > 0$ ,

$$\begin{aligned}\Delta(y^*) &\leq \Delta(x_1^*) + \Delta(x_2^*), \\ \delta(y^*) &\leq \frac{\Delta(x_1^*) + \Delta(x_2^*)}{x_1^* + x_2^*} \leq \max\{\delta(x_1^*), \delta(x_2^*)\}.\end{aligned}\tag{1.22}$$

2. Різниця:  $y = x_1 - x_2$ ,  $x_1 > x_2 > 0$ ,

$$\begin{aligned}\Delta(y^*) &\leq \Delta(x_1^*) + \Delta(x_2^*), \\ \delta(y^*) &\leq \frac{x_2^* \delta(x_1^*) + (x_1^*) \delta(x_2^*)}{x_1^* - x_2^*}.\end{aligned}\tag{1.23}$$

При близьких  $x_1^*$ ,  $x_2^*$  зростає відносна похибка (за рахунок втрати вірних цифр).

3. Добуток:  $y = x_1 \cdot x_2$ ,  $x_1, x_2 > 0$ ,

$$\begin{aligned}\Delta(y^*) &\prec x_2^* \Delta(x_1^*) + x_1^* \Delta(x_2^*), \\ \delta(y^*) &\leq \delta(x_1^*) + \delta(x_2^*).\end{aligned}\tag{1.24}$$

4. Частка:  $y = \frac{x_1}{x_2}$ ,  $x_1, x_2 > 0$ ,

$$\begin{aligned}\Delta(y^*) &\prec \frac{x_2^* \Delta(x_1^*) + x_1^* \Delta(x_2^*)}{(x_2^*)^2}, \\ \delta(y^*) &\leq \delta(x_1^*) + \delta(x_2^*).\end{aligned}\tag{1.25}$$

При малих  $x_2^*$  зростає абсолютна похибка (за рахунок зростання результату ділення).

## 1.4 Оборнена задача аналізу позибок

Нагадаємо, що *пряма задача* аналізу похибок полягає у обчисленні  $\Delta(y^*)$ ,  $\delta(y^*)$  по заданих  $\Delta(x_i^*)$ ,  $i = \overline{1, n}$ .

*Оборнена задача* полягає у знаходженні  $\Delta(x_i^*)$ ,  $i = \overline{1, n}$  по заданих  $\Delta(y^*)$ ,  $\delta(y^*)$ . Якщо  $n > 1$ , маємо одну умову

$$\sum_{i=1}^n b_i \Delta(x_i^*) < \varepsilon\tag{1.26}$$



для багатьох невідомих  $\Delta(x_i^*)$ . Вибирають їх із однієї з умов

$$b_i \Delta(x_i^*) < \frac{\varepsilon}{n}, \quad (1.27)$$

або

$$\Delta(x_i^*) < \frac{\varepsilon}{\sum_{i=1}^n b_i}. \quad (1.28)$$

## 2 Нелінійні рівняння

*Постановка задачі.* Нехай маємо рівняння  $f(x) = 0$ ,  $\bar{x}$  – його розв’язок, тобто  $f(\bar{x}) \equiv 0$ .

Задача розв’язання цього рівняння розпадається на етапи:

1. Існування та кількість коренів.
2. Відділення коренів, тобто розбиття числової вісі на інтервали, де знаходиться один корінь.
3. Обчислення кореня із заданою точністю  $\varepsilon$ .

Для розв’язання перших двох задач використовуються методи математичного аналізу та алгебри, а також графічний метод. Далі розглядаються методи розв’язання третього етапу.

### 2.1 Метод ділення навпіл

Припустимо на  $[a, b]$  знаходиться лише один корінь рівняння

$$f(x) = 0, \quad (2.1)$$

для  $f(x) \in C([a, b])$ , який необхідно визначити. Нехай  $f(a) \cdot f(b) < 0$ .

Припустимо, що  $f(a) > 0$ ,  $f(b) < 0$ . Покладемо  $x_1 = \frac{a+b}{2}$  і підрахуємо  $f(x_1)$ . Якщо  $f_1(x) < 0$ , тоді шуканий корінь  $x$  знаходиться на інтервалі  $(a, x_1)$ . Якщо ж  $f_1(x) > 0$ , то  $\bar{x} \in (x_1, b)$ , тобто з двох інтервалів  $(a, x_1)$  і  $(x_1, b)$  вибираємо той, на границях якого функція  $f(x)$  має різні знаки, знаходимо точку  $x_2$  – середину вибраного інтервалу, підраховуємо  $f(x_2)$  і повторюємо вказаний процес.

В результаті отримаємо послідовність інтервалів, що містять шуканий корінь  $\bar{x}$ , причому довжина кожного послідовного інтервалу вдвічі менше попереднього.

Цей процес продовжується до тих пір, поки довжина отриманого інтервалу  $(a_n, b_n)$  не стане меншою за  $b_n - a_n < 2\varepsilon$ . Тоді  $x_{n+1}$ , як середина інтервалу  $(a_n, b_n)$  пов'язане з  $\bar{x}$  нерівністю

$$|x_{n+1} - \bar{x}| < \varepsilon. \quad (2.2)$$

Ця умова для деякого  $n$  буде виконуватись за теоремою Больцано-Коші.

Оскільки

$$|b_{k+1} - a_{k+1}| = \frac{1}{2}|b_k - a_k|,$$

то

$$|x_{n+1} - \bar{x}| \leq \frac{1}{2^{n+1}}(b - a) < \varepsilon. \quad (2.3)$$

Звідси отримаємо нерівність для обчислення кількості ітерацій  $n$  для виконання умови (2.2):

$$n = n(\varepsilon) \geq \left\lceil \log \left( \frac{b - a}{\varepsilon} \right) \right\rceil + 1.$$

Степінь збіжності – лінійна, тобто геометричної прогресії з знаменником  $q = \frac{1}{2}$ .

Переваги методу: простота, надійність. Недоліки методу: низька швидкість збіжності; метод не узагальнюється на системи.

## 2.2 Метод простої ітерації

Спочатку рівняння

$$f(x) = 0 \quad (2.4)$$

замінюється еквівалентним

$$x = \varphi(x) \quad (2.5)$$

Ітераційний процес має вигляді

$$x_{n+1} = \varphi(x_n), \quad n = 0, 1, \dots \quad (2.6)$$

Початкове наближення  $x_0$  задається.

Для збіжності велике значення має вибір функції  $\varphi(x)$ . Перший спосіб заміни рівняння полягає в відділенні змінної з якогось члена рівняння. Більш продуктивним є перехід від рівняння (2.4) до (2.5) з функцією  $\varphi(x) = x + \tau(x) \cdot f(x)$ , де  $\tau(x)$  – знакостала функція на тому відрізку, де шукаємо корінь.

Кажуть, що ітераційний метод *збігається*, якщо  $\lim_{k \rightarrow \infty} x_k = \bar{x}$ .

Далі  $U_r = \{x : |x - a| \leq r\}$  відрізок довжини  $2r$  з серединою в точці  $a$ .

З’ясуємо умови, при яких збігається метод простої ітерації.

**Теорема 2.2.1.** *Якщо  $\max_{x \in U_r} |\varphi'(x)| \leq q < 1$ , то метод простої ітерації збігається і має місце оцінка*

$$|x_n - \bar{x}| \leq \frac{q^n}{1 - q} |x_0 - x_1| \leq \frac{q^n}{1 - q} (b - a). \quad (2.7)$$

*Доведення.* Нехай  $x_{k+1}, x_k \in U_r$ . Тоді має місце допоміжна нерівність:

$$\begin{aligned} |x_{k+1} - x_k| &= |\varphi(x_k) - \varphi(x_{k-1})| = |\varphi'(\xi_k)(x_k - x_{k-1})| \leq (*) \\ \text{тут } \xi_k &= x_k + \theta_k(x_{k+1} - x_k), \quad 0 < \theta_k < 1 \\ (*) &\leq |\varphi'(\xi_k)| \cdot |x_k - x_{k-1}| \leq q|x_k - x_{k-1}| = \dots = q^k|x_1 - x_0|. \end{aligned}$$

Використаємо її для доведення теореми:

$$\begin{aligned} |x_{k+p} - x_k| &= |x_{k+p} - x_{k+p-1} + \dots + x_{k+1} - x_k| \leq \\ &\leq |x_{k+p} - x_{k+p-1}| + \dots + |x_{k+1} - x_k| \leq \\ &\leq (q^{k+p-1} + q^{k+p-2} + \dots + q^k)|x_1 - x_0| = \\ &= \frac{q^k - q^{k+p-1}}{1 - q} |x_1 - x_0| \xrightarrow[k \rightarrow \infty]{} 0. \end{aligned} \quad (2.8)$$

Бачимо що  $\{x_k\}$  – фундаментальна послідовність. Значить вона збіжна. При  $p \rightarrow \infty$  в (2.8) отримуємо (2.7).  $\square$

Визначимо кількість ітерацій для досягнення точності  $\varepsilon$ . З оцінки в теоремі отримаємо

$$|x_n - \bar{x}| \leq \frac{q^n}{1 - q} (b - a) < \varepsilon \Rightarrow n(\varepsilon) = n \geq \left\lceil \frac{\ln \left( \frac{\varepsilon(1 - q)}{b - a} \right)}{\ln q} \right\rceil + 1.$$

Практично ітераційний процес зупиняємо при:  $|x_n - x_{n-1}| < \varepsilon$ . Але ця умова не завжди гарантує, що  $|x_n - \bar{x}| < \varepsilon$ .

**Зауваження.** Умова збіжності методу може бути замінена на умову Ліпшиця

$$|\varphi(x) - \varphi(y)| \leq q \cdot |x - y|, \quad 0 < q < 1.$$

Переваги методу: простота; при  $q < \frac{1}{2}$  – швидше збігається ніж метод ділення навпіл; метод узагальнюється на системи. Недоліки методу:

1. при  $q > \frac{1}{2}$  – збігається повільніше ніж метод ділення навпіл,
2. виникають труднощі при зведенні  $f(x) = 0$  до  $x = \varphi(x)$ .

## 2.3 Метод релаксації

Якщо в методі простої ітерації для рівняння  $x = x + \tau f(x) \equiv \varphi(x)$  вибрати  $\tau(x) = \tau = \text{const}$ , то ітераційний процес приймає вигляд

$$x_{n+1} = x_n + \tau \cdot f(x_n), \quad n = 0, 1, 2, \dots \quad (2.9)$$

$x_0$  – задано. Метод можна записати у вигляді

$$\frac{x_{k+1} - x_k}{\tau} = f(x_k), \quad k = 0, 1, \dots$$

Оскільки  $\varphi'(x) = 1 + \tau \cdot f'(x)$ , то метод збігається при умові

$$|\varphi'(x)| = |1 + \tau \cdot f'(x)| \leq q < 1.$$

Нехай  $f'(x) < 0$ , тоді (2.7) запишеться у вигляді:  $-q \leq 1 + \tau \cdot f'(x) \leq q < 1$ . Звідси

$$\tau \cdot |f'(x_k)| \leq 1 + q < 2 \quad \text{і} \quad 0 < \tau < \frac{2}{|f'(x)|}.$$

Поставимо задачу знаходження  $\tau$ , для якого  $q = q(\tau) \rightarrow \min$ . Для того, щоб вибрати оптимальний параметр  $\tau$ , розглянемо рівняння для похибки  $z_k = x_j - \bar{x}$ .

Підставивши  $x_k = x + z_k$  в (2.9), отримаємо

$$z_{k+1} = z_k + \tau \cdot f(\bar{x} + z_k).$$

В припущенні  $f(x) \in C^{(1)}([a, b])$  з теореми про середнє маємо

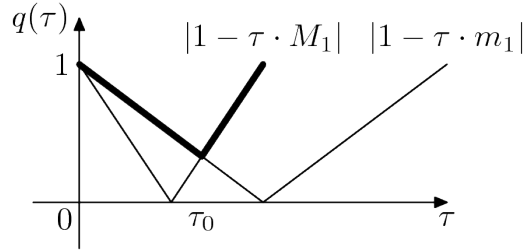
$$f(\bar{x} + z_k) = f(\bar{x}) + z_k \cdot f'(\bar{x} + \theta z_k) = z_k \cdot f'(\bar{x} + \theta z_k) = z_k \cdot f'(\xi_k),$$

$$\begin{aligned}
z_{k+1} &= z_k + \tau \cdot f'(\xi_k) \cdot z_k, \\
|z_{k+1}| &\leq |1 + \tau \cdot f'(\xi_k)| \cdot |z_k| \leq \max_{\xi_k \in U} |1 + \tau \cdot f'(\xi_k)| \cdot |z_k|, \\
|z_{k+1}| &\leq \max\{|1 - \tau \cdot M_1|, |1 - \tau \cdot m_1|\} \cdot |z_k|, \\
m_1 &= \min_{x \in [a, b]} |f'(x)|, \quad M_1 = \max_{x \in [a, b]} |f'(x)|.
\end{aligned}$$

Таким чином, задача вибору оптимального параметра зводиться до знаходження  $\tau$ , для якого функція

$$q(\tau) = \max\{|1 - \tau \cdot M_1|, |1 - \tau \cdot m_1|\}$$

приймає мінімальне значення:  $q(\tau) \rightarrow \min$ .



З графіка видно, що точка мінімуму визначається умовою  $|1 - \tau \cdot M_1| = |1 - \tau \cdot m_1|$ .

Тому

$$1 - \tau_0 \cdot m_1 = \tau_0 \cdot M_1 - 1 \Rightarrow \tau_0 = \frac{2}{M_1 + m_1} < \frac{2}{|f'(x)|}.$$

При цьому значенні  $\tau$  маємо

$$q(\tau_0) = q_0 = \frac{M_1 - m_1}{M_1 + m_1}.$$

Тоді для похибки вірна оцінка

$$|x_n - \bar{x}| \leq \frac{q_0^n \cdot (b - a)}{1 - q_0} < \varepsilon.$$

Кількість ітерацій

$$n = n(\varepsilon) \geq \left\lceil \frac{\ln \left( \frac{\varepsilon \cdot (1 - q_0)}{b - a} \right)}{\ln q_0} \right\rceil + 1.$$

**Задача 1.** Дати геометричну інтерпретацію методу простої ітерації для випадків:

$$0 < \varphi'(x) < 1; \quad -1 < \varphi'(x) < 0; \quad \varphi'(x) < -1; \quad \varphi'(x) > 1,$$

**Задача 2.** Знайти оптимальне  $\tau = \tau_0$  для методу релаксації при  $f'(x) > 0$ .

## 2.4 Метод Ньютона (метод дотичних)

Припустимо, що рівняння  $f(x) = 0$  має простий дійсний корінь  $\bar{x}$ , тобто  $f(\bar{x}) = 0$ ,  $f'(\bar{x}) \neq 0$ . Нехай виконуються умови:  $f(x) \in C^{(1)}([a, b])$ ,  $f(a) \cdot f(b) < 0$ . Тоді

$$0 = f(\bar{x}) = f(x_k + \bar{x} - x_k) = f(x_k) + f'(\xi_k) \cdot (\bar{x} - x_k),$$

де  $\xi_k = x_k + \theta_k \cdot (\bar{x} - x_k)$ ,  $0 < \theta_k < 1$ ,  $\xi_k \approx x_k$ . Тому наступне наближення виберемо з рівняння

$$f(x_k) + f'(x_k) \cdot (x_{k+1} - x_k) = 0.$$

Звідси маємо ітераційний процес

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, 2, \dots, \quad x_0 - \text{задане}.$$

Метод Ньютона ще називають методом лінеаризації або методом дотичних.

**Задача 3.** Дати геометричну інтерпретацію методу Ньютона.

Метод Ньютона можна інтерпретувати як метод простої ітерації з

$$\varphi(x) = x - \frac{f(x)}{f'(x)}, \quad \text{тобто} \quad \tau(x) = -\frac{1}{f'(x)}.$$

Тому

$$\varphi'(x) = 1 - \frac{f'(x) \cdot f'(x) - f(x) \cdot f''(x)}{(f'(x))^2} = \frac{f(x) \cdot f''(x)}{(f'(x))^2}.$$

Якщо  $\bar{x}$  – корінь  $f(x)$ , то  $\varphi'(x) = 0$ . Тому знайдеться окіл кореня, де

$$|\varphi'(x)| = \left| \frac{f(x) \cdot f''(x)}{(f'(x))^2} \right| < 1.$$

Це означає, що збіжність методу Ньютона залежить від вибору  $x_0$ .

Недолік методу Ньютона: необхідність обчислювати на кожній ітерації не тільки значення функції, а й похідної.

Модифікований метод Ньютона позбавлений цього недоліку і має вигляд:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_0)}, \quad k = 0, 1, 2, \dots$$

Цей метод має лише лінійну збіжність:  $|x_{k+1} - \bar{x}| = O(|x_k - \bar{x}|)$ .

**Задача 4.** Дати геометричну інтерпретацію модифікованого методу Ньютона.

В методі Ньютона, для якого  $f'(x_k)$  замінюється на  $\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$  дає метод січних:

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} \cdot f(x_k), \quad k = 1, 2, \dots, \quad x_0, x_1 - \text{задані.}$$

**Задача 5.** Дати геометричну інтерпретацію методу січних.

## 2.5 Збіжність методу Ньютона

**Теорема 2.5.1.** Нехай  $f(x) \in C^{(2)}([a, b])$ ,  $\bar{x}$  – простий дійсний корінь рівняння

$$f(x) = 0 \tag{2.10}$$

і  $f'(x) \neq 0$  при  $x \in U_r = \{x : |x - \bar{x}| < r\}$ . Якщо

$$\frac{M_2 \cdot |x_0 - \bar{x}|}{2m_1} = q < 1 \tag{2.11}$$

де  $m_1 = \min_{x \in U_r} |f'(x)|$ ,  $M_2 = \max_{x \in U_r} |f''(x)|$ , то для  $x_0 \in U_r$  метод Ньютона

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{2.12}$$

збігається і має місце оцінка

$$|x_n - \bar{x}| \leq q^{2^n - 1} \cdot |x_0 - \bar{x}|. \tag{2.13}$$

Доведення. З (2.12) маємо

$$x_{k+1} - \bar{x} = x_k - \frac{f(x_k)}{f'(x_k)} - \bar{x} = \frac{(x_k - \bar{x}) \cdot f'(x_k) - f(x_k)}{f'(x_k)} = \frac{F(x_k)}{f'(x_k)}, \quad (2.14)$$

де  $F(x) = (x - \bar{x}) \cdot f'(x) - f(x)$ , така, що

1.  $F(x) = 0$ ;
2.  $F'(x) = (x - \bar{x}) \cdot f''(x)$ ;

Тоді

$$F(x_k) = F(\bar{x}) + \int_{\bar{x}}^{x_k} F'(t) dt = \int_{\bar{x}}^{x_k} ((t - \bar{x}) \cdot f''(t)) dt.$$

Так як  $(t - \bar{x})$  не міняє знак на відрізку інтегрування, то скористаємося теоремою про середнє значення:

$$F(x_k) = f''(\xi_k) \int_{\bar{x}}^{x_k} (t - \bar{x}) dt = \frac{(x_k - \bar{x})^2}{2} f''(\xi_k), \quad (2.15)$$

де  $\xi_k = \bar{x} + \theta_k \cdot (x_k - \bar{x})$ ,  $0 < \theta_k < 1$ . З (2.14), (2.15) маємо

$$x_{k+1} - \bar{x} = \frac{(x_k - \bar{x})^2}{2f'(x_k)} f''(\xi_k). \quad (2.16)$$

Доведемо оцінку (2.12) за індукцією. Так як  $x_0 \in U_r$ , то

$$|\xi_0 - \bar{x}| = |\theta_0 \cdot (x_0 - \bar{x})| < |\theta_0| \cdot |x_0 - \bar{x}| < r \Rightarrow \xi_0 \in U_r.$$

Тоді  $f''(\xi_0) \leq M_2$ , тому

$$\begin{aligned} |x_1 - \bar{x}| &\leq \frac{(x_0 - \bar{x})^2 \cdot M_2}{2m_1} = \frac{M_2 \cdot |x_0 - \bar{x}|}{2m_1} |x_0 - \bar{x}| = \\ &= q \cdot |x_0 - \bar{x}| = q \cdot |x_0 - \bar{x}| < r \Rightarrow x_1 \in U_r. \end{aligned}$$

Ми довели твердження (2.13) при  $n = 1$ . Нехай воно справджується при  $n = k$ :

$$|x_k - \bar{x}| \leq q^{2^k - 1} \cdot |x_0 - \bar{x}| < r, \quad |\xi_k - \bar{x}| = |\theta_k \cdot (x_k - \bar{x})| < r.$$

Тоді  $x_k, \xi_k \in U_r$ .



Доведемо (2.13) для  $n = k + 1$ . З (2.16) маємо

$$\begin{aligned} |x_{k+1} - \bar{x}| &\leq \frac{|x_k - \bar{x}|^2 \cdot M_2}{2m_1} \leq \left(q^{2^k-1}\right)^2 \frac{|x_0 - \bar{x}|^2 \cdot M_2}{2m_1} = \\ &= q^{2^{k+1}-2} \frac{|x_0 - \bar{x}| \cdot M_2}{2m_1} |x_0 - \bar{x}| = q^{2^{k+1}-1} \cdot |x_0 - \bar{x}|. \end{aligned}$$

Таким чином (2.13) справджується для  $n = k + 1$ . Значить (2.13) виконується і для довільного  $n$ . Таким чином  $x_n \xrightarrow{n \rightarrow \infty} \bar{x}$ .  $\square$

З (2.13) маємо оцінку кількості ітерацій для досягнення точності  $\varepsilon$ :

$$n \geq \left\lceil \log_2 \left( 1 + \frac{\ln \left( \frac{\varepsilon}{b-a} \right)}{\ln q} \right) \right\rceil + 1.$$

Кажуть, що ітераційний метод має *ступінь збіжності*  $m$ , якщо

$$|x_{k+1} - \bar{x}| = O(|x_k - \bar{x}|^m).$$

Для методу Ньютона

$$|x_{k+1} - \bar{x}| = \frac{|x_k - \bar{x}|^2 \cdot |f''(\xi_k)|}{2|f'(x_k)|} \Rightarrow |x_{k+1} - \bar{x}| = O(|x_k - \bar{x}|^2).$$

Значить ступінь збіжності методу Ньютона  $m = 2$ . Для методу простої ітерації і ділення навпіл  $m = 1$ .

**Теорема 2.5.2.** Нехай  $f(x) \in C^{(2)}([a, b])$  та  $\bar{x}$  – простий корінь рівняння  $f(x) = 0$  а також  $\forall x \in [a, b]: f'(x) \neq 0$ . Якщо  $f'(x) \cdot f''(x) > 0$  ( $f'(x) \cdot f''(x) < 0$ ) то для методу Ньютона при  $x_0 = b$  послідовність наближень  $\{x_k\}$  монотонно спадає (монотонно зростає при  $x_0 = a$ ).

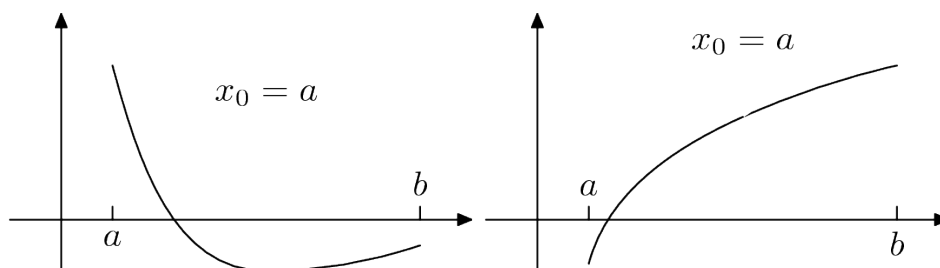
**Задача 6.** Довести теорему при

1.  $f'(x) \cdot f''(x) > 0$ ;
2.  $f'(x)f''(x) < 0$ .

**Задача 7.** Знайти ступінь збіжності методу січних.

Якщо  $f(a) \cdot f''(a) > 0$  та  $f''(x)$  не міняє знак, то потрібно вибирати  $x_0 = a$ ; при цьому  $\{x_k\} \uparrow \bar{x}$ .

Якщо  $f(b) \cdot f''(b) > 0$ , то  $x_0 = b$ ; маємо  $\{x_k\} \downarrow \bar{x}$ . Пояснення на рисунку:



**Зауваження 1.** Якщо  $\bar{x}$  –  $p$ -кратний корінь тобто  $f^{(m)}(x) = 0$ ,  $m = \overline{0, p-1}$ ,  $f^{(p)}(x) \neq 0$ , то в методі Ньютона необхідна наступна модифікація

$$x_{k+1} = x_k - p \frac{f(x_k)}{f'(x_k)} \quad \text{і} \quad q = \frac{M_{p+1} \cdot |x_0 - \bar{x}|}{m_p \cdot (p+1)} < 1.$$

**Зауваження 2.** Метод Ньютона можна застосовувати і для обчислення комплексного кореня, тоді ітераційний процес має вигляд

$$z_{k+1} = z_k - \frac{f(z_k)}{f'(z_k)}, \quad k = 0, 1, \dots$$

В теоремі про збіжність  $q = \frac{|z_0 - \bar{z}| \cdot M_2}{2m_1}$ , де  $m_1 = \min_{z \in U_r} |f'(z)|$ ,  $M_2 = \max_{z \in U_r} |f''(z)|$ .

Тут  $|z|$  – модуль комплексного числа  $z$ .

Переваги методу Ньютона:

1. висока швидкість збіжності;
2. узагальнюється на системи рівнянь;
3. узагальнюється на комплексні корені.

Недоліки методу Ньютона:

1. на кожній ітерації обчислюється не тільки  $f(x_k)$ , а і похідна  $f'(x_k)$ ;
2. збіжність залежить від початкового наближення  $x_0$ , так як від нього залежить умова збіжності  $q = \frac{M_2 \cdot |x_0 - \bar{x}|}{2m_1} < 1$ ;
3. потрібно, щоб  $f(x) \in C^{(2)}([a, b])$ .

### 3 Методи розв'язання СЛАР

Методи розв'язування СЛАР поділяються на прямі та ітераційні. При умові точного виконання обчислень прямі методи за скінчену кількість операцій в результаті дають точний розв'язок. Використовуються вони для невеликих та середніх СЛАР  $n = 10^2..10^4$ . Ітераційні методи використовуються для великих СЛАР  $n > 10^5$ , як правило розріджених. В результаті отримуємо послідовність наближень, яка збігається до розв'язку.

#### 3.1 Метод Гаусса

Розглянемо задачу розв'язання СЛАР

$$A\vec{x} = \vec{b}, \quad (3.1)$$

причому  $A = (a_{i,j})_{i,j=1}^n$ ,  $\det A \neq 0$ ,  $\vec{x} = (x_i)_{i=1}^n$ ,  $\vec{b} = (b_j)_{j=1}^n$ . Метод Крамера з обчисленням визначників для такої системи має складність  $Q = O(n!n)$ .

Запишемо СЛАР у вигляді

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,n}x_n = b_1 \equiv a_{1,n+1} \\ a_{2,1}x_1 + a_{2,2}x_2 + \dots + a_{2,n}x_n = b_2 \equiv a_{2,n+1} \\ \dots \\ a_{n,1}x_1 + a_{n,2}x_2 + \dots + a_{n,n}x_n = b_n \equiv a_{n,n+1} \end{cases} \quad (3.2)$$

Якщо  $a_{1,1} \neq 0$ , то ділимо перше рівняння на нього і виключаємо  $x_1$  з інших рівнянь:

$$\begin{cases} x_1 + a_{1,2}^{(1)}x_2 + \dots + a_{1,n}^{(1)}x_n = a_{1,n+1}^{(1)} \\ a_{2,2}^{(1)}x_2 + \dots + a_{2,n}^{(1)}x_n = a_{2,n+1}^{(1)} \\ \dots \\ a_{n,2}^{(1)}x_2 + \dots + a_{n,n}^{(1)}x_n = a_{n,n+1}^{(1)} \end{cases}$$

Процес повторюємо для  $x_2, \dots, x_n$ . В результаті отримуємо систему з трикутною матрицею

$$\begin{cases} x_1 + a_{1,2}^{(1)}x_2 + \dots + a_{1,n}^{(1)}x_n = a_{1,n+1}^{(1)} \\ x_2 + \dots + a_{2,n}^{(2)}x_n = a_{2,n+1}^{(2)} \\ \dots \\ x_n = a_{n,n+1}^{(n)} \end{cases} \quad (3.3)$$

Це прямий хід методу Гаусса. Формули прямого ходу

$$\begin{cases} k = \overline{1, n-1} : \\ a_{k,j}^{(k)} = \frac{a_{k,j}^{(k-1)}}{a_{k,k}^{(k-1)}}, & j = \overline{k+1, n+1}, \\ a_{i,j}^{(k)} = a_{i,j}^{(k-1)} - a_{i,k}^{(k-1)} a_{k,j}^{(k)}, \\ i = \overline{k+1, n}, & j = \overline{k+1, n+1}. \end{cases}$$

Звідси

$$x_n = a_{n,n+1}^{(n)}, \quad x_i = a_{i,n+1}^{(i)} - \sum_{j=i+1}^n a_{i,j}^{(i)} x_j, \quad i = \overline{n-1, 1}. \quad (3.4)$$

Це формули оберненого ходу.

Складність, тобто кількість операцій, яку необхідно виконати для реалізації методу,  $-Q_{\text{пр}} = \frac{2}{3}n^3 + O(n^2)$  для прямого ходу,  $Q_{\text{об}} = n^2 + O(n)$  для оберненого ходу.

Умова  $a_{k,k}^{(k-1)} \neq 0$  не суттєва, оскільки знайдеться  $m$ , для якого  $|a_{m,k}^{(k-1)}| = \max_i |a_{i,k}^{(k-1)}| \neq 0$  (оскільки  $\det A \neq 0$ ). Тоді міняємо місцями рядки номерів  $k$  і  $m$ . Елемент  $a_{k,k}^{(k-1)} \neq 0$  називається ведучим.

Введемо матриці

$$M_k = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & m_{k,k} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & m_{n,k} & \cdots & 1 \end{pmatrix},$$

елементи якої обчислюється так

$$m_{k,k} = \frac{1}{a_{k,k}^{(k-1)}}, \quad m_{i,k} = -\frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}}.$$

Нехай на  $k$ -му кроці  $A_{k-1}\vec{x} = \vec{b}_{k-1}$ . Множимо цю СЛАР зліва на  $M_k$ :  $M_k A_{k-1}\vec{x} = M_k \vec{b}_{k-1}$ . Позначимо  $A_k = M_k A_{k-1}$ ;  $A_0 = A$ . Тоді прямий хід методу Гаусса можна записати у вигляді

$$M_n M_{n-1} \dots M_1 A \vec{x} = M_n M_{n-1} \dots M_1 \vec{b}.$$

Позначимо останню систему, яка співпадає з (3.2), так

$$U\vec{x} = \vec{c}, \quad U = (u_{i,j})_{i,j=1}^n, \quad (3.5)$$

причому

$$\begin{cases} u_{i,i} = 1 \\ u_{i,j} = 0, \quad i > j \end{cases}$$

Таким чином  $U = M_n M_{n-1} \dots M_1 A$ . Введемо матриці

$$L_k = M_k^{-1} = \begin{pmatrix} 1 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & a_{k,k}^{(k-1)} & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & a_{n,k}^{(k-1)} & \dots & 1 \end{pmatrix},$$

Тоді

$$A = L_1 \dots L_n U = LU; \quad L = L_1 \dots L_n,$$

$L$  – нижня трикутна матриця,  $U$  – верхня трикутна матриця. Таким чином метод Гаусса можна трактувати, як розклад матриці в добуток двох трикутних матриць –  $(LU)$ -розклад.

Введемо матрицю  $P_k$  перестановок на  $k$ -му кроці (це матриця, отримана з одиничної матриці перестановкою  $k$ -того і  $m$ -того рядка). Тоді при множенні на неї матриці  $A_{k-1}$  робимо ведучим елементом максимальний за модулем.

$$P_k = \begin{pmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{pmatrix}.$$

За допомогою цих матриць перехід до трикутної системи (3.5) тепер має вигляд:

$$M_n M_{n-1} P_{n-1} \dots M_1 P_1 A \vec{x} = M_n M_{n-1} P_{n-1} \dots M_1 P_1 \vec{b}.$$

**Твердження:** знайдеться така матриця – перестановок, що  $PA = LU$  – розклад матриці на нижню трикутну з ненульовими діагональними елементами і верхню трикутну матрицю з одиницями на діагоналі.

**Висновки про переваги трикутного розкладу:**

1. Розділення прямого і оберненого ходів дає змогу економно розв’язувати декілька систем з одноковою матрицею та різними правими частинами.
2. Зберігання  $M$ , або  $L$  та  $U$  на місці  $A$ .
3. Обчислюючи  $l$  – кількість перестановок, можна встановити знак визначника.

### 3.2 Метод квадратних коренів

Цей метод призначений для розв’язання систем рівнянь із симетричною матрицею

$$A\vec{x} = \vec{b}, \quad A^T = A. \quad (3.6)$$

Він оснований на розкладі матриці  $A$  в добуток:

$$A = S^T D S, \quad (3.7)$$

$S$  – верхня трикутна матриця,  $S^T$  – нижня трикутна матриця,  $D$  – діагональна матриця.

Виникає питання: як обчислити  $S$ ,  $D$  по матриці  $A$ ? Маємо

$$\begin{aligned} (DS)_{i,j} &= \begin{cases} d_{i,i}s_{i,j}, & i \leq j \\ 0, & i > j \end{cases} \\ (S^T DS)_{i,j} &= \sum_{l=1}^n s_{i,l}^T d_{l,l} s_{l,j} = \sum_{l=1}^{i-1} s_{l,i} s_{l,j} d_{l,l} + s_{i,i} s_{i,j} d_{i,i} + \\ &+ \underbrace{\sum_{l=i+1}^n s_{l,i} s_{l,j} d_{l,l}}_{=0} = a_{i,j}, \quad i, j = \overline{1, n}. \end{aligned} \quad (3.8)$$

Якщо  $i = j$ , то

$$|s_{i,i}^2| d_{i,i} = a_{i,i} - \sum_{l=1}^{i-1} |s_{l,i}^2| d_{l,l} \equiv p_i.$$

Тому

$$d_{i,i} = \text{sign}(p_i), \quad s_{i,i} = \sqrt{|p_i|}.$$

Якщо  $i < j$ , то

$$s_{i,j} = \left( a_{i,j} - \sum_{l=1}^{i-1} s_{l,i} d_{l,l} s_{l,j} \right) / (s_{i,i} d_{i,i}), \quad i = \overline{1, n}, \quad j = \overline{i+1, n}.$$

Якщо  $A > 0$  (тобто головні мінори матриці  $A$  додатні), то всі  $d_{i,i} = +1$ .

Знайдемо розв'язок рівняння (3.6). Враховуючи (3.7), маємо:

$$S^T D \vec{y} = \vec{b} \quad (3.9)$$

$$S \vec{x} = \vec{y} \quad (3.10)$$

Оскільки  $S$  – верхня трикутна матриця, а  $S^T D$  – нижня трикутна матриця, то

$$y_i = \frac{b_i - \sum_{j=1}^{i-1} s_{j,i} s_{j,j} y_j}{s_{i,i} d_{i,i}}, \quad i = \overline{1, n} \quad (3.11)$$

$$x_n = \frac{y_n}{s_{n,n}}, \quad x_i = \frac{y_i - \sum_{j=1}^{i-1} s_{i,j} x_j}{s_{i,i}}, \quad i = \overline{n-1, 1}. \quad (3.12)$$

Метод застосовується лише для симетричних матриць. Його складність –  $Q = \frac{1}{3}n^3 + O(n^2)$ .

Переваги цього методу:

1. він витрачає в 2 рази менше пам'яті ніж метод Гаусса для зберігання  $A^T = A$  (необхідний об'єм пам'яті  $\frac{n(n+1)}{2} \sim \frac{n^2}{2}$ ;
2. метод однорідний, без перестановок;
3. якщо матриця  $A$  має багато нульових елементів, то і матриця  $S$  також.

Остання властивість дає економію в пам'яті та кількості арифметичних операцій. Наприклад, якщо  $A$  має  $m$  ненульових стрічок по діагоналі, то  $Q = O(m^2 n)$ .

### 3.3 Обчислення визначника та оберненої матриці

Кількість операцій обчислення детермінанту за означенням –  $Q_{\det} = n!$ . В методі Гаусса –  $PA = LU$ . Тому

$$\det P \det A = \det L \det U \Rightarrow \det A = (-1)^l \det L \det U = (-1)^l \prod_{k=1}^n a_{k,k}^{(k)}, \quad (3.13)$$

де  $l$  – кількість перестановок. Ясно, що за методом Гаусса

$$Q_{\det} = \frac{2}{3}n^3 + O(n^2).$$

В методі квадратного кореня  $A = S^T D S$ . Тому

$$\det A = \det S^T \det D \det S = \prod_{k=1}^n d_{k,k} \prod_{k=1}^n s_{k,k}^2. \quad (3.14)$$

Тепер  $Q_{\det} = \frac{1}{3}n^3 + O(n^2)$ .

За означенням

$$AA^{-1} = E, \quad (3.15)$$

де  $A^{-1}$  обернена до матриці  $A$ . Позначимо

$$A^{-1} = (\alpha_{i,j})_{i,j=1}^n.$$

Тоді  $\vec{\alpha}_j = (\alpha_{i,j})_{i=1}^n$  – вектор-стовпчик оберненої матриці. З (3.15) маємо

$$A\vec{\alpha}_j = \vec{e}_j, \quad j = \overline{1, n}, \quad (3.16)$$

$\vec{e}_j$  – стовпчики одиничної матриці:  $\vec{e}_j = (\delta_{i,j})_{i=1}^n$ ,  $\delta_{i,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$ .

Для знаходження  $A^{-1}$  необхідно розв'язати  $n$  систем. Для знаходження  $A^{-1}$  методом Гаусса необхідна кількість операцій  $Q = 2n^3 + O(n^2)$ .

### 3.4 Метод прогонки

Це економний метод для розв'язання СЛАР з три-діагональною матрицею:

$$-c_0 y_0 + b_0 y_1 = -f_0, \quad (3.17)$$

$$a_i y_{i-1} - c_i y_i + b_i y_{i+1} = -f_i, \quad (3.18)$$



$$a_N y_{N-1} - c_N y_N = -f_N. \quad (3.19)$$

Матриця системи

$$A = \begin{pmatrix} -c_0 & b_1 & & 0 \\ a_0 & -c_1 & \ddots & \\ & \ddots & \ddots & b_N \\ 0 & & a_N & -c_N \end{pmatrix}$$

тридіагональна.

Розв'язок представимо у вигляді

$$y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1}, \quad i = \overline{0, N-1}. \quad (3.20)$$

Замінімо в (3.20)  $i \mapsto i-1$  і підставимо в (3.18), тоді

$$(a_i \cdot \alpha_i - c_i) \cdot y_i + b_i \cdot y_{i+1} = -f_i - a_i \beta_i.$$

Звідси

$$y_i = \frac{b_i}{c_i - a_i \cdot \alpha_i} y_{i+1} + \frac{f_i + a_i \cdot \beta_i}{c_i - a_i \cdot \alpha_i}.$$

Тому з (3.18)

$$\alpha_{i+1} = \frac{b_i}{c_i - a_i \cdot \alpha_i}, \quad \beta_{i+1} = \frac{f_i + a_i \cdot \beta_i}{c_i - a_i \cdot \alpha_i}, \quad i = \overline{1, N-1}.$$

Умова розв'язності (3.17)  $c_i - a_i \cdot \alpha_i \neq 0$ .

Щоб знайти всі  $\alpha_i$ ,  $\beta_i$ , треба задати перші значення. З (3.17):

$$\alpha_1 = \frac{b_0}{c_0}, \quad \beta_1 = \frac{f_0}{c_0}. \quad (3.21)$$

Після знаходження всіх  $\alpha_i$ ,  $\beta_i$  обчислюємо  $y_N$  з системи

$$\begin{cases} a_N \cdot y_N - c_N \cdot y_N = -f_N, \\ y_{N-1} = \alpha_N \cdot y_N + \beta_N \end{cases}$$

Звідси

$$y_N = \frac{f_N + a_N \cdot \beta_N}{c_N - a_N \cdot \alpha_N}. \quad (3.22)$$

### Алгоритм

1. Покладемо  $\alpha_1 = \frac{b_0}{c_0}$ ,  $\beta_1 = \frac{f_0}{c_0}$ .

2. Позначимо  $z_i = c_i - a_i \alpha_i$ , обчислимо  $\alpha_{i+1} = \frac{b_i}{z_i}$ ,  $\beta_{i+1} = \frac{f_i + a_i \beta_i}{z_i}$ , для  $i = \overline{1, N-1}$ .
3. Знайдемо  $y_N = \frac{f_N + a_N \beta_N}{c_N - a_N \alpha_N}$ .
4. Обчислюємо  $y_i = \alpha_{i+1} \cdot y_{i+1} + \beta_{i+1}$ ,  $i = \overline{N-1, 0}$ .

Складність алгоритму  $Q = 8N - 2$ .

Метод можна застосовувати, коли  $\forall i: c_i - a_i \cdot \alpha_i \neq 0$ ,  $|\alpha_i| \leq 1$ . Якщо  $|\alpha_i| \geq q > 1$ , то  $\Delta y_0 \geq q^N \cdot \Delta y_N$  (тут  $\Delta y_i$  абсолютна похибка обчислення  $y_i$ ), а це приводить до експоненціального накопичення похибок заокруглення, тобто нестійкості алгоритму прогонки.

**Теорема 3.4.1** (про достатні умови стійкості метода прогонки). *Нехай*

$$\forall i: a_i, b_i \neq 0, \quad \text{та} \quad |c_i| \geq |a_i| + |b_i| \quad (a_0 = b_N = 0),$$

*та хоча би одна нерівність строга. Тоді*

$$|\alpha_i| \leq 1, \quad \text{та} \quad z_i = c_i - a_i \cdot \alpha_i \neq 0, \quad i = \overline{1, N}.$$

**Задача 8.** Довести теорему про стійкість методу прогонки.

### 3.5 Обумовленість систем лінійних алгебраїчних рівнянь

Нехай задано СЛАР

$$A\vec{x} = \vec{b}. \quad (3.23)$$

Припустимо, що матриця і права частина системи задані неточно і фактично розв'язуємо систему

$$B\vec{y} = \vec{h}. \quad (3.24)$$

де  $B = A + C$ ,  $\vec{h} = \vec{b} + \vec{\eta}$ ,  $\vec{y} = \vec{x} + \vec{z}$ .

Малість детермінанту  $\det A \ll 1$  не є необхідною умовою різкого збільшення похибки. Це ілюструє наступний приклад:

$$A = \text{diag}(\varepsilon), \quad a_{i,j} = \varepsilon \Delta_{i,j}.$$

Тоді  $\det A = \varepsilon^n \ll 1$ , але  $x_i = \frac{b_i}{\varepsilon}$ . Тому  $\Delta x_i = \frac{\Delta b_i}{\varepsilon}$ .

Оцінимо похибку розв'язку. Підставивши значення  $B, \vec{h}$ , та  $\vec{z} = \vec{y} - \vec{x}$ , отримаємо:

$$(A + C)(\vec{x} + \vec{z}) = \vec{b} + \vec{\eta}.$$

Віднімемо від цієї рівності (3.24) у вигляді  $A\vec{z} + C\vec{x} + C\vec{z} = \vec{\eta}$ . Тоді

$$A\vec{z} = \vec{\eta} - C\vec{x} - C\vec{z}, \quad \vec{z} = A^{-1}(\vec{\eta} - C\vec{x} - C\vec{z}).$$

Введемо норми векторів:  $\|\vec{z}\|$ :

$$\|\vec{z}\|_1 = \sum_{i=1}^n |z_i|, \quad \|\vec{z}\|_2 = \left( \sum_{i=1}^n |z_i|^2 \right)^{1/2}, \quad \|\vec{z}\|_\infty = \max_i |z_i|.$$

Норми матриці, що відповідають нормам вектора, тобто

$$\|A\|_m = \sup_{\|\vec{x}\|_m \neq 0} \frac{\|A\vec{x}\|_m}{\|\vec{x}\|_m}, \quad m = 1, 2, \infty.$$

такі:

$$\|A\|_1 = \max_j \sum_{i=1}^n |a_{i,j}|, \quad \|A\|_2 = \max_i \sqrt{\lambda_i(A^T A)}, \quad \|A\|_\infty = \max_i \sum_{j=1}^n |a_{i,j}|,$$

де  $\lambda_i(B)$  – власні значення матриці  $B$ .

Позначимо  $\delta(\vec{x}) = \frac{\|\vec{z}\|}{\|\vec{x}\|}$ ,  $\delta(\vec{b}) = \frac{\|\vec{\eta}\|}{\|\vec{b}\|}$ ,  $\delta(A) = \frac{\|C\|}{\|A\|}$  – відносні похибки  $\vec{x}$ ,  $\vec{b}$ ,  $A$ , де  $\|\cdot\|_k$  – одна з введених вище норм.

Для характеристики зв'язку між похибками правої частини та розв'язку вводять поняття обумовленості матриці системи.

Число обумовленості матриці  $A$  –  $\text{cond}(A) = \|A\| \cdot \|A^{-1}\|$ .

**Теорема 3.5.1.** Якщо  $\exists A^{-1}$  та  $\|A^{-1}\| \cdot \|C\| < 1$ , то

$$\delta(\vec{x}) \leq \frac{\text{cond}(A)}{1 - \text{cond}(A) \cdot \delta(A)} (\delta(A) + \delta(\vec{b})), \quad (3.25)$$

де  $\text{cond}(A)$  – число обумовленості.

*Доведення.*

$$A\vec{z} = \vec{\eta} - C\vec{x} - C\vec{z}, \quad \vec{z} = A^{-1}\vec{\eta} - A^{-1}C\vec{x} - A^{-1}C\vec{z}$$

$$\begin{aligned}
\|\vec{z}\| &\leq \|A^{-1}\vec{\eta}\| + \|A^{-1}C\vec{x}\| + \|A^{-1}C\vec{z}\| \leq \\
&\leq \|A^{-1}\| \cdot \|\vec{\eta}\| + \|A^{-1}\| \cdot \|C\| \cdot \|\vec{x}\| + \|A^{-1}\| \cdot \|C\| \cdot \|\vec{z}\|. \\
\|\vec{z}\| &\leq \frac{\|A^{-1}\| \cdot (\|\vec{\eta}\| + \|C\| \cdot \|\vec{x}\|)}{1 - \|A^{-1}\| \cdot \|C\|}
\end{aligned}$$

Оцінка похибки

$$\begin{aligned}
\delta(\vec{x}) &\leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\| \cdot \|C\|} \left( \frac{\|\vec{\eta}\|}{\|\vec{x}\|} + \|C\| \right) = \\
&= \frac{\|A^{-1}\| \cdot \|A\|}{1 - \|A^{-1}\| \cdot \|A\| \cdot \frac{\|C\|}{\|A\|}} \left( \frac{\|\vec{\eta}\|}{\|A\| \cdot \|\vec{x}\|} + \delta(A) \right) \leq \\
&\leq \frac{\text{cond}(A)}{1 - \text{cond}(A) \cdot \delta(A)} \left( \frac{\|\vec{\eta}\|}{\|\vec{x}\|} + \delta(A) \right).
\end{aligned}$$

□

**Наслідок.** Якщо  $C \equiv 0$ , то  $\delta(\vec{x}) \leq \text{cond}(A) \cdot \delta(\vec{b})$ .

**Властивості  $\text{cond}(A)$ :**

1.  $\text{cond}(A) \geq 1$ ;
2.  $\text{cond}(A) \geq \frac{\max |\lambda_i(A)|}{\min |\lambda_i(A)|}$ ;
3.  $\text{cond}(AB) \leq \text{cond}(A) \cdot \text{cond}(B)$ ;
4.  $A^T = A^{-1} \Rightarrow \text{cond}(A) = 1$ .

Друга властивість має місце оскільки довільна норма матриці не менше її найбільшого за модулем власного значення. Значить  $\|A\| \geq \max |\lambda_A|$ . Оскільки власні значення матриць  $A^{-1}$  та  $A$  взаємно обернені, то

$$\|A^{-1}\| \geq \max \frac{1}{|\lambda_A|} = \frac{1}{\min |\lambda_A|}.$$

Якщо  $1 \ll \text{cond}(A)$ , то система називається *погано обумовленою*.

Оцінка впливу похибок заокруглення при обчисленні розв'язку СЛАР така (Дж. Уїлкінсон):  $\delta(A) = O(n\beta^{-t})$ ,  $\delta(\vec{b}) = O(\beta^{-t})$ , де  $\beta$  – розрядність ЕОМ,  $t$  – кількість розрядів, що відводиться під мантису числа. З оцінки (3.25) витікає:  $\delta(\vec{x}) = \text{cond}(A) \cdot O(n\beta^{-t})$ . Висновок: найпростіший спосіб підвищити точність обчислення розв'язку погано обумовленої СЛАР –

збільшити розрядність ЕОМ при обчисленнях. Інші способи пов'язані з розглядом цієї СЛАР як некоректної задачі із застосуванням відповідних методів її розв'язання.

Приклад погано обумовленої системи – системи з матрицею Гільберта

$$H_n = \left( \frac{1}{i+j-1} \right)_{i,j=1}^n, \quad \text{наприклад } \text{cond}(H_8) \approx 10^9.$$

## 4 Ітераційні методи для систем

### 4.1 Ітераційні методи розв'язання СЛАР

Систему

$$A\vec{x} = \vec{b} \quad (4.1)$$

зводимо до вигляду

$$\vec{x} = B\vec{x} + \vec{f}. \quad (4.2)$$

Будь яка система

$$\vec{x} = \vec{x} - C \cdot (A\vec{x} - \vec{b}) \quad (4.3)$$

має вигляд (4.2) і при  $\det C \neq 0$  еквівалентна системі (4.1). Наприклад, для  $C = \tau \cdot E$ :

$$\vec{x} = \vec{x} - \tau \cdot (A\vec{x} - \vec{b}). \quad (4.4)$$

#### 4.1.1 Метод простої ітерації

Цей метод застосовується до рівняння (4.2)

$$\vec{x}^{(k+1)} = B\vec{x}^{(k)} + \vec{f}, \quad (4.5)$$

де  $\vec{x}^{(0)}$  – початкове наближення, задано.

Ітераційний процес збігається, тобто  $\|\vec{x}^{(k)} - \vec{x}\| \xrightarrow[k \rightarrow \infty]{} 0$ , якщо

$$\|B\| \leq q < 1 \quad (4.6)$$

При цьому має місце оцінка

$$\|\vec{x}^{(n)} - \vec{x}\| \leq \frac{q^n}{1-q} \cdot \|\vec{x}^{(1)} - \vec{x}^{(0)}\|. \quad (4.7)$$

#### 4.1.2 Метод Якобі

Припустимо  $\forall i: a_{i,i} \neq 0$ . Зведемо систему (4.1) до вигляду

$$x_i = - \sum_{j=1}^{i-1} \frac{a_{i,j}}{a_{i,i}} \cdot x_j - \sum_{j=i+1}^n \frac{a_{i,j}}{a_{i,i}} \cdot x_j + \frac{b_i}{a_{i,i}}, \quad i = \overline{1, n}.$$

Ітераційний процес запишемо у вигляді

$$x_i^{(k+1)} = - \sum_{j=1}^{i-1} \frac{a_{i,j}}{a_{i,i}} \cdot x_j^{(k)} - \sum_{j=i+1}^n \frac{a_{i,j}}{a_{i,i}} \cdot x_j^{(k)} + \frac{b_i}{a_{i,i}}, \quad k = 0, 1, \dots, \quad i = \overline{1, n}. \quad (4.8)$$

Ітераційний процес збігається до розв'язку, якщо виконується умова

$$\forall i : \sum_{\substack{j=1 \\ i \neq j}}^n |a_{i,j}| \leq |a_{i,i}|.$$

Це умова діагональної переваги матриці  $A$ . Якщо ж

$$\forall i : \sum_{\substack{j=1 \\ i \neq j}}^n |a_{i,j}| \leq q \cdot |a_{i,i}|, \quad 0 \leq q < 1. \quad (4.9)$$

то має місце оцінка точності:

$$\|\vec{x}^{(n)} - \vec{x}\| \leq \frac{q^n}{1 - q} \cdot \|\vec{x}^{(0)} - \vec{x}\|.$$

#### 4.1.3 Метод Зейделя

В компонентному вигляді ітераційний метод Зейделя записується так:

$$x_i^{(k+1)} = - \sum_{j=1}^{i-1} \frac{a_{i,j}}{a_{i,i}} \cdot x_j^{(k+1)} - \sum_{j=i+1}^n \frac{a_{i,j}}{a_{i,i}} \cdot x_j^{(k)} + \frac{b_i}{a_{i,i}}, \quad k = 0, 1, \dots, \quad i = \overline{1, n}. \quad (4.10)$$

На відміну від методу Якобі на  $k$ -му-кроці попередні компоненти розв'язку беруться з  $k + 1$ -ої ітерації.

Достатня умова збіжності методу Зейделя –  $A^T = A > 0$ .

#### 4.1.4 Матрична інтерпретація методів Якобі і Зейделя

Подамо матрицю  $A$  у вигляді

$$A = A_1 + D + A_2,$$

де  $A_1$  – нижній трикутник матриці  $A$ ,  $A_2$  – верхній трикутник матриці  $A$ ,  $D$  – її діагональ. Тоді систему (4.1) запишемо у вигляді

$$D\vec{x} = A_1\vec{x} + A_2\vec{x} + \vec{b},$$

або

$$\vec{x} = D^{-1}A_1\vec{x} + D^{-1}A_2\vec{x} + D^{-1}\vec{b},$$

Матричний запис методу Якобі:

$$\vec{x}^{(k+1)} = D^{-1}A_1\vec{x}^{(k)} + D^{-1}A_2\vec{x}^{(k)} + D^{-1}\vec{b},$$

методу Зейделя:

$$\vec{x}^{(k+1)} = D^{-1}A_1\vec{x}^{(k+1)} + D^{-1}A_2\vec{x}^{(k)} + D^{-1}\vec{b},$$

Необхідна і достатня умова збіжності методу Якобі: всі корені рівняння

$$\det(D + \lambda(A_1 + A_2)) = 0$$

по модулю більше 1.

Необхідна і достатня умова збіжності методу Зейделя: всі корені рівняння

$$\det(A_1 + D + \lambda A_2) = 0$$

по модулю більше 1.

#### 4.1.5 Однокрокові (двошарові) ітераційні методи

Канонічною формою однокрокового ітераційного методу розв'язку СЛАР є його запис у вигляді

$$B_k \frac{\vec{x}^{(k+1)} - \vec{x}^{(k)}}{\tau_{k+1}} + A\vec{x}^{(k)} = \vec{b}, \quad (4.11)$$

Тут  $\{B_k\}$  – послідовність матриць (пере-обумовлюючі матриці), що задають ітераційний метод на кожному кроці;  $\{\tau_{k+1}\}$  – ітераційні параметри.

Якщо  $B_k = E$ , то ітераційний процес називається *явним*

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - \tau_{k+1} (A\vec{x}^{(k)} + \vec{b}).$$

Якщо  $B_k \neq E$ , то ітераційний процес називається *неявним*

$$B_k \vec{x}^{(k+1)} = F^k.$$

У цьому випадку на кожній ітерації необхідно розв'язувати СЛАР.

Якщо  $\tau_{k+1} \equiv \tau$ ,  $B_k \equiv B$ , то ітераційний процес називається *стаціонарним*; інакше – *нестационарним*.

Методам, що розглянуті вище відповідають:

- методу простої ітерації:  $B_k = E$ ,  $\tau_{k+1} = \tau$ ;
- методу Якобі:  $B_k = D$ ,  $\tau_{k+1} = 1$ ;
- методу Зейделя:  $B_k = D + A_1$ ,  $\tau_{k+1} = 1$ .

#### 4.1.6 Збіжності стаціонарних ітераційних процесів у випадку симетричних матриць

Розглянемо випадок симетричних матриць  $A^T = A$  і стаціонарний ітераційний процес  $B_k \equiv E$ ,  $\tau_{k+1} \equiv \tau$ .

Нехай для  $A$  справедливі нерівності

$$\gamma_1 E \leq A \leq \gamma_2 E, \quad \gamma_1, \gamma_2 > 0. \quad (4.12)$$

Тоді при виборі  $\tau = \tau_0 = \frac{2}{\gamma_1 + \gamma_2}$  ітераційний процес збігається. Найбільш точним значенням  $\gamma_1, \gamma_2$  при яких виконуються обмеження (4.12) є  $\gamma_1 = \min \lambda_i(A)$ ,  $\gamma_2 = \max \lambda_i(A)$ . Тоді

$$q = q_0 = \frac{\gamma_2 - \gamma_1}{\gamma_2 + \gamma_1} = \frac{1 - \xi}{1 + \xi}, \quad \xi = \frac{\gamma_1}{\gamma_2}.$$

(Зауважимо, що аналогічно обчислюється  $q$  і для методу релаксації розв'язання нелінійних рівнянь, де  $\gamma_1 = m = \min |f'(x)|$ ,  $\gamma_2 = M_1 = \max |f'(x)|$ ) і справедлива оцінка

$$\|\vec{x}^{(n)} - \vec{x}\| \leq \frac{q^n}{1 - q} \cdot \|\vec{x}^{(0)} - \vec{x}\|.$$



Явний метод з багатьма параметрами  $\{\tau_k\}$ :

$$B \equiv E, \quad \{\tau_k\} : \min_{\tau} q(\tau), \quad n = n(\varepsilon) \rightarrow \min,$$

які обчислюються за допомогою нулів багаточлена Чебишова, називаються ітераційним методом з чебишевським набором параметрів.

#### 4.1.7 Метод верхньої релаксації

Узагальненням методу Зейделя є метод верхньої релаксації:

$$(D + \omega A_1) \cdot \frac{\vec{x}^{(k+1)} + \vec{x}^{(k)}}{\omega} + A\vec{x}^{(k)} = \vec{b},$$

де  $D$  – діагональна матриця з елементами  $a_{i,i}$  по діагоналі.  $\omega > 0$  – заданий числовий параметр.

Тепер  $B = D + \omega A_1$ ,  $\tau = \omega$ . Якщо  $A^T = A > 0$ , то метод верхньої релаксації збігається при умові  $0 < \omega < 2$ . Параметр підбирається експериментально з умови мінімальної кількості ітерацій.

#### 4.1.8 Методи варіаційного типу

До цих методів відносяться: метод мінімальних нев'язок, метод мінімальних поправок, метод найшвидшого спуску, метод спряжених градієнтів. Вони дозволяють обчислювати наближення без використання апріорної інформації про  $\gamma_1$ ,  $\gamma_2$  в (4.12).

Нехай  $B = E$ . Для методу мінімальних нев'язок параметри  $\tau_{k+1}$  обчислюються з умови

$$\|\vec{r}^{(k+1)}\|^2 = \|\vec{r}^{(k)}\|^2 - 2\tau_{k+1} \cdot (\vec{r}^{(k)}, A\vec{r}^{(k)}) + \tau_{k+1}^2 \cdot \|A\vec{r}^{(k)}\|^2 \rightarrow \min.$$

Тому

$$\tau_{k+1} = \frac{(\vec{r}^{(k)}, A\vec{r}^{(k)})}{\|\vec{r}^{(k)}\|^2},$$

де  $\vec{r}^{(k)} = A\vec{x}^{(k)} - \vec{b}$  – нев'язка.

Умова для завершення ітераційного процесу:

$$\|\vec{r}^{(n)}\| < \varepsilon.$$

Швидкість збіжності цього методу співпадає із швидкістю методу простої ітерації з одним оптимальним параметром  $\tau_0 = \frac{2}{\gamma_1 + \gamma_2}$ .

Аналогічно будуються методи з  $B \neq E$ . Матриця  $B$  називається переобумовлювачем і дозволяє підвищити швидкість збіжності ітераційного процесу. Його вибирають з умов

1. легко розв'язувати СЛАР  $B\vec{x}^{(k)} = F_k$  (діагональний, трикутний, добуток трикутних та інше);
2. зменшення числа обумовленості матриці  $B^{-1}A$  у порівнянні з  $A$ .

## 4.2 Методи розв'язання нелінійних систем

Розглянемо систему рівнянь

$$\begin{cases} f_1(x_1, \dots, x_n) = 0, \\ \dots \\ f_n(x_1, \dots, x_n) = 0. \end{cases}$$

Перепишемо її у векторному вигляді:

$$\vec{f}(\vec{x}) = 0. \quad (4.13)$$

### 4.2.1 Метод простої ітерації

В цьому методі рівняння (4.13) зводиться до еквівалентного вигляду

$$\vec{x} = \vec{\Phi}(\vec{x}). \quad (4.14)$$

Ітераційний процес представимо у вигляді:

$$\vec{x}^{(k+1)} = \vec{\Phi}(\vec{x}^{(k)}). \quad (4.15)$$

початкове наближення  $\vec{x}^{(0)}$  – задано.

Нехай оператор  $\vec{\Phi}$  визначений на множині  $H$ . За теоремою про стискуючі відображення ітераційний процес (4.15) сходиться, якщо виконується умова

$$\|\vec{\Phi}(\vec{x}) - \vec{\Phi}(\vec{y})\| \leq q \cdot \|\vec{x} - \vec{y}\|, \quad 0 < q < 1, \quad (4.16)$$

або

$$\|\vec{\Phi}'(\vec{x})\| \leq q < 1, \quad (4.17)$$

де  $\vec{x} \in U_r$ ,  $\vec{\Phi}'(\vec{x}) = \left( \frac{\partial \varphi_i}{\partial x_j} \right)_{i,j=1}^n$ . Для похибки справедлива оцінка

$$\|\vec{x}^{(m)} - \vec{x}\| \leq \frac{q^n}{1-q} \cdot \|\vec{x}^{(0)} - \vec{x}\|.$$

Частинним випадком методу простої ітерації є метод релаксації для рівняння (4.13):

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - \tau \cdot \vec{F}(\vec{x}^{(k)}),$$

де  $\tau < \frac{2}{\|\vec{F}'(\vec{x})\|}$ .

#### 4.2.2 Метод Ньютона

Розглянемо рівняння

$$\vec{F}(\vec{x}) = 0.$$

Представимо його у вигляді

$$\vec{F}(\vec{x}^{(k)}) + \vec{F}'(\vec{\xi}^{(k)}) \cdot (\vec{x} - \vec{x}^{(k)}) = 0, \quad (4.18)$$

де  $\vec{\xi}^{(k)} = \vec{x}^{(k)} + \theta_k \cdot (\vec{x}^{(k)} - \vec{x})$ ,  $0 < \theta_k < 1$ . Тут  $\vec{F}'(\vec{x}) = \left( \frac{\partial f_i}{\partial x_j} \right)_{i,j=1}^n$  – матриця Якобі для  $\vec{F}(\vec{x})$ . Можемо наближено вважати  $\vec{\xi}^{(k)} \approx \vec{x}^{(k)}$ . Тоді з (4.18) матимемо

$$\vec{F}(\vec{x}^{(k)}) + \vec{F}'(\vec{x}^{(k)}) \cdot (\vec{x}^{(k+1)} - \vec{x}^{(k)}) = 0. \quad (4.19)$$

Ітераційний процес представимо у вигляді:

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - \vec{F}'(\vec{x}^{(k)})^{-1} \cdot \vec{F}(\vec{x}^{(k)}). \quad (4.20)$$

Для реалізації методу Ньютона потрібно, щоб існувала обернена матриця

$$\vec{F}'(\vec{x}^{(k)})^{-1}.$$

Можна не шукати обернену матрицю, а розв'язувати на кожній ітерації СЛАР

$$\begin{cases} A_k \vec{z}^{(k)} = \vec{F}(\vec{x}^{(k)}), \\ \vec{x}^{(k+1)} = \vec{x}^{(k)} - \vec{z}^{(k)}, \end{cases} \quad k = 0, 1, 2, \dots \quad (4.21)$$

де  $\vec{x}^{(0)}$  – задано, а матриця  $A_k = \vec{F}'(\vec{x}^{(k)})$ .

Метод має квадратичну збіжність, якщо добре вибрано початкове наближення. Складність методу (при умові використання методу Гаусса розв'язання СЛАР (4.21) на кожній ітерації  $Q_n = \frac{2}{3}n^3 + O(n^2)$ , де  $n$  – розмірність системи (4.13).

### 4.2.3 Модифікований метод Ньютона

Ітераційний процес має вигляд :

$$\vec{x}^{(k+1)} = \vec{x}^{(k)} - \vec{F}'(\vec{x}^{(0)})^{-1} \cdot \vec{F}(\vec{x}^{(k)}).$$

Тепер обернена матриця обчислюється тільки на нульовій ітерації. На інших – обчислення нового наближення зводиться до множення матриці  $A_0 = \vec{F}'(\vec{x}^{(0)})^{-1}$  на вектор  $\vec{F}(\vec{x}^{(k)})$  та додавання до  $\vec{x}^{(k)}$ .

Запишемо метод у вигляді системи лінійних рівнянь (аналог (4.21))

$$\begin{cases} A_0 \vec{z}^{(k)} = \vec{F}(\vec{x}^{(k)}), \\ \vec{x}^{(k+1)} = \vec{x}^{(k)} - \vec{z}^{(k)}, \end{cases} \quad k = 0, 1, 2, \dots \quad (4.22)$$

Оскільки матриця  $A_0$  розкладається на трикутні (або обертається) один раз, то складність цього методу на одній ітерації (окрім нульової)  $Q_n = O(n^2)$ . Але цей метод має лінійну швидкість збіжності.

Можливе циклічне застосування модифікованого методу Ньютона, тобто коли обернену матрицю похідних шукаємо та обертаємо через певне число кроків ітераційного процесу.

**Задача 9.** Побудувати аналог методу січних для систем нелінійних рівнянь.

## 5 Проблема власних значень

Нехай задано матрицю  $A$  розмірності  $n \times n$ . Тоді задача на власні значення ставиться так: знайти число  $\lambda$  та вектор  $\vec{x} \neq \vec{0}$ , що задовольняють рівнянню

$$A\vec{x} = \lambda\vec{x}. \quad (5.1)$$

$\lambda$  називається власним значенням  $A$ , а  $\vec{x}$  – власним вектором. З (5.1)

$$\det(A - \lambda E) \equiv P_n(\lambda) \equiv (-1)^n \lambda^n + a_n \lambda^{n-1} + \dots + a_0 = 0.$$

Тут  $P_n(\lambda)$  – характеристичний багаточлен.

Для розв'язання багатьох задач механіки, фізики, хімії потрібне знаходження всіх власних значень  $\lambda_i$ ,  $i = \overline{1, n}$ , а іноді й всіх власних векторів

$\vec{x}_i$ , що відповідають  $\lambda_i$ . Цю задачу називають повною проблемою власних значень.

В багатьох випадках потрібно знайти лише максимальне або мінімальне за модулем власне значення матриці. При дослідженні стійкості коливальних процесів іноді потрібно знайти два максимальних за модулем власних значення матриці.

Останні дві задачі називають *частковими проблемами власних значень*.

## 5.1 Степеневий метод

1. *Знаходження*  $\lambda_{\max} : |\lambda_1| \equiv \lambda_{\max} > |\lambda_2| \geq |\lambda_3| \geq \dots$

Нехай  $\vec{x}^{(0)}$  – заданий вектор, будемо послідовно обчислювати вектори

$$\vec{x}^{(k+1)} = A\vec{x}^{(k)}, \quad k = 0, 1, \dots \quad (5.2)$$

Тоді  $\vec{x}^{(k)} = A^k \vec{x}^{(0)}$ . Нехай  $\{\vec{e}_i\}_{i=1}^n$  – система власних векторів. Представимо  $\vec{x}^{(0)}$  у вигляді:

$$\vec{x}^{(0)} = \sum_{i=1}^n c_i \vec{e}_i.$$

Оскільки  $A\vec{e}_i = \lambda_i \vec{e}_i$ , то  $\vec{x}^{(k)} = \sum_{i=1}^n c_i \lambda_i^k \vec{e}_i$ . При великих  $k$ :  $\vec{x}^{(k)} \approx c_1 \lambda_1^k \vec{e}_1$ .

Тому

$$\mu_1^{(k)} = \frac{\vec{x}_m^{(k+1)}}{\vec{x}_m^{(k)}} = \lambda_1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right).$$

Значить  $\mu_1^{(k)} \xrightarrow[k \rightarrow \infty]{} \lambda_1$ .

Якщо матриця  $A = A^T$  симетрична, то існує ортонормована система векторів  $(\vec{e}_i, \vec{e}_j) = \delta_{i,j}$ . Тому

$$\begin{aligned}\mu_1^{(k)} &= \frac{(\vec{x}^{(k+1)}, \vec{x}^{(k)})}{(\vec{x}^{(k)}, \vec{x}^{(k)})} = \frac{\left( \sum_{i=1}^n c_i \lambda_i^{k+1} \vec{e}_i, \sum_{j=1}^n c_j \lambda_j^k \vec{e}_j \right)}{\left( \sum_{i=1}^n c_i \lambda_i^k \vec{e}_i, \sum_{j=1}^n c_j \lambda_j^k \vec{e}_j \right)} = \frac{\sum_i c_i^2 \lambda_i^{2k+1}}{\sum_i c_i^2 \lambda_i^{2k}} = \\ &= \frac{c_1^2 \lambda_1^{2k+1} + c_2^2 \lambda_2^{2k+1} + \dots}{c_1^2 \lambda_1^{2k} + c_2^2 \lambda_2^{2k} + \dots} = \lambda_1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right) \xrightarrow{k \rightarrow \infty} \lambda_1.\end{aligned}$$

Це означає збіжність до максимального за модулем власного значення з квадратичною швидкістю.

Якщо  $\lambda_1 > 1$ , то при проведенні ітерацій відбувається зріст компонент вектора  $\vec{x}^{(k)}$ , що приводить до “переповнення” (overflow). Якщо ж  $\lambda_1 < 1$ , то це приводить до зменшення компонент (underflow). Позбутися негативу такого явища можна нормуючи вектори  $\vec{x}^{(k)}$  на кожній ітерації.

**Алгоритм** степеневому методу знаходження максимального за модулем власного значення з точністю  $\varepsilon$  виглядає так:

- (а)  $\vec{x}^{(0)} \rightarrow \vec{e}_0 = \frac{\vec{x}^{(0)}}{\|\vec{x}^{(0)}\|}$ ;
- (б)  $\vec{x}^{(k+1)} = A\vec{x}^{(k)}$ ,  $\mu_1^{(k)} = (\vec{x}^{(k+1)}, \vec{e}^{(k)})$ ,  $\vec{e}^{(k+1)} = \frac{\vec{x}^{(k+1)}}{\|\vec{x}^{(k+1)}\|}$ ,  $k = 0, 1, \dots$ ;
- (в)  $|\mu_1^{(k+1)} - \mu_1^{(k)}| \geq \varepsilon$  goto (б);
- (г)  $\lambda_1 \approx \mu_1^{(k+1)}$ .

За цим алгоритмом для симетричної матриці  $A^T = A$  швидкість прямування  $\mu_1^{(k)}$  до  $\lambda_{\max}$  – квадратична.

2. *Знаходження  $\lambda_2$  :  $|\lambda_1| > |\lambda_2| > |\lambda_3| \geq \dots$*  Нехай  $\lambda_1, \vec{e}_1$  відомі.

**Задача 10.** Довести, що якщо  $|\lambda_1| > |\lambda_2| > |\lambda_3| \geq \dots$ , то

$$\mu_2^{(k)} = \frac{\vec{x}_m^{(k+1)} - \lambda_1 \vec{x}_m^{(k)}}{\vec{x}_m^{(k)} - \lambda_1 \vec{x}_m^{(k-1)}} \xrightarrow{k \rightarrow \infty} \lambda_2,$$

де  $\vec{x}^{(k+1)} = A\vec{x}^{(k)}$ ,  $x_m^{(k)}$  –  $m$ -та компонента  $\vec{x}^{(k)}$ .

**Задача 11.** Побудувати алгоритм обчислення  $\lambda_2, \vec{e}_2$ , використовуючи нормування векторів та скалярні добутки для обчислення  $\mu_2^{(k)}$ .

3. Знаходження мінімального власного числа  $\lambda_{\min}(A) = \min_i |\lambda_i(A)|$ .

Припустимо, що  $\forall i: \lambda_i(A) > 0$  та відоме  $\lambda_{\max}$ . Розглянемо матрицю  $B = \lambda_{\max} \cdot E - A$ . Маємо

$$\forall i: \lambda_i(B) = \lambda_{\max} - \lambda_i(A).$$

Тому  $\max_i \lambda_i(B) = \lambda_{\max} - \min_i \lambda_i(A)$ . Звідси  $\lambda_{\min}(A) = \lambda_{\max}(A) - \lambda_{\max}(B)$ .

Якщо  $\exists i: \lambda_i(A) < 0$ , то будемо матрицю  $\bar{A} = \sigma \cdot E + A$ ,  $\sigma > 0$ ,  $\bar{A} > 0$  і для неї попередній розгляд дає необхідний результат. Замість  $\lambda_{\max}$  в матриці  $B$  можна використовувати  $\|A\|$ .

Ще один спосіб обчислення мінімального власного значення полягає в використанні обернених ітерацій:

$$A\vec{x}^{(k+1)} = \vec{x}^k, \quad k = 0, 1, \dots \quad (5.3)$$

Але цей метод вимагає більшої кількості арифметичних операцій: складність методу на основі формули (5.2)  $Q = O(n^2)$ , а на основі (5.3) –  $Q = O(n^3)$ , оскільки треба розв'язувати СЛАР, але збігається метод (5.3) швидше.

## 5.2 Ітераційний метод обертання

Цей метод розв'язання повної проблеми власних значень для симетричних матриць  $A^T = A$ . Існує матриця  $U$ , що приводить матрицю  $A$  до діагонального виду:

$$A = U\Lambda U^T, \quad (5.4)$$

де  $\Lambda$  – діагональна матриця, по діагоналі якої стоять власні значення  $\lambda_i$ ;  $U$  – унітарна матриця, тобто:  $U^{-1} = U^T$ .

З (5.4) маємо

$$\Lambda = U^T \Lambda U, \quad (5.5)$$

Нехай  $\exists \tilde{U}$  – матриця, така що  $\tilde{\Lambda} = \tilde{U}^T A \tilde{U}$  і  $\tilde{\Lambda} = \left( \tilde{\lambda}_{i,j} \right)_{i,j=1}^n$ ,  $|\tilde{\lambda}_{i,j}| < \delta \ll 1$ ,  $i \neq j$ .

Тоді діагональні елементи мало відрізняються від власних значень

$$|\tilde{\lambda}_{i,i} - \lambda_i(A)| < \varepsilon = \varepsilon(\delta).$$

Введемо

$$t(A) = \sum_{\substack{i,j=1 \\ i \neq j}}^n a_{i,j}^2.$$

З малості величини  $t(A)$  витікає, що діагональні елементи малі. По  $A = A_0$  за допомогою матриць обертання

$$U_k = \begin{pmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cos \varphi & \cdots & -\sin \varphi & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & \sin \varphi & \cdots & \cos \varphi & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{pmatrix}.$$

що повертають систему векторів на кут  $\varphi$ , побудуємо послідовність  $\{A_k\}$  таку, що  $A_k \rightarrow \Lambda$  при  $k \rightarrow \infty$ .

**Задача 12.** Показати, що матриця обертання  $U_k$  є унітарною:  $U_k^{-1} = U_k^T$ .

Послідовно будуюмо:

$$A_{k+1} = U_k^T A_k U_k, \quad (5.6)$$

Процес (5.6) називається *монотонним*, якщо:  $t(A_{k+1}) < t(A_k)$ .

**Задача 13.** Довести, що для матриці (5.6) виконується:

$$a_{i,j}^{(k+1)} = a_{i,j}^{(k)} \cos(2\varphi) + \frac{1}{2} (a_{j,j}^{(k)} - a_{i,i}^{(k)}) \sin(2\varphi), \quad (5.7)$$

Показати, що  $t(A_{k+1}) = t(A_k) - 2 \left( a_{i,j}^{(k)} \right)^2$ , якщо вибирати  $\varphi$  з умови  $a_{i,j}^{(k+1)} = 0$ .

Звідси

$$\varphi = \varphi_k = \frac{1}{2} \arctan(p^{(k)}), \quad p^{(k)} = \frac{2a_{i,j}^{(k)}}{a_{i,i}^{(k)} - a_{j,j}^{(k)}},$$

де  $\left| a_{i,j}^{(k)} \right| = \max_{m \neq l} \left| a_{m,l}^{(k)} \right|$ . Тоді  $t(A_k) \xrightarrow[k \rightarrow \infty]{} 0$ . Чим більше  $n$  тим більше ітерацій необхідно для зведення  $A$  до  $\Lambda$ .

Якщо матриця несиметрична, то застосовують  $QR$ ,  $QL$  методи.



## 6 Інтерполювання функцій

### 6.1 Постановка задачі інтерполювання

Нехай функція  $f(x) \in C([a, b])$  задана своїми значеннями  $y_i = f(x_i)$ ,  $x_i \in [a, b]$ ,  $i = \overline{0, n}$ , причому при  $x_i \neq x_j$  для  $i \neq j$ .

Функція  $\Phi(x)$  називається *інтерполюючою* для  $f(x)$  на сітці  $\{x_i\}_{i=0}^n$ , якщо  $\Phi(x_i) = y_i$ ,  $i = \overline{0, n}$ .

Задача інтерполювання функції має не єдиний розв'язок. Виберемо систему лінійно незалежних функцій  $\{\varphi_k(x)\}_{k=0}^n$ ,  $\varphi_k(x) \in C([a, b])$  і побудуємо лінійну комбінацію

$$\Phi(x) = \Phi_n(x) = \sum_{k=0}^n c_k \cdot \varphi_k(x), \quad (6.1)$$

яка називається *узагальненим багаточленом*. Умови інтерполювання дають СЛАР

$$\sum_{k=0}^n c_k \cdot \varphi_k(x_i) = y_i, \quad i = \overline{1, n} \quad (6.2)$$

розв'язком якої є  $\vec{c} = (c_0, \dots, c_n)$ . Якщо

$$D(x_0, \dots, x_n) = \begin{vmatrix} \varphi_0(x_0) & \cdots & \varphi_n(x_0) \\ \vdots & \ddots & \vdots \\ \varphi_0(x_n) & \cdots & \varphi_n(x_n) \end{vmatrix} \neq 0,$$

то система (6.2) має єдиний розв'язок.

Система функцій  $\{\varphi_k(x)\}$ ,  $k = \overline{0, n}$  називається системою Чебишова, якщо  $\forall \{x_i\}_{i=0}^n$ ,  $x_i \in [a, b]$ ,  $D(x_0, \dots, x_n) \neq 0$ ,  $i = \overline{0, n}$ ,  $\forall i \neq j: x_i \neq x_j$ .

**Приклади** систем Чебишова:

1.  $\varphi_k(x) = x^k$  – алгебраїчна система.

Визначник  $D(x_0, \dots, x_n) \neq 0$  є визначником Вандермонда:

$$D(x_0, \dots, x_n) = \begin{vmatrix} 1 & x_0 & \cdots & x_0^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{vmatrix} = \prod_{0 \leq k < m \leq n} (x_k - x_m) \neq 0,$$

2.  $\varphi_k(x) = L_k(x)$  – ортогональні багаточлени Лежандра;  $\varphi_k(x) = T_k(x)$  – ортогональні багаточлени Чебишова.
3.  $\varphi_k(x) : 1, \cos(x), \sin(x), \dots, \cos(nx), \sin(nx)$ .

Тоді

$$\Phi_n(x) = T_n(x) = a_0 + \sum_{k=1}^n (a_k \cdot \cos(kx) + b_k \cdot \sin(kx))$$

– тригонометричний багаточлен.

## 6.2 Інтерполяційна формула Лагранжа

Якщо  $\varphi_k(x) = x^k$ , то

$$\Phi_n(x) = P_n(x) = \sum_{k=0}^n c_k \cdot x^k.$$

Задача інтерполювання функції  $f(x)$  алгебраїчним, багаточленом полягає в знаходженні коефіцієнтів  $c_k$ ,  $k = \overline{0, n}$  для яких виконується умова  $f(x_i) = \varphi(x_i)$ ,  $i = \overline{0, n}$ .

Представимо інтерполяційний багаточлен у вигляді

$$P_n(x) = L_n(x) + \sum_{k=0}^n f(x_k) \cdot \Phi_k^{(n)}(x). \quad (6.3)$$

Тут  $L_n(x)$  – інтерполяційний поліном;  $\Phi_k^{(n)}(x)$  – поліноми  $n$ -го степеня, які називають множниками Лагранжа. З умови  $L_n(x_i) = f(x_i)$  випливає, що множник Лагранжа повинен задовольняти умови

$$\Phi_k^{(n)}(x_i) = \delta_{i,k}. \quad (6.4)$$

Так як  $\Phi_k^{(n)}(x)$  – багаточлен степеня  $n$ , то він має вигляд

$$\Phi_k^{(n)}(x) = A_k \cdot (x - x_0) \cdot \dots \cdot (x - x_{k-1}) \cdot (x - x_{k+1}) \cdot \dots \cdot (x - x_n),$$

де  $A_k$  – число. Знайдемо його з умови  $\Phi_k^{(n)}(x_k) = 1$ :

$$A_k = \frac{1}{(x_k - x_0) \cdot \dots \cdot (x_k - x_{k-1}) \cdot (x_k - x_{k+1}) \cdot \dots \cdot (x_k - x_n)}.$$

Таким чином багаточлен  $\Phi_k^{(n)}(x)$  мають вигляд:

$$\Phi_k^{(n)}(x) = \frac{(x - x_0) \cdot \dots \cdot (x - x_{k-1}) \cdot (x - x_{k+1}) \cdot \dots \cdot (x - x_n)}{(x_k - x_0) \cdot \dots \cdot (x_k - x_{k-1}) \cdot (x_k - x_{k+1}) \cdot \dots \cdot (x_k - x_n)} \quad (6.5)$$

Позначивши

$$\omega_n(x) = \prod_{i=0}^n (x - x_i),$$

маємо

$$\Phi_k^{(n)}(x) = \frac{\omega_n(x)}{(x - x_k) \cdot \omega'_n(x_k)}.$$

Остаточно формула Лагранжа має вигляд

$$L_n(x) = \sum_{k=0}^n f(x_k) \cdot \frac{\omega_n(x)}{(x - x_k) \cdot \omega'_n(x_k)} \quad (6.6)$$

### 6.3 Залишковий член інтерполяційного полінома

В заданих точках (точки інтерполювання) значення функції та полінома співпадають, але в інших точках в загальному випадку не співпадають. Отже доцільно розглянути питання про похибку інтерполювання.

Заміняючи функцію  $f(x)$  на  $L_n(x)$  ми допускаємо похибку  $r_n(x) = f(x) - L_n(x)$ . Це *залишковий член* інтерполювання.

З означення випливає, що,  $r_n(x_i) = 0$ ,  $x_i \in [a, b]$ . Оцінимо похибку у кожній точці  $x \in [a, b]$ . Введемо допоміжну функцію:

$$g(t) = f(t) - L_n(t) - K \cdot \omega_n(t), \quad t \in [a, b], \quad g(x_i) = 0, \quad i = \overline{0, n}.$$

Знайдемо таке  $K$ , щоб  $g(x) = 0$ , в деякій точці  $x \in [a, b]$ ,  $x \neq x_i$ ,  $i = \overline{0, n}$ . Легко бачити, що

$$K = \frac{f(x) - L_n(x)}{\omega_n(x)}.$$

Припустимо що  $f(x) \in C^{(n+1)}([a, b])$ , тоді  $g(t) \in C^{(n+1)}([a, b])$ . Функція  $g(t) = 0$  в  $(n+2)$ -х точках, а саме  $t = x$ ,  $t = x_i$ ,  $i = \overline{0, n}$ . З теореми Ролля випливає, що існує  $(n+1)$ -а точка, де  $g'(t_i) = 0$ ,  $i = \overline{0, n}$ . Продовжуючи цей процес, отримаємо, що існує хоча б одна  $\xi \in [a, b]$  така, що  $g^{(n+1)}(\xi) = 0$ . Так як

$$g^{(n+1)}(t) = f^{(n+1)}(t) - 0 - K \cdot (n+1)!,$$

то  $\exists \xi$ , що

$$g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - (n+1)! \cdot \frac{f(x) - L_n(x)}{\omega_n(x)} = 0.$$

Звідси

$$r_n(x) = f(x) - L_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \cdot \omega_n(x). \quad (6.7)$$

Оскільки  $\xi$  невідомо, то використовують *оцінку залишкового члена*:

$$|r_n(x)| = |f(x) - L_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \cdot |\omega_n(x)|, \quad (6.8)$$

де  $M_{n+1} = \sup_{x \in [a,b]} |f^{(n+1)}(x)|$ .

## 6.4 Багаточлени Чебишова. Мінімізація залишкового члена інтерполяційного полінома

Як вибрати вузли інтерполяції щоб похибка інтерполявання була мінімальною? Спочатку обґрунтуємо теоретичний апарат, завдяки якому будемо досліджувати це питання.

*Багаточленом Чебишова ( $n$ -того степеня, 1-го роду)* називається поліном, який задається такими рекурентними співвідношеннями

$$T_{n+1}(x) - 2x \cdot T_n(x) + T_{n-1}(x) = 0, \quad (6.9)$$

$$T_0(x) = 1, \quad T_1(x) = x. \quad (6.10)$$

Знайдемо явний вигляд багаточлена Чебишова. Будемо шукати розв'язок рівняння (6.9) у вигляді  $T_n(x) = q^n$ , де  $q = q(x)$ . Підставивши в (6.9), отримуємо характеристичне рівняння  $q^2 - 2xq + 1 = 0$ . Тоді при  $|x| \geq 1 \Rightarrow q_{1,2} = x \pm \sqrt{x^2 - 1}$ , а при  $|x| < 1 \Rightarrow q_{1,2} = \cos(\varphi) \pm i \sin(\varphi)$ ,  $\varphi = \arccos(x)$ .

Розглянемо обидва випадки детальніше:

1. при  $|x| \leq 1$ :  $T_n(x) = A \cdot \cos(n\varphi) + B \cdot \sin(n\varphi)$ . З (6.10) випливає  $A = 1$ ,  $B = 0$  і тому

$$T_n(x) = \cos(n \arccos(x)). \quad (6.11)$$

2. при  $|x| > 1$ :

$$T_n(x) = \frac{1}{2} \left( \left( x + \sqrt{x^2 - 1} \right)^n + \left( x - \sqrt{x^2 - 1} \right)^n \right). \quad (6.12)$$

Знайдемо нулі та екстремуми багаточлена Чебишова:  $T_n(x) = 0$ ,  $x \in [-1, 1]$ ,  $\cos(n \arccos(x)) = 0$ ,  $\arccos(x) = \frac{2k+1}{2n}\pi$ ,  $k = \overline{0, n-1}$ .

Отже нулі багаточлена Чебишова:

$$x_k = \cos\left(\frac{(2k+1)\pi}{2n}\right) \in [-1, 1], \quad k = \overline{0, n-1}.$$

Локальні екстремуми багаточлена Чебишова на  $x \in [-1, 1]$ :

$$x'_k = \cos\left(\frac{k\pi}{2n}\right), \quad k = \overline{0, n}.$$

Коефіцієнт при старшому члені багаточлена дорівнює  $2^{n-1}$ . Введемо нормований багаточлен Чебишова  $\bar{T}_n(x) = 2^{1-n}T_n(x) = x^n + \dots$ . Тоді

$$\|\bar{T}_n\|_{C([-1,1])} = \max_{x \in [-1,1]} |T_n(x)| = 2^{1-n}.$$

Відхиленням двох функцій  $f(x)$  та  $\Phi(x)$  називається величина

$$\Delta(f, \Phi) = \|f(x) - \Phi(x)\|_{C([a,b])}.$$

**Теорема 6.4.1** (Чебишова). *Серед усіх багаточленів  $n$ -го степеня з коефіцієнтом 1 при старшому степені  $\bar{T}_n(x)$  найменше відхиляється від 0 на  $[-1, 1]$ , тобто*

$$\|\bar{T}_n(x) - 0\|_{C([-1,1])} = \inf_{\bar{P}_n(x)} \|\bar{P}_n(x)\|_{C([-1,1])} = 2^{1-n}.$$

*Доведення.* Будемо доводити від супротивного: припустимо, що існує багаточлен, такий, що

$$\bar{Q}_n(x) < 2^{1-n}.$$

Тоді  $Q_{n-1}(x) = \bar{T}_n(x) - \bar{Q}_n(x)$  – поліном степеня не вище  $n-1$  і не рівний тотожно нулю. Дослідимо його знаки:

$$\text{sign}(Q_{n-1}(x'_k)) = \text{sign}(\bar{T}_n(x'_k) - \bar{Q}_n(x'_k)) = \text{sign}(\bar{T}_n(x'_k)) = \alpha \cdot (-1)^k, \quad \alpha = \pm 1.$$

Значить  $\exists z_k$ ,  $k = \overline{0, n-1}$  таке, що  $Q_{n-1}(z_k) = 0$ . Це протиріччя, бо  $Q_{n-1}(x)$  – поліном степеня  $\leq n-1$ .  $\square$

Тепер узагальнимо наш багаточлен Чебишова на довільний проміжок. Нагадаємо  $T_n(t) = \cos(n \arccos t)$ ,  $-1 \leq t \leq 1$ . Від змінної  $t \in [-1, 1]$  перейдемо до  $x \in [a, b]$ . Запровадимо заміну

$$t = \frac{2}{b-a}x - \frac{b+a}{b-a}, \quad x = \frac{b+a}{2} + \frac{b-a}{2}t.$$

Тоді

$$T_n^{[a,b]}(t) = \bar{T}_n \left( \frac{2}{b-a}x - \frac{b+a}{b-a} \right) = 2^{1-n} \cos \left( n \arccos \left( \frac{2x - (b+a)}{b-a} \right) \right).$$

Побудований нами багаточлен Чебишова на  $[a, b]$  не є нормованим. *Нормований багаточлен Чебишова* на  $[a, b]$ :

$$\bar{T}_n^{[a,b]}(x) = \frac{(b-a)^n}{2^{2n-1}} \cos \left( n \arccos \left( \frac{2x - (b+a)}{b-a} \right) \right).$$

Відповідно його нулі

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

а точки екстремуму

$$x'_k = \frac{a+b}{2} - \frac{b-a}{2}t'_k, \quad t'_k = \cos \left( \frac{k\pi}{n} \right), \quad k = \overline{0, n}.$$

Теорема Чебишова вірна і у випадку  $[a, b]$ . Тепер  $\left\| \bar{T}_n^{[a,b]} \right\|_{C([a,b])} = \frac{(b-a)^n}{2^{2n-1}}.$

Перейдемо до питання мінімізації залишкового члена. Нагадаємо, що

$$|r_n(x)| = |f(x) - L_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \cdot |\omega_n(x)|, \quad (6.13)$$

де  $M_{n+1} = \max_{x \in [a,b]} |f^{(n+1)}(x)|$ ,  $\omega_n(x) = \prod_{i=0}^n (x - x_i) = x^{n+1} + \dots$

Поставимо задачу

$$\inf_{\bar{P}_n(x)} \max_{x \in [a,b]} |\omega_n(x)|.$$

З теоремою Чебишова  $\omega_n(x) = \bar{T}_{n+1}^{[a,b]}(x)$  поліном Чебишова. Якщо співпадають поліноми, то співпадають їх нулі. Отже:  $x_k$  – вузли інтерполяції співпадають з нулями багаточлена Чебишова

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

В цьому випадку

$$|r_n(x)| = |f(x) - L_n(x)| \leq \frac{M_{n+1}}{(n+1)!} \cdot \frac{(b-a)^{n+1}}{2^{2n+1}}. \quad (6.14)$$

Цю оцінку не можна покращити! Так для  $f(x) = \overline{P}_{n+1}(x) = x^{n+1} + \dots$  її  $(n+1)$  похідна дорівнює  $(n+1)!$ , тому  $M_{n+1} = (n+1)!$ . Різниця  $f(x) - L_n(x) = \overline{T}_{n+1}^{[a,b]}(x)$ , отже

$$\max_{x \in [a,b]} |f(x) - L_n(x)| = \frac{(b-a)^{n+1}}{2^{2n+1}}.$$

## 6.5 Розділені різниці

Розділені різниці є аналогом похідної для функції, що задана таблицею.

*Розділеною різницею 1-го порядку* для функції  $f(x)$  називатимемо

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

*Розділеною різницею 2-го порядку* для функції  $f(x)$  називатимемо

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

Аналогічно визначаються розділені різниці довільного порядку.

Наведемо деякі властивості розділених різниць:

1.

$$f(x_0; \dots; x_n) = \sum_{i=0}^n \frac{f(x_i)}{\prod_{i \neq j} (x_i - x_j)}.$$

2. Розділена різниця – лінійний функціонал:

$$(\alpha_1 f_1 + \alpha_2 f_2)(x_0; x_1) = \alpha_1 f_1(x_0; x_1) + \alpha_2 f_2(x_0; x_1).$$

3. Розділена різниця – симетричний функціонал:

$$f(x_1; \dots; x_i; \dots; x_j; \dots; x_n) = f(x_1; \dots; x_j; \dots; x_i; \dots; x_n).$$

4. Для  $f(x) \in C^{(n)}([a, b])$   $\exists \xi \in [a, b]$ :  $f(x_0; x_1; \dots; x_n) = \frac{f^{(n)}(\xi)}{n!}$ .

**Задача 14.** Довести першу властивість розділених різниць.

Таблиця розділених різниць має вигляд:

$x_i$	$f_i$	р.р.1	р.р.2	...	р.р. $n$
$x_0$	$f(x_0)$				
		$f(x_0; x_1)$			
$x_1$	$f(x_1)$		$f(x_0; x_1; x_2)$		
		$f(x_1; x_2)$			
$x_2$	$f(x_2)$				
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
...	...	...	...	...	$f(x_0; \dots; x_n)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	
$x_n$	$f(x_n)$				

## 6.6 Інтерполяційна формула Ньютона

Запишемо формулу Лагранжа інтерполяційного багаточлена

$$L_n(x) = \sum_{i=0}^n f(x_i) \cdot \frac{\omega_n(x)}{(x - x_i) \cdot \omega'_n(x_i)}, \quad (6.15)$$

де  $\omega_n(x) = \prod_{j=0}^n (x - x_j)$ .

Позначимо  $\Phi_j(x) = L_j(x) - L_{j-1}(x)$ . Тоді, оскільки

$$L_n(x) = L_0(x) + (L_1(x) - L_0(x)) + \dots + (L_n(x) - L_{n-1}(x)),$$

$$L_j(x_i) = L_{j-1}(x_i) = f(x_i), \quad i = \overline{0, j-1},$$

то

$$\Phi_j(x_i) = A_j \cdot (x - x_0) \cdot \dots \cdot (x - x_{j-1}), \quad (6.16)$$

де  $A_j$  визначається з умови  $\Phi_j(x_j) = L_j(x_j) - L_{j-1}(x_j) = f(x_j) - L_{j-1}(x_j)$ . Звідси

$$\Phi_j(x) = \frac{f(x_j) - L_{j-1}(x_j)}{(x_j - x_0) \cdot \dots \cdot (x_j - x_{j-1})} \cdot (x - x_0) \cdot \dots \cdot (x - x_{j-1}).$$

Тоді

$$A_j = \frac{f(x_j) - L_{j-1}(x_j)}{(x_j - x_0) \cdot \dots \cdot (x_j - x_{j-1})} = \frac{f(x_j)}{(x_j - x_0) \cdot \dots \cdot (x_j - x_{j-1})} -$$



$$\begin{aligned}
& - \sum_{i=0}^{j-1} \frac{f(x_i)}{(x_i - x_0) \cdot \dots \cdot (x_i - x_{i-1}) \cdot (x_i - x_{i+1}) \cdot \dots \cdot (x_i - x_{j-1}) \cdot (x_j - x_i)} = \\
& = \frac{f(x_j)}{(x_j - x_0) \cdot \dots \cdot (x_j - x_{j-1})} + \sum_{i=0}^{j-1} \frac{f(x_i)}{(x_i - x_0) \cdot \dots \cdot (x_i - x_j)} = \\
& = \sum_{i=0}^j \frac{f(x_i)}{(x_i - x_0) \cdot \dots \cdot (x_i - x_{i-1}) \cdot (x_i - x_{i+1}) \cdot \dots \cdot (x_i - x_j)} = f(x_0; \dots; x_j).
\end{aligned}$$

Звідси маємо інтерполяційну формулу Ньютона вперед ( $x_0 \rightarrow x_n$ ):

$$L_n(x) = f(x_0) + f(x_0; x_1) \cdot (x - x_0) + \dots + f(x_0; \dots; x_n) \cdot (x - x_0) \cdot \dots \cdot (x - x_{n-1}). \quad (6.17)$$

Аналогічно, інтерполяційна формула Ньютона назад ( $x_n \rightarrow x_0$ ):

$$\begin{aligned}
L_n(x) = f(x_n) + f(x_n; x_{n-1}) \cdot (x - x_n) + \dots \\
\dots + f(x_n; \dots; x_0) \cdot (x - x_n) \cdot \dots \cdot (x - x_1). \quad (6.18)
\end{aligned}$$

Маємо рекурсію за степенем багаточлена

$$L_n(x) = L_{n-1}(x) + f(x_0; \dots; x_n) \cdot (x - x_0) \cdot \dots \cdot (x - x_1).$$

Звідси

$$L_n(x) = f(x_0) + (x - x_0) \cdot (f(x_0; x_1) + (x - x_1) \cdot (\dots + (x - x_{n-1}) \cdot f(x_0; x_1; \dots; x_n)))$$

і цю формулу розкриваємо починаючи з середини (це аналог схеми Горнера обчислення значення багаточлена).

Введемо нову формулу для похибки інтерполявання. Для  $x \neq x_i, i = \overline{0, n}$  розглянемо розділену різницю

$$f(x; x_0; \dots; x_n) = \frac{f(x)}{(x - x_0) \cdot \dots \cdot (x - x_n)} + \sum_{k=0}^n \frac{f(x_k)}{\prod_{i \neq k} (x - x_i)}.$$

Звідси

$$\begin{aligned}
f(x) = f(x_0) \cdot \frac{(x - x_1) \cdot \dots \cdot (x - x_n)}{(x_0 - x_1) \cdot \dots \cdot (x_0 - x_n)} + \dots + f(x_n) \cdot \frac{(x - x_1) \cdot \dots \cdot (x - x_n)}{(x_n - x_1) \cdot \dots \cdot (x_n - x_{n-1})} + \\
+ f(x; x_0; \dots; x_n) \cdot (x - x_0) \cdot \dots \cdot (x - x_n) = L_n(x) + f(x; x_0; \dots; x_n) \cdot \omega_n(x).
\end{aligned}$$

Тоді похибка має вигляд

$$r_n(x) = f(x) - L_n(x) = f(x; x_0; \dots; x_n) \cdot \omega_n(x). \quad (6.19)$$

Це нова форма для залишкового члена.

Порівнюючи з формулою залишкового члена в пункті (6.3), маємо

$$f(x; x_0; \dots; x_n) = \frac{f^{(n+1)}(\xi)}{(n+1)!},$$

що доводить останню властивість розділених різниць.

Нехай маємо сітку рівновіддалених вузлів:  $x_i = a + ih$ ,  $h = \frac{b-a}{n}$ ,  $i = \overline{0, n}$ ,  $x_0 = a$ ,  $x_n = b$ . Розначимо  $\Delta f_0 = f_1 - f_0$ ,  $\Delta^2 f_0 = \Delta f_1 - \Delta f_0 = f_2 - 2f_1 + f_0$ ,  $\dots$  – скінченні різниці. Запишемо формули Ньютона у нових позначеннях:

$$L_n(x) = L_n(x_0 + th) = f_0 + t\Delta f_0 + \dots + \frac{t \cdot (t-1) \cdot (t-n+1)}{n!} \cdot \Delta^n f_0, \quad t = \frac{x - x_0}{h}.$$

Це інтерполяційна формула Ньютона вперед по рівновіддалених вузлах.

**Задача 15.** Побудувати інтерполяційну формулу Ньютона назад по рівновіддалених вузлах.

## 6.7 Інтерполювання з кратними вузлами

Нехай  $f(x)$  задана таблицею значень  $f^{(j)}(x_i)$ ,  $i = \overline{0, n}$ ,  $j = \overline{0, k_i - 1}$ ,  $k_i$  – кратності відповідних вузлів. Побудуємо  $H_m^{(i)}(x_j) = f^{(i)}(x_j)$  – інтерполяційний багаточлен Ерміта по кратним вузлах, де  $m = \sum_{i=1}^n k_i - 1$ .

Якщо  $k_i = 1$ , то  $H_m(x) = L_n(x)$ .

Для побудови  $H_m(x)$  в загальному випадку для кожної точки  $x_i$  введемо  $k_i$  точок  $x_{i,j}^\varepsilon = x_i + j\varepsilon$ ,  $i = \overline{0, n}$ ,  $j = \overline{0, k_i - 1}$ . З умови  $\forall i: x_{i,k_i-1}^\varepsilon = x_i + \varepsilon(k-1) < x_{i+1}$  можна вибрати  $\varepsilon$ .

Нехай  $f(x) \in C^{(m)}([a, b])$ . Запишемо інтерполяційну формулу Ньютона:

$$\begin{aligned} L_m^\varepsilon = & f(x_{0,0}^\varepsilon) + f(x_{0,0}^\varepsilon; x_{0,1}^\varepsilon) \cdot (x - x_{0,0}^\varepsilon) + \dots \\ & \dots + f(x_{0,0}^\varepsilon; \dots; x_{n,k_n-1}^\varepsilon) \cdot (x - x_{0,0}^\varepsilon) \cdot \dots \cdot (x - x_{n,k_n-1}^\varepsilon). \end{aligned}$$

При  $\varepsilon \rightarrow 0$  маємо  $x_{i,j}^\varepsilon \rightarrow x_i$ . Крім того

$$f(x_{i,0}^\varepsilon; \dots; x_{i,k_i-1}^\varepsilon) = f(x_i; \dots; x_i) = \frac{f^{(k_i)}(x_i)}{k_i!}.$$

Тому  $L_m^\varepsilon(x) \rightarrow H_m(x)$  та

$$R_m(x) = f(x) - H_m(x) = \frac{f^{(m+1)}(\xi)}{(m+1)!} \cdot \Omega_m(x),$$

де  $\Omega_m(x) = (x - x_0)^{k_0} \cdot \dots \cdot (x - x_n)^{k_n}$ .

## 6.8 Збіжність процесу інтерполювання

Виникає питання, чи буде прямувати до нуля похибка інтерполювання  $f(x) - L_n(x)$ , якщо число вузлів  $n$  збільшувати?

Введемо норму  $\|f(x) - L_n\|_{C([a,b])} = \max_{x \in [a,b]} |f(x) - L_n(x)|$ . Тоді для довільної  $f(x) \in C^{(n+1)}([a,b])$  справджується оцінка

$$\|f(x) - L_n(x)\|_{C([a,b])} \leq \frac{M_{n+1}}{(n+1)!} \|\omega_n(x)\|_{C([a,b])}, \quad (6.20)$$

де  $M_{n+1} = \max_{x \in [a,b]} |f^{(n+1)}(x)|$ ,  $\omega_n(x) = \prod_{i=0}^n (x - x_i)$ . А яка оцінка буде для довільної неперервної функції?

Кажуть, що інтерполяційний процес для функції  $f(x)$  *збігається* в точці  $x \in [a,b]$ , якщо

$$\lim_{n \rightarrow \infty} L_n(x) = f(x), \quad \forall \{x_i\}_{i=1}^n : h = \max_{i=\overline{1,n}} h_i \rightarrow 0, \quad h_i = x_i - x_{i-1}.$$

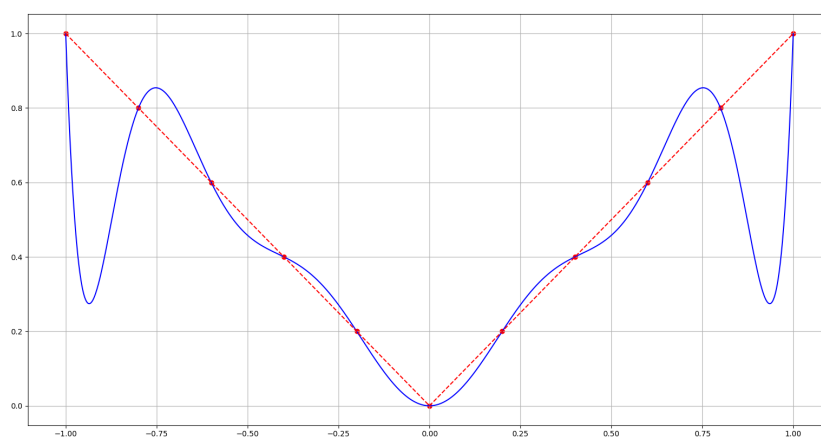
Якщо  $\|f(x) - L_n(x)\|_{C([a,b])} \xrightarrow{n \rightarrow \infty} 0$ , то інтерполяційний процес збігається рівномірно.

Розглянемо приклади поведінки інтерполяційних багаточленів при  $n \rightarrow \infty$  для деяких функцій.

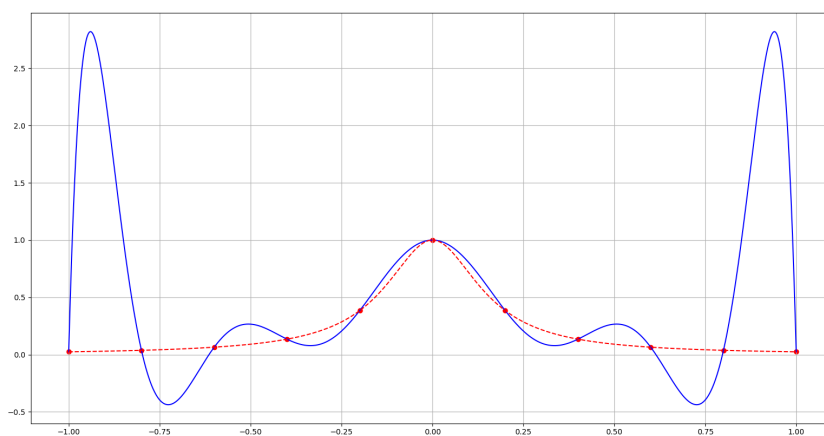
**Приклад 1.** Послідовність інтерполяційних багаточленів (сітка рівномірна), побудованих для неперервної функції  $f(x) = |x|$ ,  $-1 \leq x \leq 1$  (функція неперервна, але негладка), не збігається на  $x \in [-1, 1]$ , крім

точок  $x = -1, 0, 1$ .

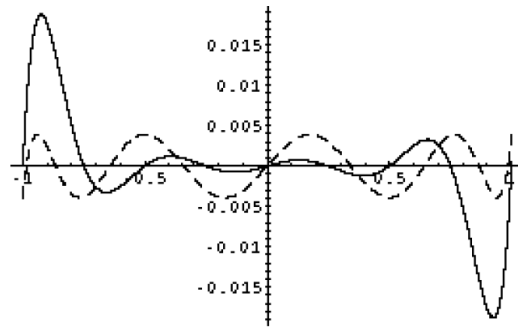
На рисунку дано графіки самої функції (штрихова лінія) та інтерполяційного багаточлена (суцільна лінія) на рівномірній сітці  $x_i = -1 + ih$ ,  $h = \frac{2}{n}$ ,  $i = \overline{0, n}$  для  $n = 10$ :



**Приклад 2.** Функція Рунге  $f(x) = \frac{1}{1+40x^2}$ ,  $-1 \leq x \leq 1$  (функція аналітична!). Для рівномірної сітки  $x_i = -1 + ih$ ,  $h = \frac{2}{n}$ ,  $i = \overline{0, n}$  маємо графіки: суцільна лінія – інтерполяційного багаточлена; пунктирна – самої функції для  $n = 10$ :



Пояснити чому рівномірна сітка дає великі похибки інтерполювання біля кінців інтервалу інтерполювання допомагає наступний рисунок. На цьому рисунку суцільною лінією представлено графік функції  $\omega_n = \prod_{i=0}^n (x - x_i)$  ( $n = 8$ ) для рівномірної сітки. Як бачимо максимальні за модулем значення цієї функції припадають на кінці інтервалу.



Для порівняння на цьому ж рисунку (штрихова лінія) побудовано графік  $\omega_n = \prod_{i=0}^n (x - x_i)$ , що відповідає чебишовським вузлам, які мінімізують похибку інтерполювання. Тепер відхилення  $\omega_n(x)$  розподілено рівномірно по всьому проміжку інтерполювання.

**Теорема 6.8.1** (Фабера).  $\forall \{x_i\}_{i=0}^n$  існує  $f(x) \in C([a, b])$ , для якої інтерполяційний процес не збігається рівномірно.

**Теорема 6.8.2** (Марцинкевича).  $\forall f(x) \in C([a, b])$  існують  $\{x_i\}_{i=0}^n$  такі, що послідовність  $\{L_n(x)\}$  збігається рівномірно до  $f(x)$ .

**Теорема 6.8.3.** Стала Лебега  $\|P_n\| = \max_j \sum_{j=0}^n |\varphi_j^{(n)}(x)|$ , де  $\varphi_j^{(n)}(x) = \frac{\omega_n(x)}{(x - x_j) \cdot \omega'_n(x_j)}$ .

**Теорема 6.8.4.** Для  $f(x) \in C([a, b])$ :

$$\|f(x) - L_n(x)\|_{C([a, b])} \leq (1 + \|P_n\|) \cdot E_n(f),$$

де  $E_n(f) = \inf Q_n(x) \|f(x) - Q_n(x)\|_{C([a, b])}$  – відхилення багаточлена  $n$ -го степеня найкращого рівномірного наближення від  $f(x)$ .

**Теорема 6.8.5.** Нехай  $P_n^E$  – оператор інтерполяції на рівномірній сітці,  $P_n^T$  – оператор інтерполяції на чебишовській сітці. Тоді на  $[-1, 1]$  маємо наближені оцінки:

$$\|P_n^E\| \approx C_1 \cdot 2^n, \quad \|P_n^T\| \approx C_2 \cdot \ln(n).$$

Останні оцінки пояснюють розбіжність процесу інтерполювання при великих  $n$ .

## 6.9 Кусково-лінійна інтерполяція

Інтерполяція багаточленом Лагранжа або Ньютона на відрізку  $[a, b]$  з використанням великої кількості вузлів інтерполяції часто приводить до поганого наближення через розбіжність процесу інтерполювання. Для того щоб уникнути великої похибки, весь відрізок  $[a, b]$  розбивають на частинні відрізки  $[x_{i-1}, x_i]$  і на кожному з частинних відрізків замінюють функцію  $f(x)$  багаточленом невисокого степеню. В цьому і полягає кусково-поліноміальна інтерполяція.

Розглянемо найпростішу таку інтерполяцію – лінійну. Нехай задана  $f(x)$  значеннями  $f(x_i)$ ,  $i = \overline{0, n}$ . Побудуємо функцію  $\Phi_1(x)$  – лінійну на  $x \in [x_{i-1}, x_i]$ , що інтерполіює ці значення:

$$\Phi_1(x) = L_1^i(x) = f(x_{i-1}) \cdot \frac{x - x_{i-1}}{x_i - x_{i-1}} + f(x_i) \cdot \frac{x_i - x}{x_i - x_{i-1}}, \quad x \in [x_{i-1}, x_i]. \quad (6.21)$$

Представимо її у вигляді

$$\Phi_1(x) = \sum_{i=0}^n f(x_i) \cdot \varphi_i(x). \quad (6.22)$$

З умов інтерполювання маємо

$$\Phi_1(x_j) = \sum_{i=0}^n f(x_i) \cdot \varphi_i(x_j) = f(x_j).$$

Звідси

$$\varphi_i(x_j) = \delta_{i,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}.$$

Значить

$$\varphi_i(x) = \begin{cases} 0, & a \leq x \leq x_{i-1} \\ \frac{x - x_{i-1}}{x_i - x_{i-1}}, & x_{i-1} \leq x \leq x_i \\ \frac{x_{i+1} - x}{x_{i+1} - x_i}, & x_i \leq x \leq x_{i+1} \\ 0, & x_{i+1} \leq x \leq b \end{cases}$$

**Теорема 6.9.1.** Для довільної  $f(X) \in C^{(2)}([a, b])$  справедлива оцінка

$$\|f(x) - \Phi_1(x)\|_{C([a,b])} \leq \frac{M_2}{8} \cdot |h|^2, \quad (6.23)$$

де  $\Phi_1(x)$  – кусково-лінійна функція, побудована по значеннях  $f(x_i)$ ,  $i = \overline{0, n}$ ,  $|h| = \max_i h_i$ ,  $h_i = x_i - x_{i-1}$ .

*Доведення.* Маємо для  $x \in [x_{i-1}, x_i]$ :

$$z(x) = f(x) - \Phi_1(x) = f(x) - L_1^i(x) = \frac{f''(\xi_i)}{2!} \cdot (x - x_{i-1}) \cdot (x - x_i).$$

Звідси

$$|f(x) - \Phi_1(x)| \leq \frac{M_2^i}{2} \cdot |(x - x_{i-1}) \cdot (x - x_i)| \leq \frac{M_2^i \cdot h_i^2}{8}, \quad (6.24)$$

де  $M_2^i = \max_{x \in [x_{i-1}, x_i]} |f''(x)|$ . Остання оцінка отримана з нерівності

$$\max_{[x_{i-1}, x_i]} |(x - x_{i-1}) \cdot (x - x_i)| = \frac{h_i^2}{4}.$$

Тоді

$$\max_{i=\overline{1, n}} \max_{x \in [x_{i-1}, x_i]} |z(x)| \leq \frac{1}{8} h^2 M_2, \quad (6.25)$$

де  $M_2 = \max_{x \in [a, b]} |f''(x)|$ ,  $h_i = \max_i h_i$ , що доводить (6.23).  $\square$

**Задача 16.** Довести оцінку  $|f'(x) - \Phi'_1(x)| \leq |h| \cdot M_2$ .

Отже маємо збіжність процесу інтерполювання за допомогою кусково-лінійної функції

$$\left\| f(x) - \Phi_1^{(n)}(x) \right\|_{C([a,b])} \xrightarrow{h \rightarrow 0, n \rightarrow \infty} 0,$$

$$\left\{ \Phi_1^{(n)}(x) \right\} \Rightarrow f(x).$$

Розглянемо деякі простори:

1.  $H_0 = L_2(a, b)$  – гільбертів простір, в якому скалярний добуток визначається так:  $(u, v) = \int_a^b (u(x) \cdot v(x)) \, dx$ , а норма  $\|u\|_0 = \sqrt{(u, u)}$ .

2.  $H_k = W_2^k(a, b)$ . Тепер скалярний добуток

$$(u, v)_k = \sum_{m=0}^k \int_a^b (u^{(m)}(x) \cdot v^{(m)}(x)) \, dx,$$

$$\text{а норма } \|u\|_k = \sqrt{\|u\|_0^2 + \dots + \|u\|_k^2}.$$

**Теорема 6.9.2.** *Нехай  $f(x) \in H_2 = W_2^2(a, b)$ . Тоді  $\|f^{(k)} - \Phi_1^{(k)}\|_0 \leq |h|^{2-k} \cdot \|f\|_2$ ,  $k = 1, 2$ .*

Зауважимо, що кусково-лінійна інтерполяція негладка, тому на практиці застосовують квадратичні, а найчастіше – кубічні поліноми на кожному проміжку.

## 6.10 Кусково-кубічна ермітова інтерполяція

Нехай деяка функція  $f(x)$  задана в точках  $x_i$  своїми значеннями та значеннями похідної:  $y_i = f(x_i)$ ,  $y'_i = f'(x_i)$ ,  $i = \overline{0, n}$ . Позначимо через  $\Phi_3(x)$  функцію, яка буде інтерполювати задану. Тоді

$$\Phi_3(x) = H_3^i(x), \quad x \in [x_{i-1}, x_i]. \quad (6.26)$$

Неважко написати явний вигляд цього багаточлена  $H_3^i(x)$  на проміжку:

$$\begin{array}{c|cccc} x_i & y_i & & & \\ & & y'_i & & \\ x_i & y_i & & \frac{y_{i-1,i} - y'_i}{h_i} & \\ & & y_{i-1,i} & & \frac{y'_i - 2y_{i-1,i} + y'_{i-1}}{h_i^2} \\ x_{i-1} & y_{i-1} & & \frac{y'_{i-1} - y_{i-1,i}}{h_i} & \\ & & y'_{i-1} & & \\ x_{i-1} & y_{i-1} & & & \end{array}$$

$$\begin{aligned} H_3^i(x) = & y_i + y'_i(x - x_i) + \frac{y_{i-1,i} - y'_i}{h_i} \cdot (x - x_i)^2 + \\ & + \frac{y'_i - 2y_{i-1,i} + y'_{i-1}}{h_i^2} \cdot (x - x_i)^2 \cdot (x - x_{i-1}) \end{aligned}$$



Можна представити кусково-кубічну функцію і в такому вигляді:

$$\Phi_3(x) = \sum_{i=0}^n (y_i \cdot \varphi_i^0(x) + y'_i \cdot \varphi_i^1(x)). \quad (6.27)$$

Умови інтерполювання:  $\Phi_3(x_i) = y_i$ ,  $\Phi'_3(x_i) = y'_i$ ,  $i = \overline{0, n}$ . Якщо ці умови підставити в (6.27), то отримаємо умови на базисні функції:

$$\varphi_i^0(x_j) = \delta_{i,j}, \quad (\varphi_i^0)'(x_j) = 0, \quad i, j = \overline{0, n}.$$

$$\varphi_i^1(x_j) = 0, \quad (\varphi_i^1)'(x_j) = \delta_{i,j}.$$

Ці функції кусково-кубічні, тобто  $\varphi_i^k(x) \in \pi_3$ ,  $x \in [x_{i-1}, x_{i+1}]$ ,  $k = 0, 1$  ( $\pi_3$  – множина багаточленів третього степеня), на всіх інших проміжках вони нульові. Нехай  $h_i \equiv h$ , і позначимо  $s = \frac{x-x_i}{h}$ ,  $x \in [x_{i-1}, x_i] \Rightarrow s \in [-1, 0]$ .

1. введемо  $\bar{\varphi}_1^0(s) = \varphi_i^0(x)$ ,  $x \in [x_{i-1}, x_{i+1}]$ ,  $s \in [0, 1]$ . Побудуємо цю функцію. Вона задовольняє умовам:

$$\bar{\varphi}_1^0(0) = 1, \quad \bar{\varphi}_1^0(1) = 0, \quad (\bar{\varphi}_1^0)'(0) = (\bar{\varphi}_1^0)'(1) = 0.$$

Її явний вигляд отримаємо за допомогою таблиці розділених різниць:

$$\begin{array}{c|cccc} 0 & 1 & & & \\ & & 0 & & \\ 0 & 1 & & -1 & \\ & & -1 & & 2 \\ 1 & 0 & & 1 & \\ & & 0 & & \\ 1 & 0 & & & \end{array}$$

$$H_3(s) = 1 + 0 \cdot s - 1 \cdot s^2 + 2s^2(s-1) = 2s^3 - 3s^2 + 1 \equiv \bar{\varphi}_1^0(s).$$

Аналогічно

2.  $\bar{\varphi}_2^0(s) = -2s^3 - 3s^2 + 1$ ,  $\varphi_i^0(x) = \bar{\varphi}_2^0(s)$ ,  $x \in [x_{i-1}, x_i]$ ,  $s \in [-1, 0]$ ;
3.  $\bar{\varphi}_1^1(s) = s(s-1)^2$ ,  $\varphi_i^1(x) = h\bar{\varphi}_1^1(s)$ ,  $x \in [x_i, x_{i+1}]$ ,  $s \in [0, 1]$ ;
4.  $\bar{\varphi}_2^1(s) = s(s+1)^2$ ,  $\varphi_i^1(x) = h\bar{\varphi}_2^1(s)$ ,  $x \in [x_{i-1}, x_i]$ ,  $s \in [-1, 0]$ .

А тепер будемо явний вигляд функцій  $\varphi_i^k(x)$  для довільного проміжку  $x \in [x_{i-1}, x_{i+1}]$ :

$$\varphi_i^0(x) = \begin{cases} 0, & a \leq x \leq x_{i-1}, \\ -2s^3 - 3s^2 + 1, & x_{i-1} \leq x \leq x_i, \\ 2s^3 - 3s^2 + 1, & x_i \leq x \leq x_{i+1}, \\ 0, & x_{i+1} \leq x \leq b, \end{cases}$$

$$\varphi_i^1(x) = \begin{cases} 0, & a \leq x \leq x_{i-1}, \\ hs(s+1)^2, & x_{i-1} \leq x \leq x_i, \\ hs(s-1)^2, & x_i \leq x \leq x_{i+1}, \\ 0, & x_{i+1} \leq x \leq b, \end{cases}$$

де  $s = \frac{x-x_i}{h}$  (якщо сітка нерівномірна, то в формулах замість  $h$ , буде  $h_i$  або  $h_{i+1}$  на відповідних інтервалах).