

Міністерство освіти та науки України  
Національний університет «Запорізька політехніка»

## **МЕТОДИЧНІ ВКАЗІВКИ**

**до лабораторних робіт з дисципліни**  
**«Математичні моделі та обчислювальні методи**  
**досліджень»**  
для студентів спеціальності 141  
«Електроенергетика, електротехніка та електромеханіка»  
(освітня програма «Електричні машини і апарати»)  
денної форми навчання

Методичні вказівки до лабораторних робіт з дисципліни «Математичні моделі та обчислювальні методи досліджень» для студентів спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» (освітня програма «Електричні машини і апарати») денної форми навчання. / Укл.: Ю.С. Безверхня, Д.С. Яримбаш – Запоріжжя : НУ ЗП, 2023. - 58 с.

Укладачі: Д.С. Яримбаш зав. кафедри, докт. техн. наук  
Ю.С. Безверхня, доцент, докт. філософії

Рецензент: Т.Є. Дівчук, доцент, канд. техн. наук

Відповідальний  
за випуск: Г.В. Дьомічева, зав.лаб.

Затверджено  
на засіданні кафедри  
«Електричні машини»  
Протокол № 2  
від 21.09.23 р.

Рекомендовано до видання  
НМК Електротехнічного  
факультету  
Протокол № 2  
від 19 жовтня 2023р.

## ЗМІСТ

Вступ .....	4
<b>1 Лабораторна робота № 1 «Рішення нелінійних рівнянь» .....</b>	<b>5</b>
1.1 Теоретичний матеріал. ....	5
1.2 Порядок виконання лабораторної роботи. ....	12
1.3 Зміст звіту .....	13
1.4 Контрольні запитання для самоперевірки .....	14
<b>2 Лабораторна робота № 2 «Рішення систем лінійних рівнянь» ..</b>	<b>15</b>
2.1 Теоретичний матеріал .....	15
2.2 Порядок виконання лабораторної роботи. ....	20
2.3 Зміст звіту .....	23
2.4 Контрольні запитання для самоперевірки .....	23
<b>3 Лабораторна робота № 3 «Рішення систем нелінійних рівнянь»</b>	<b>24</b>
3.1 Теоретичний матеріал .....	24
3.2 Порядок виконання лабораторної роботи. ....	28
3.3 Зміст звіту .....	29
3.4 Контрольні запитання для самоперевірки .....	29
<b>4 Лабораторна робота № 4 «Методи інтерполяції і апроксимації»</b>	<b>30</b>
4.1 Теоретичний матеріал .....	30
4.2 Порядок виконання лабораторної роботи. ....	37
4.3 Зміст звіту .....	39
4.4 Контрольні запитання для самоперевірки .....	40
<b>5 Лабораторна робота № 5 «Чисельне інтегрування» .....</b>	<b>41</b>
5.1 Теоретичний матеріал .....	41
5.2 Порядок виконання лабораторної роботи. ....	47
5.3 Зміст звіту .....	49
5.4 Контрольні запитання для самоперевірки .....	50
Рекомендована література. ....	51
Додаток А. Убудовані оператори. ....	52
Додаток Б. Убудовані функції .....	54
Додаток В. Оператори мови програмування Mathcad. ....	55

## ВСТУП

Однією з основних областей застосування ПК є математичні і науково-технічні розрахунки. Складні обчислювальні задачі, що виникають при моделюванні технічних пристроїв і складних електромеханічних процесів, можна розбити на ряд елементарних:

- обчислення інтегралів,
- рішення рівнянь,
- рішення диференціальних рівнянь і т.д.

Для таких задач уже розроблено методи рішення та створено математичні системи, доступні для вивчення.

Мета посібника – навчити користуватися чисельними методами обчислень із використанням сучасних інформаційних технологій. Найбільше підходить для цієї мети є одна із самих потужних і ефективних математичних систем – MathCAD, що займає особливе місце серед безлічі таких систем (Matlab, Maple, Mathematica і ін.). MathCAD залишається єдиною системою, у якій опис рішення математичних задач задається за допомогою звичних математичних формул і знаків. MathCAD дозволяє виконувати як чисельні, так і аналітичні (символьні) обчислення, має надзвичайно зручний математично-орієнтований інтерфейс і прекрасні засоби наукової графіки.

У даному посібнику ми розглянемо на прикладах, як вирішуються в MathCAD різноманітні задачі чисельного аналізу (рішення систем лінійних і нелінійних рівнянь, рішення диференціальних рівнянь, апроксимація функцій і т.д).

Кожна лабораторна робота містить стислий опис методів обчислень, приклади з необхідними коментарями, порядок виконання лабораторної роботи і варіанти індивідуальних завдань, контрольні запитання.

# 1 ЛАБОРАТОРНА РОБОТА № 1

## РІШЕННЯ НЕЛІНІЙНИХ РІВНЯНЬ

**Мета роботи:** засвоєння методів рішення нелінійних рівнянь та способів їх реалізації за допомогою системи MathCAD.

### 1.1 Теоретичний матеріал

Нелінійні рівняння можна розділити на два класи – алгебраїчні і трансцендентні.

**Алгебраїчними рівняннями** називають рівняння, що містять тільки алгебраїчні функції (цілі, раціональні, ірраціональні). Зокрема, багаточлен є цілою алгебраїчною функцією.

**Трансцендентними рівняннями** називаються рівняння, що містять інші функції (тригонометричні, показові, логарифмічні та інші).

Методи рішення нелінійних рівнянь поділяють на дві групи:

- точні методи;
- ітераційні методи.

**Точні методи** дозволяють записати корені у виді деякого кінцевого співвідношення (формули) – методи для рішення тригонометричних, логарифмічних, показових та найпростіших алгебраїчних рівнянь.

Але багато рівнянь і систем рівнянь не мають аналітичних рішень. У першу чергу це більшість трансцендентних рівнянь. Доведено також, що не можна побудувати формулу, по якій можна було б вирішити довільне алгебраїчне рівняння ступеня вище четвертого. Крім того, у деяких випадках рівняння містить коефіцієнти, відомі лише приблизно, і, отже, сама задача про точне визначення коренів рівняння втрачає зміст. Для їхнього рішення використовуються **ітераційні методи** з заданим ступенем точності.

Нехай дано рівняння

$$f(x) = 0, \tag{1.1}$$

яке задовольняє наступним умовам:

- функція  $f(x)$  безперервна на відріжку із межами  $[a, b]$  разом із своїми похідними 1-го і 2-го порядку;

- значення функції  $f(x)$  на кінцях відрізка мають різні знаки ( $f(a) \cdot f(b) < 0$ );

- перша і друга похідні функції  $f'(x)$  і  $f''(x)$  зберігають визначений знак на усьому відрізку.

Перша і друга умови гарантують, що на інтервалі  $[a, b]$  знаходиться хоча б один корінь, а з третьої умови випливає, що функція  $f(x)$  на даному інтервалі монотонна і тому корінь буде єдиним.

Вирішити рівняння (1.1) **ітераційним методом** означає установити, чи має воно корені, скільки коренів і знайти значення коренів із потрібною точністю.

Всяке значення  $\xi$ , що звертає функцію  $f(x)$  у нуль

$$f(\xi) = 0 \quad (1.2)$$

називається **коренем рівняння** (1.1) або **нулем** функції  $f(x)$ .

Задача знаходження кореня рівняння  $f(x) = 0$  ітераційним методом складається з двох етапів:

- відділення коренів – знаходження наближеного значення кореня або відрізка, що його містить;
- уточнення наближених коренів – доведення їх до заданого ступеня точності.

Процес відділення коренів починається з установлення знаків функції  $f(x)$  у граничних  $x = a$  і  $x = b$  точках області її існування.

Наближені значення коренів (**початкові наближення**) можуть бути також відомі із фізичного смислу задачі, із рішення аналогічної задачі при інших початкових даних, або можуть бути знайдені графічним способом.

У інженерній практиці поширений **графічний засіб** визначення наближених коренів.

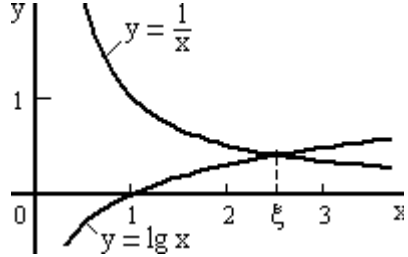
**Дійсні корені рівняння** (1.1) – це точки перетинання графіка функції  $f(x)$  із віссю абсцис. Тобто, для рішення графічним способом достатньо побудувати графік функції  $f(x)$  і визначити точки перетинання  $f(x)$  із віссю  $Ox$ , або відзначити на осі  $Ox$  відрізки, що містять по одному кореню. Побудову графіків часто вдається сильно спростити, замінивши рівняння (1.1) **рівнозначним** йому рівнянням:

$$f_1(x) = f_2(x), \quad (1.3)$$

де функції  $f_1(x)$  і  $f_2(x)$  – більш прості, ніж функція  $f(x)$ .

Побудувавши графіки функцій  $y = f_1(x)$  і  $y = f_2(x)$ , отримаємо шукані корені, якими й будуть абсиси точок перетинання графіків цих, більш простих, функцій (рис. 1.1).

**Ітераційний процес** складається у послідовному уточненні початкового наближення  $x_0$ . Кожний такий крок називається *ітерацією*. У результаті ітерацій знаходиться послідовність наближених значень кореня  $x_1, x_2, \dots, x_n$ . Якщо ці значення зі збільшенням числа ітерацій  $n$  наближаються до істинного значення кореня, то говорять, що ітераційний процес *сходиться*.



**Рисунок 1.1 - Графічний спосіб визначення коренів**

#### Метод ділення відрізка навпіл (метод бісекції)

Для знаходження кореня рівняння (1.1), що належить відрізку  $[a, b]$ , ділимо цей відрізок навпіл.

Якщо  $f\left(\frac{a+b}{2}\right) = 0$ , то  $\xi = \frac{a+b}{2}$  є коренем рівняння.

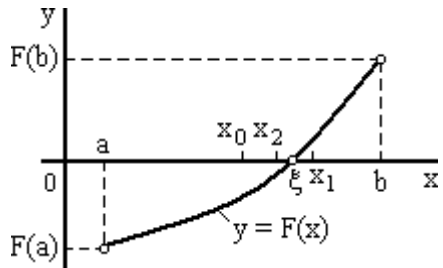
Якщо  $f\left(\frac{a+b}{2}\right) \neq 0$  (що, практично, частіше буває), то вибираємо

ту з половин  $\left[a, \frac{a+b}{2}\right]$  або  $\left[\frac{a+b}{2}, b\right]$ , на кінцях якої функція  $f(x)$  має протилежні знаки.

Новий звужений відрізок  $[a_1, b_1]$  знову ділимо навпіл і робимо ті ж самої дії (рис. 1.2).

Метод половинного ділення дуже простий і може забезпечити практично будь-яку точність, завжди сходиться.

Недоліком методу є повільна збіжність (за один крок інтервал, де знаходиться корінь, зменшується усього в два рази).



**Рисунок 1.2 – Метод бісекції**

### Метод хорд

У методі хорд процес ітерацій складається з того, що в якості наближень до кореня рівняння (1.1) приймають значення  $x_1, x_2, \dots, x_n$  точок перетинання хорди  $AB$  із віссю абсцис (рис. 1.3).

Спочатку запишемо рівняння хорди  $AB$ :

$$\frac{y - f(a)}{f(b) - f(a)} = \frac{x - a}{b - a}.$$

Для точки перетинання хорди  $AB$  із віссю абсцис ( $x = x_1, y = 0$ ) одержимо рівняння:

$$x_1 = a - \frac{f(a)}{f(b) - f(a)}(b - a). \quad (1.4)$$

Нехай для певності  $f''(x) > 0$  при  $a \leq x \leq b$  (випадок  $f''(x) < 0$  зводиться до нашого, якщо записати рівняння у виді  $-f(x) = 0$ ). Тоді крива  $y = f(x)$  буде випукла униз, отже, розташована нижче своєї хорди  $AB$ . Можливі два випадки:  $f(a) > 0$  (рис. 1.3, а);  $f(b) > 0$  (рис. 1.3, б).

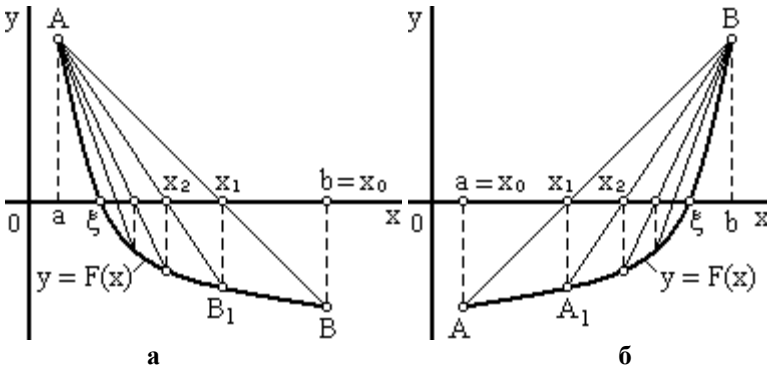


Рисунок 1.3 - Метод хорд

У першому випадку (рис. 1.3, а) кінець  $a$  нерухомий і послідовні наближення:  $x_0 = b$ ;

$$x_{i+1} = x_i - \frac{f(x_i)}{f(x_i) - f(a)}(x_i - a), \quad (1.5)$$

де  $i = 1, 2, \dots, n$  утворюють обмежену монотонно убутну послідовність, причому

$$a < \xi < \dots < x_{i+1} < x_i < \dots < x_1 < x_0.$$

В другому випадку (рис. 1.3, б) нерухомий кінець  $b$ , а послідовні наближення:  $x_0 = a$ ;

$$x_{i+1} = x_i - \frac{f(x_i)}{f(b) - f(x_i)}(b - x_i) \quad (1.6)$$

утворюють обмежену монотонно зростаючу послідовність, причому

$$x_0 < x_1 < \dots < x_i < x_{i+1} < \dots < \xi < b.$$

Узагальнюючи ці результати, укладаємо:

- нерухомий той кінець, для якого знак функції  $f(x)$  збігається зі знаком її другої похідної  $f''(x)$ ;
- послідовні наближення  $x_n$  лежать по ту сторону кореня  $\xi$ , де функція  $f(x)$  має знак, протилежний знаку її другої похідної  $f''(x)$ .

Ітераційний процес продовжується доти, поки не буде виявлено, що  $|x_i - x_{i-1}| < \varepsilon$ , де  $\varepsilon$  – задана гранична абсолютна похибка.

### Метод Ньютона

Відмінність цього ітераційного методу від попереднього складається в тому, що замість хорди на кожному кроку проводиться дотична до кривої  $y = f(x)$  при  $x = x_i$  і шукається точка перетинання дотичної з віссю абсцис (рис. 1.4). При цьому не обов'язково задавати відрізок  $[a, b]$ , що містить корінь рівняння (1.1), достатньо знайти лише деяке початкове наближення кореня –  $x = x_0$ .

Застосовуючи метод Ньютона, варто керуватися таким правилом: у якості вихідної точки  $x_0$  вибирається той кінець інтервалу  $[a, b]$ , якому відповідає ордината того ж знака, що і знак  $f''(x)$ .

Рівняння дотичної, проведеної до кривої  $y = f(x)$  через точку  $B_0$  з координатами  $x_0$  і  $f(x_0)$ , має вид:

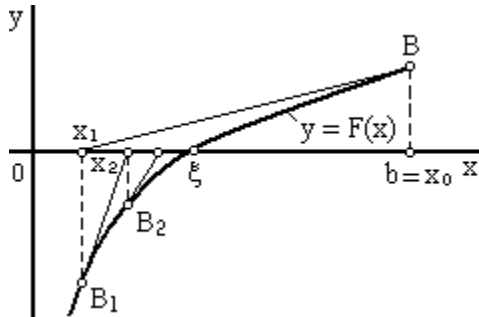


Рисунок 1.4 - Метод Ньютона

$$y - f(x_0) = f'(x_0)(x - x_0).$$

Звідси знайдемо наступне наближення кореня  $x_1$  як абсцису точки перетинання дотичної з віссю  $Ox$  ( $y = 0$ )

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

Аналогічно можуть бути знайдені і наступні наближення як точки притинання з віссю абсцис дотичних, проведених у точках  $B_1, B_2$  і так далі. Формула для  $i + 1$  наближення має вид:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}. \quad (1.7)$$

Для закінчення ітераційного процесу може бути використана або умова  $|f(x_i)| < \varepsilon$ , або умова близькості двох послідовних наближень  $|x_i - x_{i-1}| < \varepsilon$ .

Ітераційний процес сходиться якщо  $f(x_0) \cdot f''(x_0) > 0$ .

### Метод простої ітерації

Часто метод ітерації називають **методом послідовних наближень**. Для використання методу ітерації вихідне нелінійне рівняння  $f(x) = 0$  заміняється **рівноважним** рівнянням

$$x = \varphi(x). \quad (1.8)$$

Нехай відомо початкове наближення кореня  $x = x_0$ . Підставляючи це значення в праву частину рівняння (1.8), одержимо нове наближення:

$$x_1 = \varphi(x_0).$$

Далі, підставляючи щораз нове значення кореня в (1.8), одержуємо послідовність значень:

$$x_{i+1} = \varphi(x_i), \quad (i = 0, 1, \dots). \quad (1.9)$$

Геометрично метод ітерації може бути пояснений у такий спосіб. Побудуємо на площині  $xOy$  графіки функцій  $y = x$  і  $y = \varphi(x)$ . Кожний дійсний корінь  $\xi$  рівняння (1.8) є абсцисою точки перетинання  $M$  кривої  $y = \varphi(x)$  з прямою  $y = x$  (рис. 1.5, а).

Починаючи від деякої точки  $A_0 [x_0, \varphi(x_0)]$ , будуємо ламану  $A_0B_1A_1B_2A_2\dots$  («сходи»), ланки якої поперемінно рівнобіжні осі  $Ox$  та осі  $Oy$ , вершини  $A_0, A_1, A_2, \dots$  лежать на кривій  $y = \varphi(x)$ , а вершини  $B_1, B_2, B_3, \dots, B_n$  – на прямій  $y = x$ .

Спільні абсциси точок  $A_1$  і  $B_1$ ,  $A_2$  і  $B_2$ , ..., очевидно, являють собою відповідно послідовні наближення  $x_1, x_2, \dots$  кореня  $\xi$ .

Можливий також інший вигляд ламаної  $A_0B_1A_1B_2A_2\dots$  – «спіраль» (рис. 1.5, б). Рішення у вигляді «сходів» утворюється, якщо похідна  $\varphi'(x)$  позитивна, а рішення у вигляді «спіралі», якщо  $\varphi'(x)$  негативна.

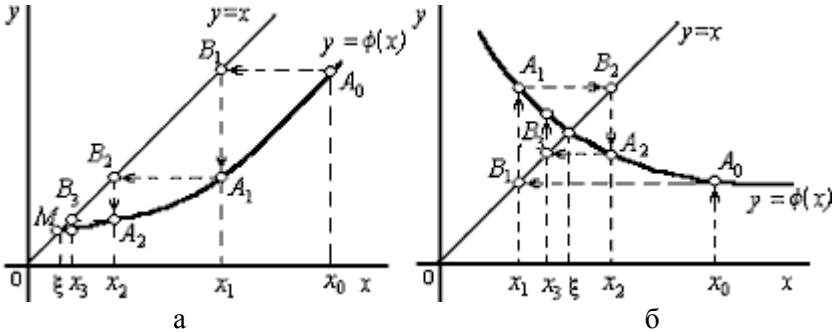


Рисунок 1.5 - Збіжні ітераційні процеси

На рис. 1.5 крива  $y = \varphi(x)$  в окозі кореня  $\xi$  – полого, тобто  $|\varphi'(x)| < 1$ , і процес ітерації сходиться. Проте, якщо роздивитися випадок, де  $|\varphi'(x)| > 1$ , то процес ітерації може бути розбіжним (рис. 1.6). Тому для практичного застосування методу ітерації потрібно з'ясувати достатні умови збіжності ітераційного процесу.

Нехай функція  $\varphi(x)$  визначена і диференційована на відрізку  $[a, b]$ , причому усі її значення  $\varphi(x) \in [a, b]$ .

Тоді, якщо існує правильний дріб  $q$  такий, що

$$|\varphi'(x)| \leq q < 1 \text{ при } a < x < b,$$

то:

- процес ітерації  $x_{i+1} = \varphi(x_i)$ , ( $i = 0, 1, \dots$ ) сходиться незалежно від початкового значення  $x_0 \in [a, b]$ ;

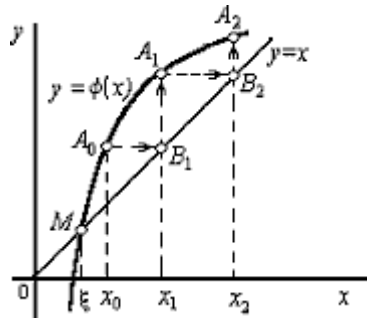


Рисунок 1.6 - Розбіжний ітераційний процес

- **граничне значення**  $\xi = \lim_{n \rightarrow \infty} x_n$  є єдиним коренем рівняння  $x = \varphi(x)$  на відрізку  $[a, b]$ .

**Примітка.** За число  $q$  можна прийняти найменше значення або нижню грань модуля похідної  $|\varphi'(x)|$  при  $a < x < b$ .

## 1.2 Порядок виконання лабораторної роботи

**1.2.1** Побудувати графік функції  $f(x)$  за варіантом (табл. 1.1) і приблизно визначити один із коренів рівняння.

**1.2.2** Вирішити рівняння  $f(x) = 0$  із точністю  $\epsilon = 10^{-5}$  за допомогою операторів і функцій панелі програмування MathCAD методами: ділення відрізка навпіл, хорд, Ньютона (дотичних), простих ітерацій.

Визначити число ітерацій у кожному методі.

**1.2.3** Вирішити рівняння  $f(x) = 0$  із точністю  $\epsilon = 10^{-5}$  за допомогою убудованої функції MathCAD *root* і порівняти результати із попередніми (пп. 1.2.2).

**Таблиця 1.1 - Варіанти індивідуальних завдань**

№	$f(x)$	$x \in [a; b]$
1	$e^{x-1} - x^3 - x$	[0; 1]
2	$x - 1 / (3 + \sin(3.6x))$	[0; 1]
3	$\arccos(x) - x^3$	[0; 1]
4	$2x^2 - \arcsin(x)$	(0; 1)
5	$3x - 14 + e^x - e^{-x}$	[1; 3]
6	$2x^2 + 1.2 - \cos(x) - 2$	[0; 1]
7	$\cos(2/x) - 2\sin(1/x) + 1/x$	[1; 2]
8	$0.1x^2 - x \ln(x)$	[1; 2]
9	$0.25x^3 + x - 2$	[0; 2]
10	$\arccos((1-x^2)/(1+x^2)) - x$	[2; 3]
11	$3x - 4 \ln(x) - 5$	[2; 4]
12	$e^x - e^{-x} - 2$	[0; 1]
13	$1 - x - \operatorname{tg}(x)$	[0; 1]
14	$1 - x + \sin(x) - \ln(1+x)$	[0; 2]

## Продовження таблиці 1.1

№	$f(x)$	$x \in [a; b]$
15	$x^5 - x - 1$	[1;2]
16	$3\sin(2x) - \cos(x)$	[1;2]
17	$x^4 - x^2 - 2$	[0;2]
18	$\cos(x^2) - 7x$	[0;1]
19	$3\sin(x) + x - 1$	[0;2]
20	$10 + 66x - 3x^3$	[2;5]
21	$3xe^{-x} - 1$	[0;1]
22	$2\sin(6x)$	(0;1)
23	$(e^x - 1.5)/(x^2 + 1)$	[0;1]
24	$e^x/(e^x + 1) - x$	[0;2]
25	$\cos(3x/2) + \sin(2x)$	[0;2]
26	$\ln(x) - 2x + 4$	[2;4]
27	$0.3x^4 - \cos(3x)$	[0;2]
28	$2x - 3\cos(4x)$	[0;1]
29	$x^3 - \cos(5x)$	[0;1]
30	$\cos(2x) - x$	[0;2]
31	$3\sin(x) - 2\cos^2(x)$	[0;2]
32	$\sin^3(x) + 3\sin(x) - 2$	[0;2]
33	$x^4 - 4x^2 - 4$	[0;1]
34	$\ln(x) + 2\lg(x)$	(0;3]
35	$\operatorname{tg}(x)^2 - 2\operatorname{tg}(x) + 1$	[0;1]
36	$x^4 - 6x^2 + 7$	[1;2]
37	$x^3 - 2x^2 - 1$	[0;3]
38	$\ln(x^2) - \cos(x)$	[1;3]
39	$\sin(x+2) - x^2 + 2x$	[0;2]
40	$\lg(x) + \cos(x)$	[1;3]

## 1.3 Зміст звіту

1.3.1 Опис методів рішення рівнянь.

1.3.2 Розрахунки в системі MathCAD.

1.3.3 Висновки.

## 1.4 Контрольні запитання для самоперевірки

1.4.1 Які методи рішення нелінійних рівнянь вам відомі?

1.4.2 У яких випадках необхідно використовувати ітераційні методи?

1.4.3 Яким умовам повинна відповідати функція  $f(x)$  і що вони гарантують?

1.4.4 Що значить вирішити рівняння ітераційним методом?

1.4.5 З яких етапів складається задача знаходження нуля функції  $f(x)$  ітераційним методом?

1.4.6 Назвіть способи відділення коренів.

1.4.7 У чому складається ітераційний процес?

1.4.8 Сутність методу половинного розподілу.

1.4.9 Сутність методу хорд.

1.4.10 Який із кінців відрізка  $[a, b]$  у методі хорд вважається нерухомим?

1.4.11 Умова закінчення ітераційного процесу в методі хорд?

1.4.12 Сутність методу Ньютона.

1.4.13 Як вибрати початкове наближення для методу Ньютона?

1.4.14 Як у MathCAD організувати ітераційний процес?

1.4.15 Що впливає на швидкість збіжності ітераційного процесу?

1.4.16 Сутність методу ітерації, як ще називають цей метод.

1.4.17 Які види ітераційних процесів вам відомі?

1.4.18 Сформулюйте достатні умови збіжності методу ітерації?





рішення якої знаходять по рекурентних формулах:

$$x_n = \beta_n, \quad x_i = \beta_i - \sum_{j=i+1}^n \alpha_{i,j} x_j, \quad (i = n-1, n-2, \dots, 1). \quad (2.5)$$

У матричному записі це означає, що спочатку (прямий хід методу Гауса) елементарними операціями над рядками призводять розширену матрицю системи до ступінчастого виду:

$$A_p = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & \alpha_{12} & \dots & \alpha_{1n} & \beta_1 \\ 0 & 1 & \dots & \alpha_{2n} & \beta_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & \beta_n \end{bmatrix},$$

а потім (зворотний хід методу Гауса) цю ступінчасту матрицю перетворюють так, щоб у перших  $n$  стовпчиках утворилася одинична матриця:

$$\begin{bmatrix} 1 & 0 & \dots & 0 & x_1 \\ 0 & 1 & \dots & 0 & x_2 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & x_n \end{bmatrix}.$$

Останній  $(n + 1)$  стовпчик цієї матриці містить **рішення системи**.

При рішенні системи лінійних рівнянь методом Гауса рекомендується використовувати наступні **функції MathCAD**.

У MathCAD прямий і зворотний ходи методу Гауса виконує функція  $rref(A)$ .

$rref(A)$

Повертає східчасту форму матриці  $A$

$augment(A, B, \dots)$

Повертає масив, сформований розташуванням  $A$  і  $B$  зліва праворуч. Масиви  $A$  і  $B$  повинні бути скалярами і мати однакове число рядків

$submatrix(A, ir, jr, ic, jc)$

Повертає субматрицю, що складається з всіх елементів, що знаходяться в рядках з  $ir$  по  $jr$  і стовпчиках з  $ic$  по  $jc$  матриці  $A$

$M^{<n>}$

Повертає  $n$ -ний стовпчик масиву  $M$

### Метод ітерацій

Нехай дана лінійна система (2.1). Увівши в матрицю (2.3), систему (2.1) отриману систему коротко можна записати у виді матричного рівняння (2.2). Припускаючи, що діагональні коефіцієнти

$$a_{ij} \neq 0 \quad (i = 1, 2, \dots, n),$$

вирішимо перше рівняння системи (2.1) щодо  $x_1$ , друге – щодо  $x_2$  і т.д. В результаті отримаємо еквівалентну систему

$$\begin{cases} x_1 = \beta_1 + \alpha_{12}x_2 + \alpha_{13}x_3 + \dots + \alpha_{1n}x_n, \\ x_2 = \beta_2 + \alpha_{21}x_1 + \alpha_{23}x_3 + \dots + \alpha_{2n}x_n, \\ \dots \\ x_n = \beta_n + \alpha_{n1}x_1 + \alpha_{n2}x_2 + \dots + \alpha_{n,n-1}x_{n-1} \end{cases} \quad (2.6)$$

де  $\beta_i = \frac{b_i}{a_{ii}}$ ;  $\alpha_{ij} = -\frac{a_{ij}}{a_{ii}}$  при  $i \neq j$ ;

$\alpha_{ij} = 0$  при  $i = j$  ( $i, j = 1, 2, \dots, n$ ).

Введемо матриці

$$\alpha = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2n} \\ \dots & \dots & \dots & \dots \\ \alpha_{n1} & \alpha_{n2} & \dots & \alpha_{nn} \end{bmatrix} \quad \text{та} \quad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_n \end{bmatrix},$$

і запишемо систему (2.6) в матричній формі  $\mathbf{x} = \beta + \alpha\mathbf{x}$ , а будь-яке  $(k+1)$  наближення обчислимо по формулі:

$$\mathbf{x}^{(k+1)} = \beta + \alpha\mathbf{x}^{(k)}. \quad (2.7)$$

Запишемо формули наближень у розгорнутому виді:

$$\begin{cases} x_i^{(0)} = \beta_i, \\ x_i^{(k+1)} = \beta_i + \sum_{j=1}^n \alpha_{ij}x_j^{(k)} \\ \left( \alpha_{ii} = 0; i = 1, \dots, n; k = 0, 1, 2, \dots \right). \end{cases} \quad (2.8)$$

Приведемо достатню умову збіжності методу ітерацій.

**Теорема.** Процес ітерації для приведеної лінійної системи (2.6) сходиться до єдиного її рішення, якщо будь-яка канонічна норма матриці  $\alpha$  менше одиниці, тобто для ітераційного процесу (2.7) достатньою є умова:

$$\|\alpha\| < 1. \quad (2.9)$$

**Наслідок 1.** Процес ітерації для системи (2.6) сходиться, якщо:

$$- \|\alpha\|_m = \max_i \sum_j |\alpha_{ij}| < 1 \text{ (} m\text{-норма або невизначена норма),}$$

або

$$- \|\alpha\|_l = \max_j \sum_i |\alpha_{ij}| < 1 \text{ (} l\text{-норма або норма } L1),$$

або

$$- \|\alpha\|_k = \sqrt{\sum_{i,j} |\alpha_{ij}|^2} < 1 \text{ (} k\text{-норма або Евклідова норма).}$$

**Наслідок 2.** Для системи (2.1) процес ітерації сходиться, якщо виконані нерівності:

$$- |a_{ii}| > \sum_{j=1}^n |a_{ij}| \quad (i = 1, 2, \dots, n),$$

або

$$- |a_{jj}| > \sum_{i=1}^n |a_{ij}| \quad (j = 1, 2, \dots, n),$$

де штрих у знака суми означає, що при підсумовуванні пропускаються значення  $i = j$ , тобто збіжність має місце, якщо модулі діагональних елементів матриці  $A$  системи (2.1), або для кожного рядка перевищують суму модулів недиагональних елементів цього рядка, або ж для кожного стовпчика перевищують суму модулів недиагональних елементів цього стовпчика.

У MathCAD існують спеціальні функції для обчислення норм матриць:

<b><i>normi(A)</i></b>	Повертає невизначену норму матриці $A$ .
<b><i>norml(A)</i></b>	Повертає $L1$ норму матриці $A$ .
<b><i>norme(A)</i></b>	Повертає Евклідову норму матриці $A$ .

У якості умови закінчення ітераційного процесу можна прийняти:

$$\frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k+1)}\|} \leq \varepsilon,$$

де  $\varepsilon$  – задана похибка наближеного рішення  $x \approx x^{(k+1)}$ .

### Метод Зейделя

Метод Зейделя являє собою модифікацію методу ітерацій. Основна його ідея полягає в тому, що при обчисленні  $(k + 1)$  наближення невідомої  $x_i$  враховуються вже обчислені раніше  $(k + 1)$  наближення невідомих  $x_1, x_2, \dots, x_{i-1}$ .

Нехай отримана еквівалентна система (2.6). Виберемо довільно початкові наближення коренів  $x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}$ . Далі, припускаючи, що  $k$ -ті наближення  $x_n^{(k)}$  коренів відомі, відповідно до Зейделя, будемо будувати  $(k + 1)$  наближення коренів по формулах:

$$\begin{aligned} x_1^{(k+1)} &= \beta_1 + \alpha_{12}x_2^{(k)} + \alpha_{13}x_3^{(k)} + \dots + \alpha_{1n}x_n^{(k)}, \\ x_2^{(k+1)} &= \beta_2 + \alpha_{21}x_1^{(k+1)} + \alpha_{23}x_3^{(k)} + \dots + \alpha_{2n}x_n^{(k)}, \\ &\dots \\ x_n^{(k+1)} &= \beta_n + \alpha_{n1}x_1^{(k+1)} + \alpha_{n2}x_2^{(k+1)} + \dots + \alpha_{nn}x_n^{(k)} \quad (k = 0, 1, 2, \dots). \end{aligned} \quad (2.10)$$

Зауважимо, що зазначені вище умови збіжності для простої ітерації залишаються вірними для ітерації по методу Зейделя. Звичайно метод Зейделя дає кращу збіжність, чим метод простої ітерації, але призводить до більш громіздких обчислень.

## 2.2 Порядок виконання лабораторної роботи

Вирішити систему лінійних рівнянь за варіантом (табл. 2.1) за допомогою матричних операцій і функції *Isolve*; методом Гауса; методом простих ітерацій; методом Зейделя.

Ітераційними методами рішення задачі знайти з точністю  $\varepsilon=10^{-5}$ . Оцінити похибку обчислень.

**Вказівка.** Для виконання достатньої умови збіжності скористатися перестановкою рядків у вихідній системі рівнянь.

Таблиця 2.1 - Варіанти індивідуальних завдань

№2	Система рівнянь	№2	Система рівнянь
1	$4x_1 + 20x_2 + x_3 - 24 = 0$ $16x_1 + 2x_2 - 2x_4 + 13 = 0$ $-4x_1 + 4x_3 + 32x_4 = 0$ $2x_1 + 10x_3 - 7 = 0$	9	$4x_1 - 5x_2 + 40x_3 - 19 = 0$ $10x_1 - 4x_2 + 50x_4 = 0$ $32x_1 + 4x_3 - 4x_4 - 34 = 0$ $32x_2 - 9x_4 + 49 = 0$
2	$3x_1 + 12x_2 - x_3 - 18 = 0$ $-5x_1 + 2x_2 + 32x_4 + 15 = 0$ $2x_1 + 16x_3 - 3x_4 = 0$ $12x_1 + 3x_2 - 21 = 0$	10	$4x_1 + 2x_2 + 32x_3 + 19 = 0$ $2x_1 + 30x_2 - 4x_4 - 39 = 0$ $36x_1 + 4x_3 - 5x_4 - 40 = 0$ $11x_3 + 40x_4 - 31 = 0$
3	$2x_1 + 16x_2 - 3x_3 - 9 = 0$ $-8x_1 + x_2 + x_3 + x_4 - 98 = 0$ $25x_1 - 2x_3 - 7x_4 - 5 = 0$ $-3x_2 + 20x_3 + 7 = 0$	11	$9x_1 + 40x_2 + 2x_3 - 81 = 0$ $12x_1 - 4x_2 + 96x_4 - 119 = 0$ $-4x_1 + 64x_3 + 8x_4 + 15 = 0$ $36x_1 + 9x_4 - 7 = 0$
4	$5x_1 - 2x_2 + x_3 + x_4 - 27 = 0$ $4x_1 + 25x_2 - 3x_4 - 34 = 0$ $20x_1 + 2x_3 - 7x_4 + 28 = 0$ $-9x_3 + 40x_4 - 5 = 0$	12	$7x_1 - 5x_2 + 64x_3 - 18 = 0$ $9x_1 + 50x_2 - 4x_4 = 0$ $9x_2 - 7x_3 + 80x_4 - 128 = 0$ $40x_1 + 11x_2 + 19 = 0$
5	$-7x_1 + 2x_2 + 40x_3 - 21 = 0$ $9x_1 - 5x_2 + 50x_4 + 14 = 0$ $25x_1 + 4x_3 - x_4 - 13 = 0$ $32x_2 + 9x_4 - 21 = 0$	13	$11x_1 + 64x_2 - 2x_3 + 34 = 0$ $50x_1 + 3x_2 - 12x_4 = 0$ $13x_2 - 9x_3 + 100x_4 - 131 = 0$ $17x_1 + 80x_3 - 85 = 0$
6	$8x_1 + 40x_2 - 3x_3 - 28 = 0$ $-7x_1 + 5x_2 + 50x_4 = 0$ $8x_1 + 64x_3 - 11x_4 - 18 = 0$ $32x_1 + 5x_4 - 12 = 0$	14	$15x_1 + 80x_2 - 4x_3 - 93 = 0$ $64x_1 + 7x_2 - 5x_4 - 131 = 0$ $11x_2 - 8x_3 + 128x_4 + 34 = 0$ $37x_2 + 100x_3 - 125 = 0$
7	$-9x_1 + 4x_2 + 64x_3 - 24 = 0$ $10x_1 + 50x_2 - 4x_4 + 5 = 0$ $-14x_2 + 7x_3 + 80x_4 - 14 = 0$ $40x_1 + 9x_2 - 29 = 0$	15	$17x_1 + 100x_2 - 9x_3 = 0$ $80x_1 - 7x_2 - 5x_4 + 79 = 0$ $21x_2 + 128x_3 - 4x_4 - 139 = 0$ $19x_3 + 256x_4 + 54 = 0$
8	$-8x_1 + 64x_2 + 5x_3 - 37 = 0$ $50x_1 - 13x_2 + 2x_4 - 38 = 0$ $17x_2 - 9x_3 + 100x_4 = 0$ $-11x_1 + 80x_3 - 115 = 0$	16	$4x_1 - x_2 + 20x_3 - 38 = 0$ $18x_1 + 3x_2 - 2x_4 + 14 = 0$ $10x_2 + x_3 - x_4 - 15 = 0$ $4x_2 + 20x_4 - 29 = 0$

Продовження таблиці 2.1

№	Система рівнянь	№	Система рівнянь
17	$-13x_1 + 80x_2 + 2x_3 - 64 = 0$ $64x_1 + 9x_2 - 5x_4 - 29 = 0$ $12x_2 - 9x_3 + 128x_4 = 0$ $27x_2 + 100x_3 - 231 = 0$	25	$3x_1 + 20x_2 - 2x_3 - 41 = 0$ $5x_1 - 4x_2 + 20x_4 + 19 = 0$ $5x_2 + 32x_3 - 3x_4 - 34 = 0$ $12x_1 + 3x_4 - 29 = 0$
18	$-13x_1 + 100x_2 + 9x_3 + 128 = 0$ $80x_1 + 10x_2 - 5x_4 - 34 = 0$ $-14x_2 + 128x_3 + 7x_4 - 95 = 0$ $31x_3 + 256x_4 + 69 = 0$	26	$4x_1 + 25x_2 - x_3 - 17 = 0$ $6x_1 + 5x_2 + 40x_4 = 0$ $25x_1 + 3x_3 + 4x_4 + 34 = 0$ $-5x_2 + 30x_3 - 9 = 0$
19	$x_1 - 2x_2 + 16x_3 - 31 = 0$ $10x_1 - x_2 + x_4 = 0$ $12x_2 + x_3 - x_4 + 28 = 0$ $2x_2 + 16x_4 - 29 = 0$	27	$9x_1 - 2x_2 + 36x_3 - 19 = 0$ $4x_1 + 25x_2 - 3x_4 + 18 = 0$ $40x_1 + 5x_3 - 4x_4 - 44 = 0$ $11x_3 + 40x_4 - 21 = 0$
20	$2x_1 + 20x_2 - 3x_3 - 39 = 0$ $4x_1 - 2x_2 + 24x_4 = 0$ $2x_2 + 16x_3 - x_4 + 25 = 0$ $12x_1 + 3x_4 - 18 = 0$	28	$9x_1 - 2x_2 + 40x_3 - 78 = 0$ $11x_1 - 3x_2 + 50x_4 + 114 = 0$ $30x_1 - 4x_3 + 5x_4 + 21 = 0$ $32x_2 + 8x_4 - 40 = 0$
21	$2x_1 + 16x_2 - x_3 - 32 = 0$ $3x_1 - 8x_2 + 60x_4 + 64 = 0$ $4x_1 + 24x_3 - 3x_4 = 0$ $12x_1 + 3x_2 - 45 = 0$	29	$2x_1 + 40x_2 + 5x_3 - 42 = 0$ $4x_1 - 9x_2 + 72x_4 - 88 = 0$ $4x_1 + 64x_3 + 8x_4 - 119 = 0$ $36x_1 + 9x_4 - 54 = 0$
22	$5x_1 - 2x_2 + 40x_3 - 39 = 0$ $4x_1 + 32x_2 - 6x_4 = 0$ $7x_1 + 3x_3 + 32x_4 - 21 = 0$ $20x_1 + 4x_3 + 19 = 0$	30	$8x_1 - 3x_2 + 64x_3 - 131 = 0$ $-7x_1 + 50x_2 + 5x_4 + 84 = 0$ $12x_2 - 9x_3 + 80x_4 - 52 = 0$ $40x_1 + 9x_2 - 78 = 0$
23	$5x_1 + 30x_2 - 3x_3 - 17 = 0$ $-8x_1 + 5x_2 + 40x_4 - 31 = 0$ $24x_1 + 3x_3 - 4x_4 - 39 = 0$ $7x_2 + 25x_3 - 8 = 0$	31	$7x_1 + 64x_2 - 2x_3 - 111 = 0$ $50x_1 + 5x_2 - 8x_4 - 98 = 0$ $18x_2 + 5x_3 + 112x_4 - 219 = 0$ $15x_1 + 80x_3 + 31 = 0$
24	$2x_1 + x_2 + 2x_3 + 3x_4 - 8 = 0$ $3x_1 + 3x_3 - 6 = 0$ $2x_1 - x_2 - 4x_4 - 4 = 0$ $x_1 + 2x_2 - x_3 + 2x_4 - 4 = 0$	32	$7x_1 + 64x_2 - 2x_3 - 111 = 0$ $50x_1 + 5x_2 - 8x_4 - 98 = 0$ $18x_2 + 5x_3 + 112x_4 - 219 = 0$ $15x_1 + 80x_3 + 31 = 0$

Продовження таблиці 2.1

№	Система рівнянь	№	Система рівнянь
33	$2x_1 + x_2 - 5x_3 + x_4 + 4 = 0$ $x_1 - 3x_2 - 6x_4 + 7 = 0$ $2x_1 - x_3 + 2x_4 - 2 = 0$ $x_1 + 4x_2 - 7x_3 + 6x_4 + 2 = 0$	37	$x_1 + 2x_2 + 3x_3 + 4x_4 - 26 = 0$ $2x_1 + 3x_2 + 4x_3 + x_4 - 34 = 0$ $3x_1 + 4x_2 + x_3 + 2x_4 - 26 = 0$ $4x_1 + x_2 + 2x_3 + 3x_4 - 26 = 0$
34	$x_1 + 2x_2 + 3x_3 + 4x_4 - 22 = 0$ $2x_1 + 3x_2 + x_3 + 2x_4 - 17 = 0$ $x_1 + 4x_2 + x_3 - x_4 - 8 = 0$ $x_1 - 2x_3 - 3x_4 - 7 = 0$	38	$2x_1 - x_2 + 4x_3 + x_4 - 66 = 0$ $2x_2 - 6x_3 + x_4 + 63 = 0$ $8x_1 - 3x_2 + 6x_3 - 5x_4 - 146 = 0$ $2x_1 - 7x_2 + 6x_3 - x_4 - 80 = 0$
35	$2x_1 - 8x_2 - 3x_3 - 2x_4 + 18 = 0$ $x_1 - 2x_2 + 3x_3 - 2x_4 - 28 = 0$ $x_2 + x_3 + 2x_4 - 10 = 0$ $11x_2 + x_3 + 2x_4 - 21 = 0$	39	$6x_1 - x_2 + 10x_3 - 4x_4 - 158 = 0$ $2x_1 + x_2 + 10x_3 + 7x_4 - 128 = 0$ $3x_1 - 2x_2 - 2x_3 - x_4 - 7 = 0$ $x_1 - 12x_2 + 2x_3 - x_4 - 17 = 0$
36	$9x_1 + 10x_2 - 7x_3 - x_4 - 23 = 0$ $7x_1 - x_3 - 5x_4 - 37 = 0$ $5x_1 - 2x_3 + x_4 - 22 = 0$ $4x_1 + x_2 + 2x_3 + 3x_4 - 26 = 0$	40	$2x_1 + 2x_2 + 6x_3 + x_4 - 15 = 0$ $-x_2 + 2x_3 + x_4 - 18 = 0$ $4x_1 - 3x_2 + x_3 - 5x_4 - 37 = 0$ $3x_1 - 5x_2 + x_3 - x_4 - 30 = 0$

### 2.3 Зміст звіту

2.3.1 Опис методів рішення рівнянь.

2.3.2 Розрахунки в системі MathCAD.

2.3.3 Висновки.

### 2.4 Контрольні запитання для самоперевірки

2.4.1 Назвіть точні методи рішення систем лінійних рівнянь?

2.4.2 Які функції Mathcad використовуються для їхньої реалізації?

2.4.3 Сформулюйте достатні умови збіжності методу ітерації для систем лінійних рівнянь.

2.4.4 Які види норм матриць вам відомі і як їх обчислювати?

2.4.5 Назвіть особливості методу Зейделя.

2.4.6 Назвіть функції для рішення систем рівнянь у Mathcad і особливості їхньої застосування.

### 3 ЛАБОРАТОРНА РОБОТА № 3 РІШЕННЯ СИСТЕМ НЕЛІНІЙНИХ РІВНЯНЬ

**Мета роботи:** засвоєння методів рішення систем нелінійних рівнянь та способів їх реалізації за допомогою системи MathCAD.

#### 3.1 Теоретичний матеріал

Нехай для обчислення невідомих  $x_1, x_2, \dots, x_n$  потрібно вирішити систему  $n$  нелінійних рівнянь

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0, \\ f_2(x_1, x_2, \dots, x_n) &= 0, \\ &\dots \\ f_n(x_1, x_2, \dots, x_n) &= 0. \end{aligned} \tag{3.1}$$

На відміну від систем лінійних рівнянь для систем нелінійних рівнянь не відомі прямі методи рішення. Лише в окремих випадках систему можна вирішити безпосередньо. Наприклад, для системи з двох рівнянь іноді вдається виразити одну невідому через іншу й, у такий спосіб, звести задачу до рішення одного нелінійного рівняння щодо одного невідомого. Тому ітераційні методи для нелінійних систем набувають особливої актуальності.

#### Метод простої ітерації

Систему рівнянь (3.1) приведемо до виду

$$\begin{aligned} x_1 &= f_1(x_1, x_2, \dots, x_n), \\ x_2 &= f_2(x_1, x_2, \dots, x_n), \\ &\dots \\ x_n &= f_n(x_1, x_2, \dots, x_n). \end{aligned} \tag{3.2}$$

Алгоритм рішення цієї системи методом простої ітерації нагадує метод Гауса-Зейделя, використаний для рішення систем лінійних рівнянь.

Нехай у результаті попередньої ітерації отримані значення невідомих  $x_1 = a_1, x_2 = a_2, \dots, x_n = a_n$ . Тоді вираження для невідомих на такій ітерації мають вид

$$x_1 = f_1(a_1, a_2, \dots, a_n),$$

$$x_2 = f_2(x_1, a_2, \dots, a_n),$$

...

$$x_i = f_i(x_1, \dots, x_{i-1}, a_i, \dots, a_n),$$

...

$$x_n = f_n(x_1, \dots, x_{n-1}, a_n).$$

Ітераційний процес продовжується доти, поки зміни всіх невідомих у двох послідовних ітераціях не стануть малими, тобто абсолютні розміри їхніх різниць не стануть меншими заданого малого числа.

При використанні методу простої ітерації успіх багато в чому визначається удалим вибором початкових наближень невідомих: вони повинні бути достатньо близькими до істинного рішення. У протилежному випадку ітераційний процес може не зійтися.

### Метод Ньютона

Цей метод має набагато більш швидку збіжність, чим метод простої ітерації. У випадку одного рівняння  $f(x)=0$  алгоритм методу Ньютона буде легко отриманий шляхом запису рівняння дотичної до кривої  $y = f(x)$ .

У основі методу Ньютона для системи рівнянь лежить використання розкладання функцій  $f_i(x_1, x_2, \dots, x_n)$  у ряд Тейлора, причому члени, що містять другі (і більш високих порядків) похідні, не враховуються.

Запишемо систему рівнянь (3.1) у векторній формі

$$f(x) = 0 \tag{3.3}$$

де

$$f = \begin{bmatrix} f_1 \\ f_2 \\ \dots \\ f_n \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}.$$

Для рішення системи (3.3) будемо користуватися методом послідовних наближень.

Якщо відомо наближення  $x^{(k)} = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})$  одного з



або в стислому записі

$$f'(x) = W(x) = \left[ \frac{\partial f_i}{\partial x_j} \right],$$

$$i = 1, 2, \dots, n; j = 1, 2, \dots, n,$$

Тому формула (3.7) може бути записана в такому виді:

$$f(x^{(k)}) + W(x^{(k)})\Delta x^{(k)} = 0$$

Якщо  $\det W(x) = \det \left[ \frac{\partial f}{\partial x} \right] \neq 0$ , то  $\Delta x^{(k)} = -W^{-1}(x^{(k)})f(x^{(k)})$ .

Звідси очевидно, що метод Ньютона рішення системи (3.1) складається в побудові ітераційної послідовності:

$$x^{(k+1)} = x^{(k)} - W^{-1}(x^{(k)})f(x^{(k)}) = 0, (k = 0, 1, 2, \dots). \quad (3.8)$$

Якщо всі поправки стають достатньо малими, рахунок припиняється. Інакше нові значення  $x_i$  використовуються як наближені значення коренів, і процес повторюється доти, поки не буде знайдено рішення або не стане ясно, що одержати його не вдасться.

### Метод градієнту (метод найшвидшого спуску)

У методі найшвидшого спуску рішення системи (3.3) шукають у виді

$$x^{(p+1)} = x^{(p)} - \mu_p W'_p f^{(p)}. \quad (3.9)$$

Тут  $x^{(p)}$  і  $x^{(p+1)}$  – вектори невідомих на  $p$  і  $p+1$  кроках ітерацій; вектор нев'язок на  $p$ -му кроку –  $f^{(p)} = f(x^{(p)})$ ;  $W'_p$  – транспонована матриця Якобі на  $p$ -му кроку;

$$\mu_p = \frac{(f^{(p)}, W'_p W'_p f^{(p)})}{(W'_p W'_p f^{(p)}, W'_p W'_p f^{(p)})};$$

$$W_p = \left[ \frac{\partial f_i^{(k)}}{\partial x_j^{(k)}} \right], \text{ при } i = 1, 2, \dots, n; j = 1, 2, \dots, n.$$

**Зауваження.** При рішенні системи нелінійних рівнянь методом градієнту матрицю Якобі необхідно перераховувати на кожному кроку (ітерації).

### 3.2 Порядок виконання лабораторної роботи

Вирішити систему нелінійних рівнянь за варіантом (табл. 3.1) методами: простих ітерацій; Ньютона; градієнту.

Рішення задачі знайти з точністю  $\varepsilon=10^{-5}$ . Оцінити похибку обчислень.

**Таблиця 3.1 - Варіанти індивідуальних завдань**

№	Система рівнянь	№	Система рівнянь
1	$\sin(x_1+x_2)-x_2-1.2=0$ $2x_1+\cos(x_2)-2=0$	15	$\sin(0.5x_1+x_2)-1.2x_1-1=0$ $(x_1)^2+(x_2)^2-1=0$
2	$\cos(x_1-1)+x_2-0.5=0$ $\sin(x_1)+2x_2-2=0$	16	$\tan(x_1x_2+0.3)-(x_1)^2=0$ $0.9(x_1)^2+2(x_2)^2-1=0$
3	$\sin(x_1)+2x_2-2=0$ $\cos(x_1)+x_2-1.5=0$	17	$\sin(x_1+x_2)-1.3x_1-1=0$ $(x_1)^2+0.2(x_2)^2-1=0$
4	$\cos(x_1)+x_2-1.5=0$ $2x_1-\sin(x_2-0.5)-1=0$	18	$\tan(x_1x_2)-(x_1)^2=0$ $0.8(x_1)^2+2(x_2)^2-1=0$
5	$\sin(x_1+1.5)-x_2+2.9=0$ $\cos(x_2-2)+x_1=0$	19	$\sin(x_1+x_2)-1.5x_1-0.1=0$ $3(x_1)^2+(x_2)^2-1=0$
6	$\cos(x_1+0.5)+x_2-0.8=0$ $\sin(x_2)-2x_1-1.6=0$	20	$\tan(x_1x_2+0.2)-(x_1)^2=0$ $0.7(x_1)^2+2(x_2)^2-1=0$
7	$\sin(x_1-1)+x_2-0.1=0$ $x_1-\sin(x_2+1)-0.8=0$	21	$\sin(x_1+x_2)-1.2x_1-0.1=0$ $(x_1)^2+(x_2)^2-1=0$
8	$\cos(x_1+x_2)+2x_2=0$ $x_1+\sin(x_2)-0.6=0$	22	$\tan(x_1x_2+0.2)-(x_1)^2=0$ $0.6(x_1)^2+2(x_2)^2-1=0$
9	$\cos(x_1+0.5)-x_2-2=0$ $\sin(x_2)+2x_1-1=0$	23	$\sin(x_1+x_2)-x_1+0.1=0$ $x_2-\cos(3x_1)+0.1=0$
10	$\sin(x_1+x_2)-x_2-1.5=0$ $x_1+\cos(x_2-0.5)-0.5=0$	24	$\cos(x_2-2)+x_1=0$ $\sin(x_1+0.5)-x_2+2.9=0$
11	$\sin(x_2+1)+x_1-1.2=0$ $2(x_1)^2+x_2-2=0$	25	$\sin(x_1)+2x_2-2=0$ $\cos(x_2-1)+x_1-0.7=0$
12	$\cos(x_2-1)+x_1-0.5=0$ $x_2-\cos(x_1)-3=0$	26	$\sin(x_2)+x_1+0.4=0$ $2x_2-\cos(x_1+1)=0$
13	$\tan(x_1x_2+0.4)-(x_1)^2=0$ $0.6(x_1)^2+2(x_2)^2-1=0$	27	$\sin(x_1+0.5)-x_2-1=0$ $\cos(x_2-2)+x_1=0$
14	$\cos(x_1+0.5)+x_2-1=0$ $\sin(x_2)-2x_1-2=0$	28	$\sin(x_1+2)-x_2-1.5=0$ $\cos(x_2-2)+x_1-0.5=0$

### Продовження таблиці 3.1

29	$\sin(x_1+x_2)-1.6x_1-1=0$ $(x_1)^2+(x_2)^2-1=0$	35	$\sin(x_1)-2x_2-1=0$ $\sin(x_2-1)+x_1-1.3=0$
30	$\tan(x_1x_2+0.1)-(x_1)^2=0$ $(x_1)^2+2(x_2)^2-1=0$	36	$\cos(x_1)+x_2-1.5=0$ $2x_1-\sin(x_2-0.5)-1=0$
31	$\cos(x_2-1)+x_1-0.8=0$ $x_2-\cos(x_1)-2=0$	37	$\sin(x_1-1)+x_2-1.3=0$ $x_1-\sin(x_2+1)-0.8=0$
32	$\cos(x_1-1)+x_2-1=0$ $\sin(x_2)+2x_1-1.6=0$	38	$\cos(x_1+0.5)-x_2-2=0$ $\sin(x_2)-2x_1-1=0$
33	$\sin(x_2+1)-x_1-1=0$ $2x_1+\cos(x_1)-0.5=0$	39	$\cos(x_1-2)+x_2=0$ $\sin(x_2+0.5)-x_1-1=0$
34	$\sin(x_1-1)+x_2-1.5=0$ $x_1-\sin(x_2-1)-1=0$	40	$\cos(x_1+0.5)+x_2-0.8=0$ $\sin(x_2)-2x_1-1.6=0$

### 3.3 Зміст звіту

3.3.1 Опис методів рішення рівнянь.

3.3.2 Розрахунки в системі MathCAD.

3.3.3 Висновки.

### 3.4 Контрольні запитання для самоперевірки

3.4.1 Постановка задачі рішення нелінійних рівнянь. Основні етапи рішення задачі.

3.4.2 Метод простої ітерації: опис методу, умови і швидкість збіжності, критерій закінчення; приведення до виду, зручному для ітерацій.

3.4.3 Метод Ньютона: опис методу.

3.4.4 Хиби методу Ньютона. Модифікації методу Ньютона.

3.4.5 Як знайти початкове наближення для методу Ньютона?

3.4.6 Як знайти початкове наближення для методу градієнта?

3.4.7 Назвіть критерії припинення ітераційного процесу при рішенні системи нелінійних рівнянь методами: Ньютона; найшвидшого спуску?

## 4 ЛАБОРАТОРНА РОБОТА № 4 МЕТОДИ ІНТЕРПОЛЯЦІЇ І АПРОКСИМАЦІЇ

**Мета роботи:** засвоєння методів інтерполяції і апроксимації, правила їх застосування в системі MathCAD.

### 4.1 Теоретичний матеріал

Експериментальні дані, отримані у лабораторних або промислових умовах, являються основою для проведення подальших досліджень. В результаті проведення експерименту дослідник одержує деяку таблицю значень – табл. 4.1.

**Таблиця 4.1 - Таблиця експериментальних значень**

X	x <sub>0</sub>	x <sub>1</sub>	x <sub>2</sub>	...	x <sub>N</sub>
Y	y <sub>0</sub>	y <sub>1</sub>	y <sub>2</sub>	...	y <sub>N</sub>

При обробці експериментальних даних можуть виникнути два типи задач:

- для функції, що задана як таблиця, потрібно обчислити значення даної функції для проміжного значення аргументу. Цей тип задач розв'язується **методом інтерполяції**;
- для функції, що задана як таблиця або графічно, підібрати аналітичну формулу, яка зображує з якоюсь точністю дані значення функції. Такі аналітичні формули називаються емпіричними. Задачі даного типу вирішуються **методом апроксимації**.

**Апроксимація функцій** полягає в наближеній заміні заданої функції  $f(x)$  деякою функцією  $\varphi(x)$  так, щоб відхилення функції  $\varphi(x)$  від  $f(x)$  у заданій області було найменшим. Функція  $\varphi(x)$  при цьому називається **апроксимуючою**.

Необхідність **інтерполяції функцій** в основному пов'язана з двома причинами:

- функція  $f(x)$  має складний аналітичний опис, що викликає певні труднощі при його використанні;
- аналітичний опис функції  $f(x)$  невідомий, тобто  $f(x)$  задана у вигляді таблиці. При цьому необхідно мати аналітичний опис, що приблизно представляє  $f(x)$  (наприклад, для обчислення: значень  $f(x)$  у довільних точках, визначення інтегралів і похідних від  $f(x)$  і т. п.).

### Інтерполяція

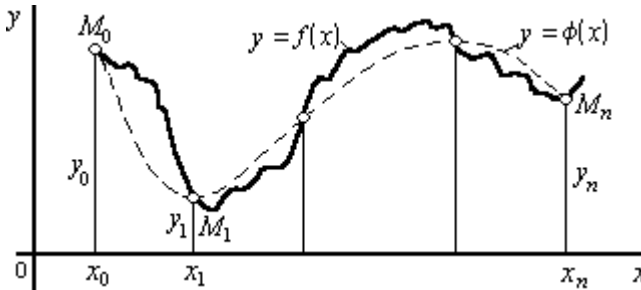
Найпростіша задача **інтерполяції** полягає в наступному. Для заданих точок  $x_i = x_0, x_1, \dots, x_n$ , які називаються **вузлами інтерполяції**, і значень у цих точках деякої функції  $f(x_i) = y_0, y_1, \dots, y_n$  побудувати поліном  $\varphi(x)$  (**інтерполяційний поліном**) ступеня  $n$  у вигляді

$$\varphi(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0, \quad (4.1)$$

який приймає у вузлах інтерполяції  $x_i$  ті ж значення  $y_i$ , що і функція  $f(x_i)$ :

$$\varphi(x_0) = y_0, \varphi(x_1) = y_1, \dots, \varphi(x_n) = y_n, i = 0, 1, \dots, n \quad (4.2)$$

Геометрично це означає (рис. 4.1), що потрібно знайти криву  $\varphi(x)$  деякого визначеного типу, що проходить через задану систему  $M(x_i, y_i)$  ( $i = 0, 1, \dots, n$ ).



**Рисунок 4.1 – Геометричний зміст інтерполяції**

Розрізняють два види інтерполяції:

- **глобальна** – з'єднання всіх точок  $\varphi(x)$  єдиним інтерполяційним поліномом;
- **локальна** – з'єднання точок відрізками прямої (по двох точках), відрізками параболи (по трьох точках).

#### Глобальна інтерполяція

Найпростішим видом **глобальної інтерполяції** є **параболічна інтерполяція**, коли, використовуючи описані вище умови (4.2), для знаходження невідомих  $n+1$  коефіцієнтів  $a_0, a_1, \dots, a_n$  вираження (4.1) одержують систему з  $n+1$  рівнянь:

$$\begin{cases} a_n x_0^n + a_{n-1} x_0^{n-1} + \dots + a_1 x_0 + a_0 = y_0, \\ a_n x_1^n + a_{n-1} x_1^{n-1} + \dots + a_1 x_1 + a_0 = y_1, \\ \dots \\ a_n x_n^n + a_{n-1} x_n^{n-1} + \dots + a_1 x_n + a_0 = y_n \end{cases} \quad (4.3)$$

Накладення умов (4.2) на поліном (4.1) дозволяє однозначно визначити його коефіцієнти. Дійсно, вимагаючи для  $\varphi(x)$  виконання умов (4.2), одержуємо систему  $n+1$  рівнянь із  $n+1$  невідомими:

$$\sum_{k=0}^n a_k x_i^k = y_i, \quad \text{при } (i = 0, 1, \dots, n)$$

Вирішуючи цю систему щодо невідомих  $a_0, a_1, \dots, a_n$  одержимо аналітичне вираження поліному (4.1). Система (4.3) завжди має єдине рішення, тому що її визначник

$$D = \begin{vmatrix} x_0^n & x_0^{n-1} & \dots & x_0 & 1 \\ x_1^n & x_1^{n-1} & \dots & x_1 & 1 \\ \dots & \dots & \dots & \dots & \dots \\ x_n^n & x_n^{n-1} & \dots & x_n & 1 \end{vmatrix}$$

відомий в алгебрі як **визначник Вандермонда**, відмінний від нуля. Звідси випливає, що інтерполяційний поліном  $\varphi(x)$  для функції  $f(x)$ , заданої таблично, існує і він єдиний.

### Інтерполяційна формула Лагранжа

Інтерполяційна формула Лагранжа має вид

$$L_n(x) = \sum_{i=0}^n y_i \frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)} \quad (4.4)$$

Для побудови інтерполяційної формули Лагранжа в MathCAD зручно використовувати функцію *if*.

Повертає значення *tval*, якщо *cond* відмінний від 0 (істина). Повертає значення *fval*, якщо *cond* дорівнює 0 (неправда).

Часто інтерполяція ведеться для функцій, що задані у виді таблиці, із рівновіддаленими значеннями аргументу *h* кроком

$$h_i = x_{i+1} - x_i = \text{const.}$$

Введемо попередньо поняття **кінцевих різниць**:

$$\begin{aligned}\Delta y_i &= y_{i+1} - y_i, (i = 0, 1, \dots, n-1) \\ \Delta^2 y_i &= \Delta y_{i+1} - \Delta y_i, (i = 0, 1, \dots, n-2) \\ \Delta^k y_i &= \Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i, (i = 0, 1, \dots, n-k)\end{aligned}\quad (4.5)$$

Введемо також змінну  $t = (x - x_0)/h$ .

З урахуванням уведених позначень **перша інтерполяційна формула Ньютона** має вид:

$$\begin{aligned}N_{n1}(x) = N_{n1}(x_0 + th) &= y_0 + t\Delta y_0 + \frac{t(t-1)}{2!}\Delta^2 y_0 + \dots \\ &\dots + \frac{t(t-1)\dots(t-n+1)}{n!}\Delta^n y_0.\end{aligned}\quad (4.6)$$

**Друга інтерполяційна формула Ньютона** для інтерполяції назад має вид:

$$\begin{aligned}t &= \frac{x - x_n}{h}, \\ N_{n2}(x) = N_{n2}(x_n + th) &= y_n + t\Delta y_{n-1} + \frac{t(t+1)}{2!}\Delta^2 y_{n-2} + \dots \\ &\dots + \frac{t(t+1)\dots(t+n-1)}{n!}\Delta^n y_0.\end{aligned}\quad (4.7)$$

Проте, інтерполяція при великому числі вузлів призводить до необхідності працювати з багаточленами вищих ступенів (наприклад, 50-го або навіть 100-го), що неприйнятно як із погляду обчислень, так і через схильність таких багаточленів до осциляції (коливанням) між вузлами сітки. Тому на практиці часто використовують **локальну інтерполяцію**.

### Локальна інтерполяція

Найпростішим і часто використовуваним видом локальної інтерполяції є **лінійна інтерполяція**. Вона складається в тому, що задані точки  $M(x_i, y_i)$  ( $i = 0, 1, \dots, n$ ) з'єднуються прямолінійними відрізками, і функція  $f(x)$  наближається до ламаної з вершинами в даних точках (рис. 4.2).

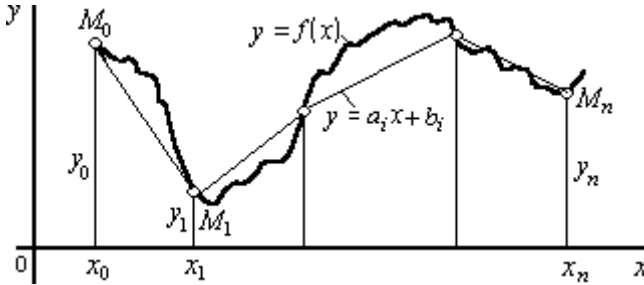


Рисунок 4.2 – Лінійна інтерполяція

Рівняння кожного відрізка ламаної лінії в загальному випадку різні. Оскільки є  $n$  інтервалів  $(x_i, x_{i+1})$ , то для кожного з них у якості рівняння інтерполяційного полінома використовується рівняння прямої, що проходить через дві точки. Зокрема, для  $i$ -го інтервалу можна написати рівняння прямої, що проходить через точки  $(x_i, y_i)$  і  $(x_{i+1}, y_{i+1})$ , у виді:

$$\frac{y - y_i}{y_{i+1} - y_i} = \frac{x - x_i}{x_{i+1} - x_i}. \quad (4.8)$$

Звідси

$$y = a_i x + b_i, x_i \leq x \leq x_{i+1} \quad (4.9)$$

$$a_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i}, b_i = y_i - a_i x_i.$$

Отже, при використанні лінійної інтерполяції спочатку потрібно визначити інтервал, у який потрапляє значення аргументу  $x$ , а потім підставити його у формулу (4.9) і знайти наближене значення функції у цій точці.

У випадку **квадратичної інтерполяції** в якості інтерполяційної функції на відрізку  $(x_{i-1}, x_{i+1})$  приймається квадратний тричлен.

Рівняння квадратного тричлена

$$y = a_i x^2 + b_i x + c_i, x_{i-1} \leq x \leq x_{i+1}, \quad (4.10)$$

містять три невідомих коефіцієнти  $a_i, b_i, c_i$  для визначення котрих необхідні три рівняння.

Ними служать умови проходження параболи (4.10) через три точки  $(x_{i-1}, y_{i-1})$ ,  $(x_i, y_i)$ ,  $(x_{i+1}, y_{i+1})$ . Ці умови можна записати у виді:

$$\begin{aligned}
 a_i x_{i-1}^2 + b_i x_{i-1} + c_i &= y_{i-1}, \\
 a_i x_i^2 + b_i x_i + c_i &= y_i, \\
 a_i x_{i+1}^2 + b_i x_{i+1} + c_i &= y_{i+1}.
 \end{aligned}
 \tag{4.11}$$

Інтерполяція для будь-якої точки  $x \in [x_0, x_n]$  проводиться по трьох найближчих точках.

### Кубічна сплайн-інтерполяція

У останні роки інтенсивно розвивається новий розділ сучасної обчислювальної математика – **теорія сплайнів**. Сплайни дозволяють ефективно вирішувати задачу опрацювання експериментальних залежностей між параметрами, що мають достатньо складну структуру.

Розглянуті вище методи локальної інтерполяції, по суті, є найпростішим сплайном першого ступеня (для лінійної інтерполяції) і другого ступеня (для квадратичної інтерполяції).

У середовищі Mathcad є для цього інструментарій: засоби **лінійної інтерполяції** (функція *linterp*) і **інтерполяції сплайном** (функція *interp*) - лінійним (*lspline*), параболічним (*pspline*) і кубічним (*cspline*).

**linterp(vx, vy, x)**

**lspline(vx, vy)**

**pspline(vx, vy)**

**cspline(vx, vy)**

**interp(vs, vx, vy, x)**

Використовує вектори даних  $vx$  і  $vy$ , щоб повернути лінійно інтерпольоване значення  $y$ , що відповідає третьому аргументу  $x$ .

Всі ці функції повертають вектор коефіцієнтів других похідних, що ми будемо називати  $vs$ . Вектор  $vs$ , використовується у функції *interp*.

Повертає інтерпольоване значення  $y$ , що відповідає аргументу  $x$ .

### Апроксимація

Вивчаючи теорію інтерполяції, ви познайомилися з інтерполяційними формулами, що у точності відтворюють значення даної функції у вузлах інтерполяції. Проте в ряді випадків виконання цієї умови утруднено або навіть недоцільно:

- якщо задані величини  $x$  і  $y$  є експериментальними даними, то можуть містити в собі суттєві помилки, тому що отримані в результаті вимірів або спостережень. Тому побудова апроксимуючого багаточлена, що відтворює в точності задане значення функції, означало б ретельне копіювання допущених при вимірах помилок;

- якщо є точні значення функції в деяких точках, але число таких точок  $n$  дуже велике, то інтерполяційний багаточлен буде дуже високого ступеня (якщо тільки різниці не будуть ставати постійними).

Тому виникає задача побудови багаточлена деякого, цілком визначеного ступеня, але меншого, чим  $n-1$ , що хоча і не дає точних значень функції у вузлах інтерполяції, але достатньо близько до них підходить.

### Метод найменших квадратів

Найбільше поширеним методом апроксимації експериментальних даних є **метод найменших квадратів**. Метод дозволяє використовувати апроксимуючі функції довільного виду і відноситься до групи глобальних методів.

Запишемо суму квадратів відхилень для всіх точок  $x = x_0, x_1, \dots, x_n$ :

$$S = \sum_{i=0}^n \varepsilon_i^2 = \sum_{i=0}^n [\varphi(x_i, a_0, a_1, \dots, a_m) - y_i]^2. \quad (4.12)$$

Параметри  $a_0, a_1, \dots, a_m$  емпіричної формули  $y = \varphi(x, a_0, a_1, \dots, a_m)$  будемо знаходити з умови мінімуму функції  $S = S(a_0, a_1, \dots, a_m)$ . У цьому складається ідея методу найменших квадратів.

Важливою особливістю методу є те, що апроксимуюча функція може бути довільною. Її вид визначається особливостями розв'язуваної задачі, наприклад, фізичними розуміннями, якщо проводиться апроксимація результатів фізичного експерименту. Найбільше часто зустрічаються апроксимація прямої лінії (лінійна регресія), апроксимація поліномом (поліноміальна регресія), апроксимація лінійною комбінацією довільних функцій (узагальнена регресія).

Розглянемо застосування методу найменших квадратів для окремого випадку, широко використовуваного на практиці. У якості емпіричної формули розглянемо многочлен

$$\varphi(x) = a_m x^m + a_{m-1} x^{m-1} + \dots + a_1 x + a_0. \quad (4.13)$$

Для даних значень  $x = x_0, x_1, \dots, x_n$  и  $y = y_0, y_1, \dots, y_n$  необхідно підібрати багаточлен заданого ступеня  $m < n$  виду, який у заданих точках  $x_i$  приймає значення як можна більш близькі до табличних

значень  $y_i$ . Коефіцієнти  $a_i$  багаточлена (4.13) знаходять із рішення системи

$$\begin{cases} b_{00}a_0 + b_{01}a_1 + \dots + b_{0m}a_m = c_0, \\ b_{10}a_0 + b_{11}a_1 + \dots + b_{1m}a_m = c_1, \\ \dots \\ b_{m0}a_0 + b_{m1}a_1 + \dots + b_{mm}a_m = c_m. \end{cases} \quad (4.14)$$

де

$$b_{k,l} = \sum_{i=0}^n x_i^{k+l}, c_k = \sum_{i=0}^n x_i^k y_i,$$

$$k = 0, 1, \dots, m, l = 0, 1, \dots, m.$$

## 4.2 Порядок виконання лабораторної роботи

**4.2.1** Обчислити значення заданої за варіантом (табл. 4.2) функції  $y_i = f(x_i)$  у вузлах інтерполяції  $x_i = a + h \cdot i$ , з кроком  $h = (b - a)/10$ , при  $i = 0, 1, \dots, 10$ , на відрізку  $[a, b]$ .

**4.2.2** По обчисленим у пп. 4.2.1 експериментальним даним  $(x_i, y_i)$  провести параболічну інтерполяцію.

Для знаходження коефіцієнтів шуканого полінома (4.1) скласти систему лінійних алгебраїчних рівнянь (4.3), використовуючи визначник Вандермонда  $D_{i,j} = x_i^j$ . Знайти рішення системи рівнянь у з використанням функції *Isolve*.

Побудувати графік інтерполяційного багаточлена  $v(t) = \sum_{j=0}^m v_j t^j$

і відзначити на ньому вузлові точки  $(x_i, y_i)$ .

**4.2.3** Для обчисленої у пп. 4.2.1 табличної функції скласти формулу інтерполяційного багаточлена Лагранжа, використовуючи оператори підсумовування і перемножування по дискретному аргументу, а також функцію *if*.

Побудувати графік інтерполяційного багаточлена і відзначити на ньому вузлові точки  $(x_i, y_i)$ .

**4.2.4** Провести інтерполяцію заданої функції за допомогою першої та другої інтерполяційних формул Ньютона.

Побудувати графіки інтерполяційних багаточленів і відзначити на ньому вузлові точки  $(x_i, y_i)$ .

**Таблиця 4.2 - Варіанти індивідуальних завдань до п. 4.2.1**

№	$f(x)$	$[a, b]$	№	$f(x)$	$[a, b]$
1	$\sin x^2$	[0;2]	21	$2x+\cos(x^2)$	[0;2]
2	$\cos x^2$	[0;2]	22	$\sin(x^2)-x$	[0;2]
3	$e^{\sin x}$	[0;5]	23	$0.1x^4-\cos(3x)$	[0;2]
4	$1/(1+x^2)$	[0;3]	24	$\ln(x)+\cos(x)$	[2;4]
5	$e^{-(x+\sin^2 x)}$	[0;3]	25	$e^x/(e^x+1-x)$	[0;5]
6	$1/(1+e^{-2x})$	[0;3]	26	$2x-3\cos(4x)$	[0;1]
7	$\sin(x+e^{\sin x})$	[0;3]	27	$x^3-\cos(5x)$	[0;1]
8	$e^{-(x+2/x)}$	[1;3]	28	$\cos(3x)-x$	[0;2]
9	$x\cos(x+\ln(1+x))$	[1;5]	29	$e^x/(x^2+1)$	[0;5]
10	$10\ln 2x/(1+x)$	[1;5]	30	$2\sin(6x)$	[0;1]
11	$\sin(x^2)e^{-(x/2)}$	[0;3]	31	$3xe^{-x}-1$	[0;2]
12	$\cos(x+\cos^3 x)$	[0;3]	32	$10+66x-3x^3$	[0;4]
13	$\cos(x+e^{\cos x})$	[2;6]	33	$3\sin(x)+x-1$	[1;5]
14	$\cos(2x+x^2)$	[1;2]	34	$\cos(x^2)-0.5x$	[0;2]
15	$e^{\cos x}\cos x^2$	[0;2]	35	$x^4-x^2-2$	[0;2]
16	$e^{x-1}-x^3-x$	[0;1]	36	$x^5-x-1$	[1;2]
17	$x-1/(3+\sin(3.6x))$	[2;4]	37	$1-x-\operatorname{tg}(x)$	[1;2]
18	$3x+e^x-e^{-x}$	[1;4]	38	$e^x-e^{-x}-2$	[6;10]
19	$2x^2+5\cos(x^2)-2$	[1;3]	39	$\sin(x^2)-4\ln(x)-5$	[3;4]
20	$0.1x^2-x\sin(x)$	[1;3]	40	$0.25x^3+\cos(x)-2$	[0;4]

**4.2.5** Створити таблицю експериментальних даних за варіантом (табл. 4.3) у вузлах інтерполяції  $x_i = a + h \cdot i$  при  $i = 0, 1, \dots, 10$ , з кроком  $h=(b-a)/10$  на відрізку  $[a, b]$ .

**Таблиця 4.3 - Варіанти індивідуальних завдань до п. 4.2.5**

№	$y_i$	$[a, b]$
1	2.86; 2.21; 2.96; 3.27; 3.58; 3.76; 3.93; 3.67; 3.90; 3.64; 4.09	[0, 1]
2	1.14; 1.02; 1.64; 1.64; 1.96; 2.17; 2.64; 3.25; 3.47; 3.89; 3.36	[-1, 1]
3	4.70; 4.64; 4.57; 4.45; 4.40; 4.34; 4.27; 4.37; 4.42; 4.50; 4.62	[2, 4]
4	0.43; 0.99; 2.07; 2.54; 1.67; 1.29; 1.24; 0.66; 0.43; 0.35; 0.70	[2, 4]

Продовження таблиці 4.3

№	$y_i$	$[a, b]$
5	1.55; 1.97; 1.29; 0.94; 0.88; 0.09; 0.02; 0.84; 0.81; 0.09; 0.15	[1, 4]
6	3.24; 1.72; 1.95; 2.77; 2.47; 0.97; 1.75; 1.55; 0.12; 0.70; 1.19	[0, 4]
7	2.56; 1.92; 2.85; 2.94; 2.39; 2.16; 2.51; 2.10; 1.77; 2.28; 1.70	[-1, 2]
8	1.77; 0.92; 2.21; 1.50; 3.21; 3.46; 3.70; 4.02; 4.36; 4.82; 4.03	[-1, 3]
9	1.53; 0.45; 1.68; 0.12; 0.68; 2.36; 2.58; 2.53; 3.45; 2.70; 2.82	[4, 8]
10	2.50; 3.90; 3.54; 4.63; 3.87; 5.25; 4.83; 3.24; 3.08; 3.00; 4.70	[0, 5]
11	2.95; 3.38; 2.71; 2.37; 2.29; 2.75; 2.76; 2.74; 2.57; 2.40; 2.99	[1, 5]
12	-0.23; -0.03; -0.98; -0.97; -0.43; -0.91; -0.27; -0.19; 0.88; 1.06; 0.72	[2, 4]
13	2.36; 0.03; -0.38; -1.33; 0.25; -1.36; 0.95; 3.16; 4.03; 4.92; 4.20	[0, 2]
14	3.82; 4.07; 3.53; 4.83; 5.53; 5.04; 5.09; 5.87; 5.53; 4.72; 4.73	[3, 4]
15	2.35; 2.16; 2.39; 2.39; 2.18; 2.09; 2.44; 2.56; 3.35; 3.22; 2.65	[-3, 4]
16	2.46; 2.41; 2.36; 3.57; 3.67; 3.23; 3.78; 3.47; 3.20; 3.84; 4.19	[0, 1]
17	1.15; 1.12; 1.74; 1.65; 1.94; 2.16; 2.62; 3.21; 3.49; 3.92; 3.38	[-1, 1]
18	4.74; 4.62; 4.59; 4.47; 4.41; 4.36; 4.29; 4.35; 4.44; 4.52; 4.60	[2, 4]
19	0.41; 0.99; 2.08; 2.64; 1.77; 1.39; 1.34; 0.76; 0.53; 0.45; 0.71	[2, 4]
20	1.35; 1.96; 1.27; 0.93; 0.85; 0.11; 0.18; 0.82; 0.80; 0.09; 0.14	[1, 4]
21	3.14; 1.62; 1.75; 2.78; 2.57; 0.96; 1.76; 1.45; 0.13; 0.72; 1.29	[0, 4]
22	2.53; 1.91; 2.84; 2.93; 2.38; 2.17; 2.52; 2.13; 1.78; 2.25; 1.74	[-1, 2]
23	1.87; 0.95; 2.27; 1.54; 3.27; 3.49; 3.75; 4.06; 4.39; 4.88; 4.07	[-1, 3]
24	1.43; 0.45; 1.58; 0.16; 0.67; 2.37; 2.57; 2.52; 3.44; 2.72; 2.81	[4, 8]
25	2.55; 3.99; 3.55; 4.66; 3.88; 5.22; 4.88; 3.22; 3.00; 3.00; 4.77	[0, 5]
26	2.96; 3.39; 2.72; 2.36; 2.28; 2.76; 2.77; 2.75; 2.58; 2.41; 2.98	[1, 5]
27	-0.24; -0.05; -0.97; -0.96; -0.44; -0.90; -0.26; -0.15; 0.87; 1.05; 0.71	[2, 4]
28	2.35; 0.05; -0.35; -1.35; 0.25; -1.35; 0.95; 3.15; 4.05; 4.95; 4.25	[0, 2]
29	3.81; 4.06; 3.56; 4.86; 5.56; 5.03; 5.05; 5.85; 5.55; 4.77; 4.77	[3, 4]
30	2.31; 2.18; 2.34; 2.39; 2.14; 2.06; 2.41; 2.53; 3.37; 3.21; 2.63	[-3, 4]

**4.2.6** Апроксимувати багаточленами другого і шостого ступеня по методу найменших квадратів функцію (за пп. 4.2.5), що задана таблицею значень  $x_i$  і  $y_i$ , порівняти якість наближень. Побудувати графіки багаточленів і відзначити вузлові точки ( $x_i, y_i$ ).

### 4.3 Зміст звіту

- 4.3.1 Опис методів рішення рівнянь.
- 4.3.2 Розрахунки в системі MathCAD.
- 4.3.3 Висновки.

### 4.4 Контрольні запитання для самоперевірки

- 4.4.1 Що таке апроксимація функцій?
- 4.4.2 Для чого потрібна інтерполяція функцій?
- 4.4.3 Охарактеризуйте види інтерполяції.
- 4.4.4 Чим визначається близькість інтерполяційного полінома до заданої функції?
- 4.4.5 Чим визначається ступінь інтерполяційного полінома?
- 4.4.6 Які види глобальної інтерполяції вам відомі?
- 4.4.7 Які інтерполяційні формули застосовуються, якщо вузли інтерполяції рівновіддалені?
- 4.4.8 Що таке кінцеві різниці?
- 4.4.9 Що таке емпірична формула і як її підібрати?
- 4.4.10 Яку інтерполяційну формулу Ньютона необхідно застосовувати на початку рішення таблично заданої функції, і яку – наприкінці? Чому?
- 4.4.11 Як можна підвищити точність інтерполяції?
- 4.4.12 Які методи локальної інтерполяції вам відомі? Який із них найменш точний?
- 4.4.13 Який метод локальної інтерполяції проводиться по трьох точках?
- 4.4.14 У чому полягає метод найменших квадратів?

## 5 ЛАБОРАТОРНА РОБОТА № 5 ЧИСЕЛЬНЕ ІНТЕГРУВАННЯ

**Мета роботи:** засвоєння методів чисельного інтегрування та приблизне обчислення значення інтеграла різними способами.

### 5.1 Теоретичний матеріал

Задача чисельного інтегрування функції полягає в обчисленні визначеного інтеграла на основі ряду значень підінтегральної функції. Чисельне обчислення однократного інтеграла називається **механічною квадратурою**.

Ми будемо розглядати методи наближеного обчислення визначених інтегралів

$$J = \int_a^b f(x) dx, \quad (5.1)$$

засновані на заміні інтеграла кінцевою сумою

$$I_n = \sum_{k=0}^n C_k f(x_k), \quad (5.2)$$

де  $C_k$  – числові коефіцієнти, а  $x_k \in \square[a, b]$ , при  $k = 0, 1, \dots, n$ .

Наближена рівність

$$\int_a^b f(x) dx \approx \sum_{k=0}^n C_k f(x_k) \quad (5.3)$$

називається **квадратурною формулою**, а  $x_k$  – **вузлами** квадратурної формули. Похибка квадратурної формули визначається співвідношенням

$$\psi_n = \int_a^b f(x) dx - \sum_{k=0}^n C_k f(x_k). \quad (5.4)$$

У загальному випадку похибка квадратурної формули (5.4) залежить як від вибору коефіцієнтів  $C_k$ , так і від розташування вузлових  $x_k$ . Введемо на відрізок  $[a, b]$  рівномірну сітку з кроком  $h$ , тоді  $x_i = a + i \cdot h$ , де ( $i=0, 1, \dots, n$ ;  $h \cdot n = b - a$ ). Тепер вираження (5.1) можна представити у виді суми інтегралів по часткових відрізках:

$$J = \int_a^b f(x)dx = \sum_{i=1}^n \int_{x_{i-1}}^{x_i} f(x)dx. \quad (5.5)$$

Таким чином, для побудови формули чисельного інтегрування на відрізку  $[a, b]$  достатньо побудувати квадратурну формулу на частковому відрізку  $[x_{i-1}, x_i]$  і скористатися формулою (5.5).

### Постановка задачі

Потрібно обчислити визначений інтеграл

$$I = \int_a^b f(x)dx$$

за умовою, що  $a$  і  $b$  – кінцеві та  $f(x)$  виявляється безперервною функцією  $x$  на усьому інтервалі  $a \leq x \leq b$ . Загальний підхід до рішення задачі такий. Визначений інтеграл  $I$  являє собою площу, обмежену кривою  $f(x)$ , віссю  $x$  та ординатами у точках  $x = a$  і  $x = b$ .

Ми будемо обчислювати  $I$ , розбиваючи інтервал від  $a$  до  $b$  на кілька менших інтервалів, знаходити площу кожної «смуги», яка виходить при такому роздрібненні та підсумовувати площі цих смуг. Чим менше інтервал роздрібнення тим точніше буде обчислена сума. Проте, при цьому значно збільшиться кількість обчислень. Тому на практиці доводиться обмежуватись кінцевим роздрібненням інтервалу інтегрування функції, допускаючи при цьому деяку похибку.

Різноманітність методів чисельного інтегрування обумовлено стратегією вибору точок роздрібнення, яка забезпечує у кожному конкретному випадку мінімально можливу помилку.

### Метод середніх прямокутників

Метод середніх прямокутників заснований на апроксимації функції  $y = f(x)$  на кожному частковому інтервалі  $[x_i, x_{i+1}]$  багаточленом нульового ступеня, тобто константою, рівною за значенням функції  $y_i$  в центрі (середині) часткового інтервалу.

Геометричний смисл інтеграла – площа під кривою  $y = f(x)$ . Приблизно її можна обчислити як суму заштрихованих прямокутників при заміні кривої  $y = f(x)$  східчастою штриховою лінією, середина якої є серединою кожного часткового інтервалу:

$$I = h_x(y_1^c + y_2^c + \dots + y_n^c) = h_x(f(x_1^c) + f(x_2^c) + \dots + f(x_n^c)) \quad (5.6)$$

Графічно метод середніх прямокутників поданий на рис. 5.1.

Похибка формули (5.6) визначається вираженням

$$|\psi| \leq \frac{1}{24} h_i^3 f^{IV}(x_{i-1/2})$$

Таким чином, похибка формули (5.6) пропорційна  $O(h^3)$ .

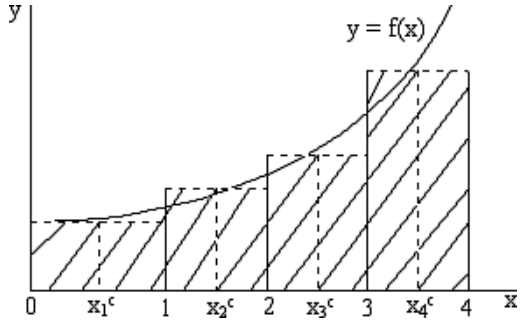


Рисунок 5.1 – Інтегрування методом середніх прямокутників

### Метод лівих прямокутників

Метод лівих прямокутників заснований на апроксимації функції  $y = f(x)$  на кожному частковому інтервалі  $[x_i, x_{i+1}]$  багаточленом нульового ступеня, тобто константою, рівною за значенням функції в лівій границі часткового інтервалу.

Геометричний смисл інтеграла – площа під кривою  $y = f(x)$ . Приблизно її можна обчислити як суму заштрихованих прямокутників при заміні кривої  $y = f(x)$  східчастою штриховою лінією, що починається ліворуч кожного часткового інтервалу:

$$I = h_x(y_0 + y_1 + \dots + y_{n-1}) = h_x(f(x_0) + f(x_1) + \dots + f(x_{n-1})). \quad (5.7)$$

Графічно метод лівих прямокутників поданий на рис. 5.2.

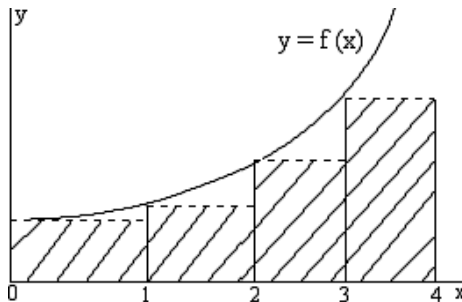


Рисунок 5.2 – Інтегрування методом лівих прямокутників

### Метод правих прямокутників

Метод правих прямокутників заснований на апроксимації функції  $y = f(x)$  на кожному частковому інтервалі  $[x_i, x_{i+1}]$  багаточленом нульового ступеня, тобто константою, рівною за значенням функції  $y_i$  у правій межі часткового інтервалу.

Геометричний смисл інтеграла – площа під кривою  $y = f(x)$ . Приблизно її можна обчислити як суму заштрихованих прямокутників при заміні кривої  $y = f(x)$  східчастою штриховою лінією, що починається справа кожного часткового інтервалу

$$I = h_x(y_1 + y_2 + \dots + y_n) = h_x(f(x_1) + f(x_2) + \dots + f(x_n)). \quad (5.8)$$

Через порушення симетрії у формулах (5.7) та (5.8) їхня похибка значно більше, ніж у методі середніх прямокутників.

Графічно метод правих прямокутників поданий на рис. 5.3.

### Метод трапецій

Метод трапецій заснований на апроксимації функції  $y = f(x)$  на кожному частковому інтервалі  $[x_i, x_{i+1}]$  інтерполяційним поліномом першого ступеня, тобто графічно апроксимуюча функція є кусочно-лінійною

$$y = P_1(x) = y_{i-1} + \frac{y_i - y_{i-1}}{x_i - x_{i-1}}(x - x_{i-1}). \quad (5.9)$$

Графічно метод трапецій поданий на рис. 5.4.

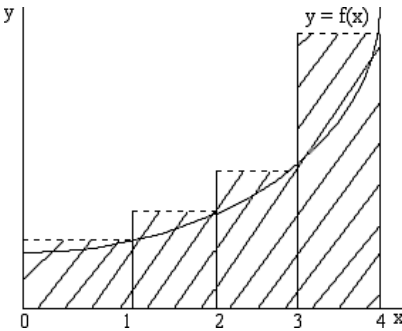


Рисунок 5.3 – Інтегрування методом правих прямокутників

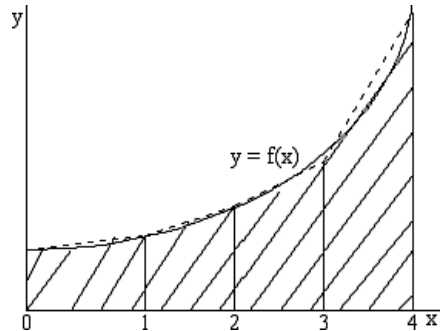


Рисунок 5.4 – Інтегрування методом трапецій

Геометричний смисл інтеграла – площа під кривою  $y = f(x)$ . Приблизно її можна обчислити як суму заштрихованих трапецій

$$h_x \left( \frac{y_{i-1} + y_i}{2} \right):$$

$$I = h_x \left( \frac{f(x_0) + f(x_n)}{2} + f(x_1) + f(x_2) + \dots + f(x_{n-1}) \right)$$

Похибка формули (5.9) визначається вираженням:

$$|\psi| \leq \frac{1}{12} h_i^3 f''(x_i). \quad (5.10)$$

Таким чином, похибка методу трапецій  $\Psi \sim O(h^3)$ , але вона в два рази більше, ніж для формули середніх прямокутників.

### Метод парабол (Сімпсона)

Метод парабол заснований на апроксимації функції  $y = f(x)$  на парі сусідніх часткових інтервалів  $[x_{i-1}, x_i]$ ,  $[x_i, x_{i+1}]$  інтерполяційним поліномом другого ступеня, тобто параболою

$$y = P_1(x) = a_0 + a_1x + a_2x^2.$$

Тоді виявляється, що частковий інтеграл

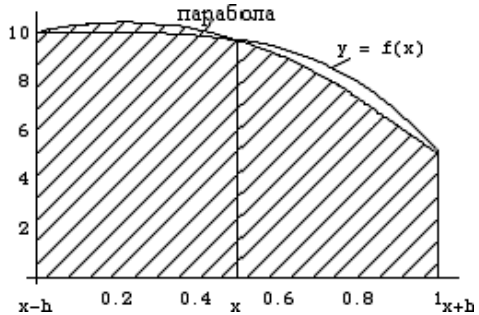


Рисунок 5.5 – Інтегрування методом Сімпсона

$$\int_{x_{i-1}}^{x_{i+1}} P_2(x) dx = \frac{h_x}{3} (y_{i-1} + 4y_i + y_{i+1}).$$

Обираючи число часткових інтервалів  $n$  парним, одержимо формулу для наближеного обчислення інтеграла методом парабол:

$$I = \frac{h_x}{3} (f(x_0) + 4(f(x_1) + f(x_3) + \dots + f(x_{n-1})) + 2(f(x_2) + f(x_4) + \dots + f(x_{n-2})) + f(x_n)) \quad (5.11)$$

Похибка формули (5.11) оцінюється таким вираженням:

$$|\psi| \leq \frac{1}{180} h_i^4 f^{IV}(x_i)$$

Таким чином, похибка формули Сімпсона пропорційна  $O(h^4)$ .

**Зауваження.** Слід зазначити, що у формулі Сімпсона відрізок інтегрування обов'язково розбивається на **парне** число інтервалів.

### Метод невизначених коефіцієнтів

Метод невизначених коефіцієнтів для чисельного інтегрування складається в обчисленні визначеного інтеграла (5.1) за допомогою формули

$$I = \int_a^b f(x) dx = \sum_{i=0}^n A_i y_i .$$

Коефіцієнти  $A_i$  знаходяться в результаті рішення системи рівнянь:

$$\begin{cases} I_0 = A_0 + A_1 + \dots + A_n \\ I_1 = A_0 x_0 + A_1 x_1 + \dots + A_n x_n \\ \dots \\ I_n = A_0 x_0^n + A_1 x_1^n + \dots + A_n x_n^n \end{cases}$$

де,  $I_k = \int_a^b x^k = \frac{b^{k+1} - a^{k+1}}{k+1}$  при  $k = 0, 1, \dots, n$ .

Систему рівнянь можна вирішити матрично з використанням функції *lsolve*.

### Обчислення інтегралів методами Монте-Карло

Методи, котрі розглянуті раніше називаються **детермінованими**, тобто позбавленими елемента випадковості.

**Методи Монте-Карло** (ММК) – це чисельні методи рішення математичних задач за допомогою моделювання випадкових величин. ММК дозволяють успішно вирішувати математичні задачі, обумовлені імовірними процесами. Більш того, при рішенні задач, не пов'язаних із будь-якими імовірностями, можна штучно придумати імовірну модель (і навіть не одну), що дозволяє вирішувати ці задачі. Розглянемо обчислення визначеного інтеграла (5.1).

При обчисленні цього інтеграла по формулі прямокутників інтервал  $[a, b]$  ми розбивали на  $N$  однакових інтервалів, у серединях яких обчислювалися значення підінтегральної функції. Вираховуючи

значення функції у випадкових вузлах, можна одержати більш точний результат

$$J = \int_a^b f(x)dx \approx \frac{b-a}{N} \sum_{i=1}^N f(x_i); \quad (5.12)$$

$$x_i = a + \gamma_i(b-a), \quad (5.13)$$

де  $\gamma_i$  – випадкове число, рівномірно розподілене на інтервалі  $[0, 1]$ .

Похибка обчислення інтеграла ММК  $\sim 1/\sqrt{N}$ , що значно більше, ніж у раніше вивчених детермінованих методів.

Проте при обчисленні кратних інтегралів детермінованими методами оцінка похибки переростає в задачу часом більш складну, чим обчислення інтеграла. У той же час похибка обчислення кратних інтегралів ММК слабо залежить від кратності і легко обчислюється в кожному конкретному випадку практично без додаткових витрат.

Для генерування послідовності випадкових чисел із нормальним законом розподілу в MathCAD можна використовувати функцію **rnd**.

**rnd(x)** Повертає рівномірно розподілене випадкове число між 0 и  $x$ .

Для реалізації методу Монте-Карло зручно використовувати функцію **mean**.

**mean(A)** Повертає середнє значення елементів масиву  $A$  розмірності  $m \times n$  відповідно до формули

$$mean(A) = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} A_{i,j}$$

## 5.2 Порядок виконання лабораторної роботи

**5.2.1** Визначити функцію  $f(x)$  як таблицю за варіантом (табл. 5.1), вирахувавши  $y_i = f(x_i)$  в точках  $x_i = a + hi$ , при  $i = 0, 1, \dots, 8$ , із кроком  $h=(b-a)/8$  на відрізьку  $[a, b]$ .

**5.2.2** Обчислити приблизно значення інтеграла (табл. 5.1)

$$I = \int_a^b f(x)dx$$

кожним із наступних методів: лівих прямокутників, правих прямокутників, середніх прямокутників, трапецій; парабол.

**5.2.3** Вирішити задачу пп. 5.2.2 методом невизначених коефіцієнтів для чисельного інтегрування.

**5.2.4** Вирішити задачу пп. 5.2.2 за допомогою вбудованого оператора системи MathCAD і порівняти результати.

**5.2.5** Обчислити інтеграл методом Монте-Карло.

Для цього необхідно:

- визначити діапазон випадкових чисел, наприклад  $j := 0..N$ ;
- визначити за допомогою функції *rnd* рівномірно розподілену випадкову величину  $h_j$  на відрізку інтегрування  $[a, b]$ ;
- створити вектор  $F_j = f(h_j)$ ;
- обчислити інтеграл.

**Таблиця 5.1 - Варіанти індивідуальних завдань**

№	$f(x)$	$[a;b]$	№	$f(x)$	$[a;b]$
1	$1/\sqrt{2x^2 + 1}$	[0.8; 1.6]	9	$\sqrt{2x} + \sqrt[3]{x}$	[0;8]
2	$\frac{\lg(x+2)}{x}$	[1.2;2]	10	$\sqrt{x+1}$	[3;8]
3	$\sin(2x)/x^2$	[0.8;1.2]	11	$\frac{\cos(x)}{x^2+1}$	[0.8;1.2]
4	$\frac{\operatorname{tg}(x^2)}{x^2+1}$	[0.2;1]	12	$\frac{\sqrt{x+1}}{\operatorname{ctg}(2x)}$	[0.1;0.7]
5	$\frac{\cos(x)}{x+1}$	[0.6;1.4]	13	$\frac{\sqrt{x}}{2} \sin\left(\frac{x}{2}\right)$	[1.2;2.8]
6	$1/\sqrt{0.5x^2 + 2}$	[0.4;1.2]	14	$\frac{x^2}{1+x^6}$	[1; $\sqrt{3}$ ]
7	$\frac{\sin(x^2 - 1)}{2\sqrt{x}}$	[1.3;2.1]	15	$2x + \frac{3}{\sqrt{x}}$	[1;4]
8	$\frac{\sin(x)}{x+1}$	[0.18;0.98]	16	$2x^2 - 4x$	[0;2]

Продовження таблиці 5.1

№	$f(x)$	[a;b]	№	$f(x)$	[a;b]
17	$1/\sqrt{x^2 + 2}$	[0.5;1.3]	29	$\frac{x+1}{\sqrt{x^2 + 1}}$	[0.4;1.4]
18	$\frac{\lg(x^2+0.5)}{2x^2+1}$	[0.4;0.8]	30	$1/x^2(x-1)$	[4;5]
19	$\frac{\lg(x^2+2)}{x+1}$	[1.4;2.2]	31	$\ln^2 x/x$	[1;e]
20	$1/\sqrt{2x^2 + 0.3}$	[0.8;1.7]	32	$x^2/\sqrt{x+1}$	[2.2;3.4]
21	$\frac{x}{2} \lg\left(\frac{x^2}{2}\right)$	[1.6;3.2]	33	$\frac{x^2 + 0.5}{\sqrt{x^2 + 1}}$	[0.5;1.6]
22	$x^3 \sqrt{4 + 5x^4}$	[0;1]	34	$\frac{\sqrt{x^2+2}}{x+2}$	[0.2;2.5]
23	$3(x-1)^2$	[1;2]	35	$\frac{2x-1}{x^3+x}$	[1;2]
24	$3\left(x^2 + x^2 e^{x^3}\right)$	[0;1]	36	$1/(x^2 + 1)$	[3/4;4/3]
25	$\sqrt{x}$	[1;4]	37	$x^3/\sqrt{x^4 + 4}$	[0;2]
26	$1/(x^4 - 1)$	[2;3]	38	$x/\sqrt{1 - x^2}$	[0;1/2]
27	$1/(x^3 + 1)$	[1;2]	39	$e^{1/2}/x^2$	[1;2]
28	$x/(x^2 + 3x + 2)$	[0;1]	40	$1/\sqrt{5 + 4x - x^2}$	[2;4]

### 5.3 Зміст звіту

5.3.1 Опис методів чисельного інтегрування.

5.3.2 Розрахунки в системі MathCAD.

5.3.3 Висновки.

## 5.4 Контрольні запитання для самоперевірки

5.4.1 Сформулюйте задачу чисельного інтегрування.

5.4.2 Методи середніх, лівих і правих прямокутників. Порівняйте їхню похибку, трудомісткість?

5.4.3 Задача чисельного інтегрування вирішена методом трапецій. Запропонуйте й обґрунтуйте шляхи підвищення точності (зменшення похибки) розрахунків.

5.4.4 Порівняйте метод трапецій і метод Симпсона.

5.4.5 Порівняйте методи Монте-Карло чисельного інтегрування з будь-яким детермінованим.

5.4.6 Необхідно обчислити інтеграл методами трапецій та Симпсона, розбивши область інтегрування на 27 інтервалів. Що можна сказати про точність і придатність цих методів?

## РЕКОМЕНДОВАНА ЛІТЕРАТУРА

- 1 Дзісь В.Г., Левчук О.В., Дячинська О.М. Прикладна математика на основі MathCAD: Навчальний посібник. Вінниця: ВНАУ, 2020. 378с.
- 2 Задачин В.М., Конюшенко І.Г. Чисельні методи: Навчальний посібник. Вид. ХНЕУ ім. С. Кузнеця, 2014. 180 с.
- 3 Копча-Горячкіна Г.Е. Чисельні методи в інформатиці. Навчально-методичний посібник. Частина 1. Ужгород: Видавництво Закарпатського державного університету. 2011. 76 с.
- 4 Лаазарев Ю.Ф. Моделювання на ЕОМ. Навчальний посібник. К.: Політехніка, 2007. 290с.
- 5 Ляшенко Б.М., Кривонос О.М., Вакалюк Т.А. Методи обчислень: навчально-методичний посібник. Житомир: видавництво ДЖУ, 2014. 228 с.
- 6 Ляшенко М.Я., Головань М.С. Чисельні методи: Підручник. К.:Либідь. 1996. 288с.
- 7 Шевчук О. Ф., Найко Д.А. Теорія ймовірностей та математична статистика: навч. посіб. Вінниця: ВНАУ, 2020. 382 с.
- 8 Жалдак М.І., Рамський Ю.С. Чисельні методи математики. - К.:Радянська школа, 1984. -206 с.
- 9 Бігун Я.Й. Числові методи. Інтерполювання. Числове інтегрування та диференціювання: Навчальний посібник / Чернівецький національний університет ім. Ю.Федьковича. – Чернівці: Рута, 2005. – 80с.
- 10 .Окуненко В.М., Ясинський В.К. Чисельні методи в моделюванні систем. – Чернівці: Золоті литаври, 2006. – 592с.
- 11 .Цегелик Г.Г. Чисельні методи. – Львів: Видавничий центр Львівського національного університету, 2004. – 408с.
- 12 Числові методи / П.П.Овчинников, В.М.Михайленко: За заг. ред. П.П.Овчинникова. – 3-тє видання, виправлене. – К.: Техніка, 2004. – 792с.
- 13 Ясинський В.К. Основи обчислювальних методів. Навч. посіб. – Чернівці: Золоті литаври, 2005. – 396с.
- 14 Херхагер М. MathCad 2000: полное руководство / М. Херхагер. Х. Партолль ; пер. с нем. ; под ред. К. Ю. Королькова. – Киев : ВНУ, 2000. – 416 с.

## Додаток А

### Убудовані оператори

У таблиці А.1, наведеній нижче, використовуються такі позначення:

- $X$  і  $Y$  – змінні або вираження будь-якого типу;
- $x$  і  $y$  – речовинне число;
- $z$  і  $w$  – речовинне або комплексне число;
- $m$  і  $n$  – ціле число;
- $A$  і  $B$  – масиви (вектори або матриці);
- $i$  – дискретний аргумент;
- $t$  – будь-яка змінна;
- $f$  – будь-яка функція.

**Таблиця А.1**

Оператор	Клавіши	Призначення оператора
$X := Y$	$X : Y$	локальне присвоювання $X$ значення $Y$
$X \equiv Y$	$X \sim Y$	глобальне присвоювання $X$ значення $Y$
$X =$	$X =$	виведення значення $X$
$X + Y$	$X + Y$	додавання $X$ і $Y$
$X +$ $+ Y$	$X$ [Ctrl][↵] $Y$	перенос великих формул, які не вміщуються на сторінці
$X - Y$	$X - Y$	вирахування з $X$ значення $Y$
$X \cdot Y$	$X * Y$	множення $X$ на $Y$
$\frac{X}{z}$	$X / z$	ділення $X$ на $z$
$z^w$	$z \wedge w$	зведення $z$ у ступінь $w$
$\sqrt{z}$	$z \backslash$	обчислення квадратного кореня з $z$
$\sqrt[n]{z}$	$n$ [Ctrl]\ $z$	обчислення кореня $n$ -го ступеня з $z$
$n!$	$n!$	обчислення факторіала
$B_n$	$B [ n$	введення нижнього індексу $n$
$A_{n,m}$	$A [ n , m$	введення подвійного індексу
$A^{<n>}$	$A$ [Ctrl]6 $n$	введення верхнього індексу
$\sum_{i=m}^n X$	[Ctrl][Shift]4	підсумовування $X$ по $i = m, m + 1, \dots, n$

Продовження таблиці А.1

Оператор	Клавіши	Призначення оператора
$\sum_i X$	\$	підсумовування $X$ по дискретному аргументі $i$
$\prod_{i=m}^n X$	[Ctrl][Shift]3	перемножування $X$ по $i = m, m + 1, \dots n$
$\prod_i X$	#	перемножування $X$ по дискретному аргументі $i$
$\sum_i X$	\$	підсумовування $X$ по дискретному аргументі $i$
$\int_a^b f(t)dt$	&	обчислення визначеного інтеграла $f(t)$ на інтервалі $[a, b]$
$\frac{d}{dt} f(t)$	?	обчислення похідної $f(t)$ по $t$
$\frac{d^n}{dt^n} f(t)$	[Ctrl]?	обчислення похідної $n$ -го порядку функції $f(t)$ по $t$
( )	,	введення пари круглих скобок із шаблоном
$x > y$	$x > y$	більше ніж
$x < y$	$x < y$	менше ніж
$x \geq y$	$x$ [Ctrl]0 $y$	більше або дорівнює
$x \leq y$	$x$ [Ctrl]9 $y$	менше або дорівнює
$z = w$	$z$ [Ctrl]= $w$	логічна рівність повертає 1, якщо операнди рівні, інакше 0
$z \neq w$	$z$ [Ctrl]3 $w$	не дорівнює
$ z $	$z$	обчислення модуля комплексного $z$

## Додаток Б

### Убудовані функції

#### Тригонометричні функції

<b>sin(z)</b> – синус	<b>csc(z)</b> – косеканс
<b>cos(z)</b> – косинус	<b>sec(z)</b> – секанс
<b>tan(z)</b> – тангенс	<b>cot(z)</b> – котангенс

#### Гіперболічні функції

<b>sinh(z)</b> – гіперболічний синус
<b>tanh(z)</b> – гіперболічний тангенс
<b>csch(z)</b> – гіперболічний косеканс
<b>cosh(z)</b> – гіперболічний косинус
<b>sech(z)</b> – гіперболічний секанс
<b>coth(z)</b> – гіперболічний котангенс

#### Обернені тригонометричні функції

<b>asin(z)</b> – обернений тригонометричний синус
<b>acos(z)</b> – обернений тригонометричний косинус
<b>atan(z)</b> – обернений тригонометричний тангенс

#### Показові і логарифмічні функції

<b>exp(z)</b> – показова функція (або $e^z$ )
<b>ln(z)</b> – натуральний логарифм (по основі $e$ )
<b>log(z)</b> – десятковий логарифм (по основі 10)

#### Функції роботи з частиною числа

<b>Re(z)</b>	- виділення дійсної частини $z$
<b>Im(z)</b>	- виділення мнімої частини $z$
<b>arg(z)</b>	- обчислення аргументу (фази)
<b>floor(x)</b>	- найбільше ціле, менше або рівне $x$
<b>ceil(x)</b>	- найменше ціле, більше або рівне $x$
<b>mod(x,y)</b>	- залишок від ділення $x/y$ із знаком $x$
<b>angle(x,y)</b>	- позитивний кут із віссю $x$ для точки з координатами $(x,y)$

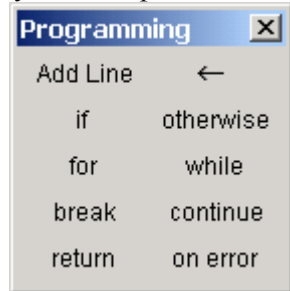
## Додаток В

### Оператори мови програмування MATHCAD

Для підвищення гнучкості у системі MathCAD передбачена можливість написання невеликих програм для рішення тих проблем, що не можуть бути реалізовані стандартними засобами. Звичайно прибігати до програмування доводиться в тих випадках, коли стандартні засоби або не можуть вирішити задачу, або неефективні.

Для написання програм використовується програмна палітра, що викликається кнопкою панелі керування. Як видно, усього є 10 операторів, із котрих і будується програма.

Оператори повинні вводитися **тільки з палітри**, писати їх «вручну» не рекомендується.



#### Опис операторів програмування

Add Line

- додати програмний ряд.



- оператор локального присвоювання.

У програмі не можна використовувати оператор присвоювання «:=», замість нього використовується оператор локального присвоювання, відмінність якого полягає в тому, що локальна змінна визначена тільки усередині свого блока і при виході з програми втрачає своє значення. Наприклад:

$$\left| \begin{array}{l} s \leftarrow 0 \\ i \leftarrow 1 \end{array} \right.$$

if

- умовний оператор. Створює конструкцію виду: **if** де перший операнд виконується, якщо справедлива умова, котра є другим операндом, наприклад:

$$x \leftarrow -1 \text{ if } x < 0$$

otherwise

- «інакше». З оператором **if** можливі більш складні конструкції при використанні ще одного оператора, що реалізує альтернативу.

Аналог традиційної конструкції **якщо ... то ... інакше ...**

$$\left| \begin{array}{l} x \leftarrow -1 \text{ if } x < 0 \\ x \leftarrow 1 \text{ otherwise} \end{array} \right.$$

**for** - оператор циклу. Забезпечує повторювані обчислення якщо відомо кількість кроків, наприклад:

Перший операнд – змінна циклу, це інтервальна змінна і її значення визначені в другому операнді. Третій операнд – тіло циклу, що може складатися з блока операторів і виконується поки не вичерпаються всі значення змінної циклу. Приклад:

$$\left| \begin{array}{l} s \leftarrow 0 \\ \text{for } i \in 0..100 \\ s \leftarrow s + i^2 \end{array} \right.$$

**break** - оператор, введений для підвищення гнучкості програмування і дозволяє закінчити цикл достроково, не вичерпавши всього списку.

**Приклад В.1.** Розглянемо задачу: необхідно знайти перше входження 0 у числовому масиві і повернути його індекс:

$$t(M) := \left| \begin{array}{l} \text{for } k \in 0..last(M) \\ \text{break if } M_k = 0 \\ k \end{array} \right.$$

Ми наводимо працюючу програму, де уведена функція *last(M)* яка повертає останній індекс масиву. Значенням програми, що повертається, є останній виконуваний оператор – *k*.

**continue** - оператор що дозволяє перервати виконання поточної ітерації і перейти до наступної.

$$\begin{array}{l} minmax(M) := \\ \left| \begin{array}{l} min \leftarrow M_0 \\ max \leftarrow M_0 \\ \text{for } k \in 1..last(M) \\ \left| \begin{array}{l} \text{if } M_k < min \\ \left| \begin{array}{l} min \leftarrow M_k \\ \text{continue} \end{array} \right. \\ max \leftarrow M_k \text{ if } M_k > max \end{array} \right. \\ \left( \begin{array}{l} min \\ max \end{array} \right) \end{array} \right. \end{array}$$

**Приклад В.2.**

Знаходження максимального і мінімального елементів масиву.

$$B := \begin{pmatrix} 1 \\ -5 \\ 4 \\ 8 \end{pmatrix} \quad minmax(B) = \begin{pmatrix} -5 \\ 8 \end{pmatrix}$$

Дійсно, якщо  $M_k < min$ , то нема рації перевіряти  $M_k > max$ , а краще перейти до наступного кроку циклу.

**while** - оператор циклу з передумовою. Використовується в тих випадках, коли заздалегідь невідома кількість кроків, необхідних для рішення задачі. Умова перевіряється перед початком кожного кроку циклу.

Реалізуємо наприклад алгоритм обчислення квадратного кореня  $a$  використовуючи ітераційну формулу

$$x_n = \frac{x_{n-1}}{2} + \frac{a}{2x_{n-1}}.$$

$$sqrt(a) := \begin{array}{|l} x \leftarrow \frac{a}{2} \\ \varepsilon \leftarrow 1 \\ \text{while } \varepsilon > TOL \\ \quad \left| \begin{array}{l} z \leftarrow \frac{x}{2} + \frac{a}{2 \cdot x} \\ \varepsilon \leftarrow |x - z| \\ x \leftarrow z \end{array} \right. \\ x \end{array}$$

$$sqrt(2) = 1.414$$

$$sqrt(10000) = 100$$

**return** - оператор служить для припинення роботи програми і повернення результату: *return* ■

Припустимо нам необхідно обчислити позицію першого входження числа в масив:

$$num(x, M) := \text{for } i \in 0..last(M)$$

$$\text{return } i \text{ if } x = M_i$$

$$num(4, B) = 2$$

**on error** - оператор служить для обробки помилкових ситуацій

■ *on error* ■

Наприклад, нам необхідно описати функцію  $f(x) := \frac{1}{x}$ , щоб не було особливості в 0.

$$f(x) := 0 \text{ on error } \frac{1}{x} \quad f(0) = 0 \quad f(0.01) = 100$$

**Примітка.** Оператор *on error* може використовуватися в арифметичних вираженнях.

**Приклад В.3.** Робота з масивами даних (матрицями).

Розглянемо програму упорядкування чисел по убаванню в одномірному масиві. Нехай даний масив чисел:

Ми скористаємося вкладеними циклами й у якості тіла циклу по *i* використовуємо ще один цикл по *j*

$$\begin{array}{l}
 \text{msort}(W) := \left| \begin{array}{l}
 k \leftarrow \text{last}(W) \\
 \text{for } i \in 0..k-1 \\
 \quad \text{for } j \in k-1, k-2..i \\
 \quad \quad \text{if } W_j < W_{j+1} \\
 \quad \quad \quad \left| \begin{array}{l}
 q \leftarrow W_j \\
 W_j \leftarrow W_{j+1} \\
 W_{j+1} \leftarrow q
 \end{array} \right. \\
 \quad \quad \quad W
 \end{array} \right. \\
 \\
 Z := \begin{pmatrix} 1 \\ 6 \\ -2 \\ 0 \\ -1 \end{pmatrix} \qquad \text{msort}(Z) = \begin{pmatrix} 6 \\ 1 \\ 0 \\ -1 \\ -2 \end{pmatrix}
 \end{array}$$

Тут реалізований найпростіший алгоритм сортування, коли більше число як би впливає наверх при кожному кроку циклу по *i*, у той час як у циклі по *j* на кожному кроку відбувається порівняння пари чисел і заміна, якщо більше число знаходиться нижче, причому ця заміна здійснюється знизу. Відзначимо, що в системі є стандартна функція сортування *sort()*.

**Примітка.** Другий цикл ми організували з негативним кроком від кінцевого значення до початкового.