МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
НТУУ «КПІ»
КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

МЕТОДИЧНІ ВКАЗІВКИ І ЗАВДАННЯ ДО ЛАБОРАТОРНИХ РОБІТ ЗА КУРСОМ «АЛГОРИТМИ ТА МЕТОДИ ОБЧИСЛЕНЬ»

(для студентів спеціальності "Комп'ютерна інженерія")

Розглянуто
на засіданні кафедри ВТ
протокол №__ від ____12 р.

Затверджено
на засіданні навчально-
видавничої ради НТУУ «КПІ»
протокол № від 2012 р.

- 2012-

УДК 681.3.06

Методичні вказівки до виконання лабораторних робіт з дисципліни «Комп’ютерна інженерія». /Укл.: Новотарський М.А.: НТУУ «КПІ»- 2012, 70 с.

Методичні вказівки до виконання лабораторних робіт з дисципліни «Алгоритми та методи обчислень» містять завдання до лабораторних робіт. До кожної роботи наведено методичні рекомендації з теорії, яка використовується при виконанні лабораторних робіт, наведені деякі алгоритми вирішення завдань, вимоги до виконання роботи, індивідуальні завдання та контрольні питання.

Укладач: Новотарський М.А.

Рецензент: Стиренко С.Г.

Лабораторна робота №1

Тема: «Поняття алгоритму. Задавання алгоритмів у вигляді блок-схем».
Мета: Навчитися створювати блок-схеми лінійного алгоритму; розгалуженого алгоритму та циклічного алгоритму за допомогою редактора блок-схем **afce** або іншого довільного редактора.
Завдання: Відповідно до варіанту завдання розробити блок-схеми обчислення виразів для лінійного алгоритму, алгоритму, що розгалужується та циклічного алгоритму. У відповідності до блок-схеми створити програму обчислення виразу на алгоритмічній мові Pascal.

Теоретичні основи:

Алгоритм - чіткий опис послідовності дій, які необхідно виконати при розв'язуванні задачі. Можна сказати, що алгоритм описує процес перетворення вхідних даних у результати, тому для розв'язування будь-якої задачі необхідно:

1. Ввести вхідні дані.
2. Перетворити вхідні дані в результати (вихідні дані).
3. Вивести результати.

Розробка алгоритму розв'язання задачі - це розбиття задачі на послідовно виконувані етапи, причому результати виконання попередніх етапів можуть використовуватися при виконанні подальших. При цьому мають бути чітко вказані як зміст кожного етапу, так і порядок виконання етапів. Окремий етап алгоритму являє собою або іншу, більш просту задачу, алгоритм розв'язання якої відомий (розроблений заздалегідь), або повинен бути достатньо простим і зрозумілим без пояснень.

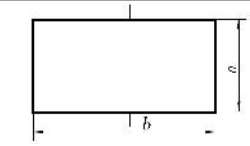
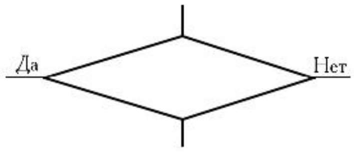
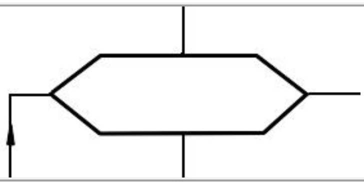
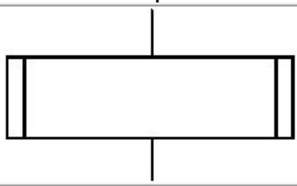
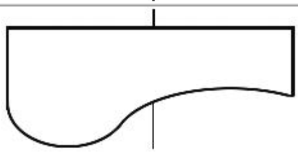
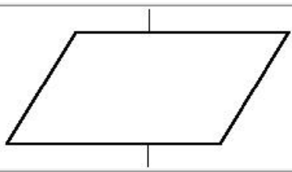
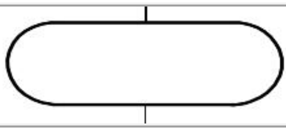
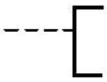
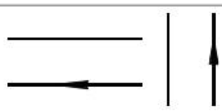
Зображення алгоритму у вигляді блок-схеми

Блок-схемою називається наочне графічне зображення алгоритму, коли окремі його етапи зображуються за допомогою різних геометричних фігур - блоків, а зв'язки між етапами (послідовність виконання етапів) вказуються за допомогою стрілок, що з'єднують ці фігури. Конфігурація і розміри блоків, а також порядок графічного оформлення блок-схем регламентовані ГОСТ 19002-80 і ГОСТ 19003-80 "Схеми алгоритмів і програм". Блоки супроводжуються написами.

У табл. 1 наведені найбільш часто використовувані блоки, зображені елементи зв'язків між ними і дано коротке пояснення до них. Блоки і елементи зв'язків називають елементами блок-схем.

Представлених в таблиці елементів цілком достатньо для зображення алгоритмів, які необхідні при виконанні студентських робіт. При з'єднанні блоків слід використовувати тільки вертикальні і горизонтальні лінії потоків. Горизонтальні потоки, що мають напрямок справа наліво, і вертикальні потоки, що мають напрямок знизу вгору, повинні бути обов'язково позначені стрілками. Інші потоки можуть бути помічені або залишені не поміченими. Лінії потоків повинні бути паралельні лініям зовнішньої рамки або границь аркуша.

Таблиця 1

Назва	Елемент	Коментар
Процес		Обчислювальна дія або послідовність обчислювальних дій
Рішення		Перевірка умови
Модифікація		Заголовок циклу
Зумовлений процес		Звернення до процедури
Документ		Виведення даних, друк даних
Введення / Виведення		Введення / Виведення даних
З’єднувач		Розрив лінії потоку
Початок, Кінець		Початок, кінець, пуск, зупинка, вхід і вихід в допоміжних алгоритмах
Коментар		Використовується для розміщення написів
Горизонтальні і вертикальні потоки		Лінії зв’язків між блоками, напрям потоків
Злиття		Злиття ліній потоків
Міжсторінковий з’єднувач		Немає

Відстань між паралельними лініями потоків повинна бути не менше 3 мм, між іншими елементами схеми - не менше 5 мм. Горизонтальний і вертикальний розміри блоку повинні бути кратними 5 мм (ділитися на 5 без остачі).

Співвідношення горизонтального і вертикального розмірів блоку b / a = 1.5 є основним. При ручному виконанні блоку допустимо відношення b / a = 2. Блоки "Початок", "Кінець" і "З’єднувач" мають висоту a / 2, тобто вдвічі менше основної висоти блоків. Для розміщення блоків рекомендується поле аркуша розбивати на горизонтальні і вертикальні (для розгалужених схем) зони. Для зручності опису блок-схеми кожен її блок слід пронумерувати. Зручно використовувати наскрізну нумерації блоків. Номер блоку розташовують у розриві в лівій верхній частині рамки блоку.

В залежності від структури алгоритми бувають: лінійні, що розгалужуються та циклічні.

Лінійні алгоритми зазвичай представляють найпростішими блок-схемами, що описують відносно прості обчислення або процеси. Етапи виконання окремих етапів в таких алгоритмах є незмінними і виконуються послідовно один за одним.(Рис. 1.)

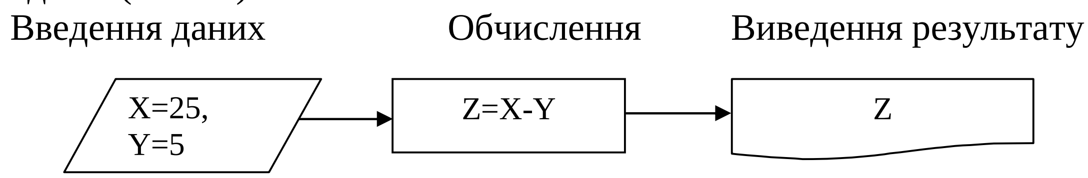


Рис 1. Приклад лінійного алгоритму

Алгоритми, що розгалужуються представляють блок-схемами, що дозволяють обирати виконання процесу у тому чи іншому напрямку в залежності від умов, заданих заздалегідь, або визначаються в процесі виконання програми.

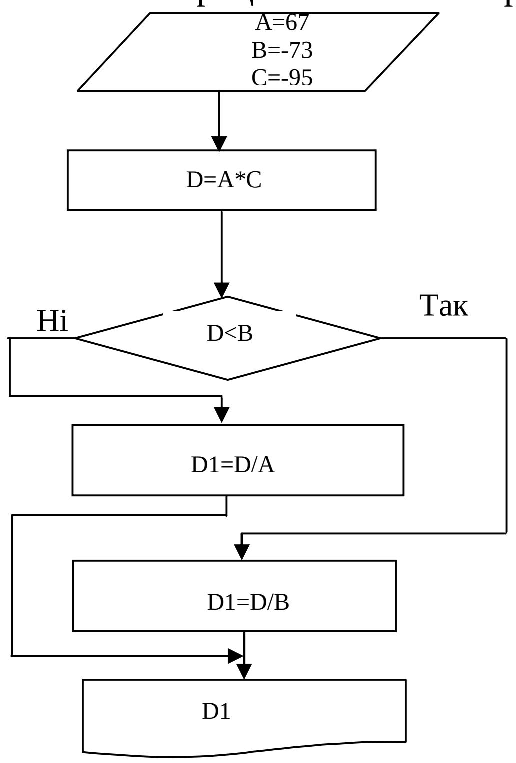
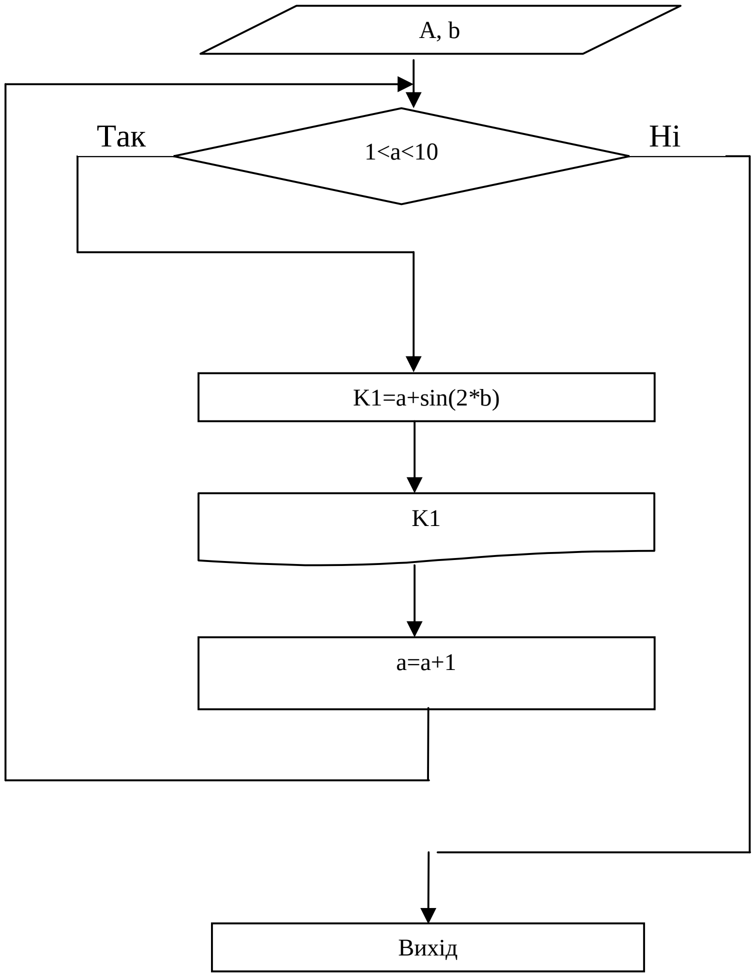


Рис. 2. Схема алгоритму, що розгалужується.

Циклічн ...бирати як напрямок виконання процесу, так і в залежності від заданих умов циклічно (повторно) виконувати певні процеси або операції. Більшість алгоритмів є циклічними.

На рис. 3. показано приклад обчислення значення виразу K1=a+sin2b, якщо ***a*** змінюється в діапазоні від 1 до 10 з кроком 1.



Відповідно до номеру у списку групи вибрати варіант завдання для виконання лабораторної роботи. Варіанти завдань зведені в таблицю 2

<u>Таблия 2</u>

	Лінійний	Що розгалужується	Циклічний
1	$Y1=(a+c)^2+(a+c)^3$	Знайти корені рівняння $2*x^2+4*x-48=0$	Обчислити $f=\sum_{i=0}^{10}\left(a_i^{\ 2}+56*c_i*fg_i\right)$
2	Y1=(a-z)+(a-z)/6	Знайти значення виразу $y=2a^2+x$ для a>10, та $y=2a^2-x$ у іншому випадку	Обчислити $F=\prod_{i=1}^{5}(a_i+b_i)$
3	$Y1=(a*c)^2+(a*c)^3$	Якщо b/z>d то f=sin(wf)	Обчислити H=10!*a23
4	$Y1=(a/c)^2+(a/c)^3$	Якщо f=0, то h=lg(lk)+d*sin(wer)	Обчислити та вивести на друк значення кожного

	Лінійний	Що розгалужується	Циклічний
			$K=a^2+2*a*b*c^3$ *a* - змінюється в діапазоні від –4 до 18 з кроком 1
5	$Y1=(a+x)/5+(a+c)^3$	Якщо kc>p $y=\sin^2(a)$ kc<p y= sin(a)	Розробити алгоритм заповнення у базу даних віку студентів Вашої групи.
6	$Y1=(2*a)^C+(2*c)^H$	Обчислити w=(rt*4-24*x)/(25*x-rt*cf), передбачити запобігання діленню на нуль.	Обчислити $Y=\sum_{a=2}^{10}\left(a^2+a^3\right)$
7	$k1=(a*c)/7+(a*c)^3$	Обчислити $y=k*x^2$ для x=[3-7]	Обчислити $Y=\prod_{a=1}^{10}\left(a^4+a\right)$
8	$Y1=(t1/5)^2+(t1/5)^3$	Обчислити $y=k*x^2$ для x=[3-7]	Обчислити Y=r+kx/6!
9	$z1=(5+f1*c)^2+(a+f1*c)^3$	Обчислити $y=k*x^2$ для 3<x<7	Обчислити середнє арифметичне десяти значень змінної *X*.
10	Z2=r1+21+(r1+21)/j	Обчислити $y=k*x^2$ для 3>x>7	Обчислити середнє геометричне десяти значень змінної *X*.
11	A1=sin(a/6)+2*sin(a/6)	Обчислити $y=k*x^2*\lg(f*g)$, передбачити запобігання від’ємного значення під знаком логарифма.	Обчислити $Y=\prod_{a=1}^{40}\left(a^4+a\right)$ *a* змінюється з кроком 4
12	A1=2*cos1/2-sin1/2	Обчислити $k=\sqrt{\frac{d=b-kj}{23*gf+6*vc}}$, передбачити: - неможливість обчислення значення у разі від’ємного значення підкореневого виразу; - запобігання ділення на нуль.	Обчислити $f=\sum_{i=0}^{50}\left(a_i{}^2+56*c_i*fg_i\right)$ *i* змінюється з кроком 5
13	$A3=(a/b)^2+(a/b)^3$	Обчислити $g1=fd*kx^{kx}$ для додатних значень показника степені.	Вивести на друк всі значення $y_i=a_i+b_i$ *I* змінюється від 5 до 10
14	$D7=(a+bx)^2*(a+bx)^5$	Обчислити $g1=fd*kx^{kx}$ для від’ємних значень показника степені.	Знайти суму квадратів a^2+b^2для десяти значень *a*, *b*.

	Лінійний	Що розгалужується	Циклічний
15	$D6=(a+b/x)^2+(a+b/x)^8$	Обчислити та вивести на друк тільки додатні значення виразу df=m*5-kl/7	Знайти квадрат суми для десяти значень *a* і *b*
16	S1=(h/s+8)*(h/s+8)/k	Обчислити та вивести на друк тільки від’ємні значення виразу $z43=d*m^5-kl/7*dg$	Обчислити $f=\sum_{a=0}^{50} a^2 + 56 * c * fg$ *a* змінюється з кроком 0,5
17	S2=(h/s-5)/2+(h/s+8)*k	Обчислити вираз $a=\frac{g^{gh}}{nb^{kj}}$ для додатних значень знаменника.	Обчислити $Y=\prod_{a=1}^{5}\left(a^4 + a\right)$ *a* змінюється з кроком 0,25
18	$M3=\sin(a+2)-(\sin(a+2))^3$	Обчислити вираз $a=\frac{g^{gh}}{nb^{kj}}$ для від’ємних значень знаменника.	Обчислити різницю квадратів суми для 10 значень *a* і *b*.
19	$M6=\lg(s/u)+(s/u)^7$	Обчислити вираз $a=\frac{g^{gh}}{nb^{kj}}$ для додатних значень чисельника.	Обчислити середнє арифметичне для 25 значень змінної *a*.
20	C4=lg(a+b)-ln(a-b)	Обчислити вираз $a=\frac{g^{gh}}{nb^{kj}}$ для додатних значень показника степеня знаменника.	Обчислити середнє квадратичне для 100 значень змінної *df*
21	C4=lg(h-f)/ln(h-f)	Обчислити вираз $a=\frac{g^{gh}}{nb^{kj}}$ для додатних значень знаменника.	Знайти значення виразу ty=5!+12!
22	C4=sin(f/h)/cos(f/h)	Обчислити та вивести на друк значення виразу lk=(x+24*x)/rt*dg, для від’ємних значень виразу у дужках.	Знайти значення виразу ty=5!/9!
23	F1=sin(n/k)+ln(n/k)	Обчислити площу кругу, передбачити неможливість обчислення у разі від’ємного значення діаметра.	Обчислити $Y=\prod_{a=1}^{10} a^4 + 5!$ *a* змінюється з кроком 1

	Лінійний	Що розгалужується	Циклічний
24	F1=cos(n*f/y)-lg(n*f/y)	Обчислити об’єм циліндру. Передбачити неможливість обчислення у разі від’ємного значення діаметру.	Обчислити f=$\sum_{i=0}^{50} a_i{}^2 + 6!$
25	F1=$(d+r/g)^F/(d+r/g)^D$	Обчислити значення er=ctg(gh)	Обчислити J=$\frac{1}{n_1}+\frac{1}{n_2}+...+\frac{1}{n_m}$ m=25
26	B1=$(z+2)^{g+h}/(z+2)^{(g+h)/3}$	Обчислити s=$\sqrt{24 * gh - sd * b - yt\ /\ vb}$	Обчислити J=$\frac{1}{n_1} * \frac{1}{n_2} * ... * \frac{1}{n_m}$ *m*=15
27	B2=$s^{c+sd}+fg^{c+sd}$	Обчислити cv=27*tg(z/3)	Обчислити Kl=$\frac{n_1 * n_2 * ...n_m}{n_1 + n_2 + ... + n_m}$ m=45
28	B2=$(c+2fg)/7-(g+k)^{(c+2fg)}$	Обчислити швидкість автомобіля для першої половини шляху і прискорення – для другої.	Обчислити ut=$\frac{\sum_{a=1}^{5} a^3}{\prod_{b=1}^{5} b^6}$
29	B52=$(c+27gd)^{(c+27gd)}$	Вивести на друк число 1, якщо кількість студентів у групі не перевищує 25, а у протилежному випадку – дійсне число студентів.	Вивести на друк числа від 1 до 1000 з кроком, що дорівнює сумі попереднього та попереднього помноженого на два. Перший крок дорівнює одиниці.
30	F15=$\sin^2(a+d)-\sin^3(a+d)$	Вивести на друк повідомлення “truth”, якщо значення виразу re+21*j+5/kj =45 є вірним, та ”false” – у протилежному.	Розробити алгоритм заповнення у базу даних прізвищ студентів Вашої групи.

Вимоги до програмного забезпечення:

1. Модульна структура програми;
2. Уведення даних із клавіатури і з зовнішнього файлу;
3. Перевірка коректності введених даних;

4. Меню.

Зміст звіту:

1. Титульний лист;
2. Тема завдання;
3. Завдання;
4. Блок-схеми алгоритмів;
5. Роздруківки блок-схем;
6. Роздруківка тексту програми;
7. Роздруківка результатів виконання програми;
8. Аналіз результатів.

Контрольні питання

1. Дати визначення алгоритму.
2. З яких основних етапів складається алгоритм розв'язування задачі за допомогою комп'ютера?
3. Які основні елементи блок-схем ви знаєте?
4. Перерахуйте загальні властивості алгоритмів.
5. Які існують способи задавання алгоритмів.

Лабораторна робота №2

Тема: «Машина Тьюринга».

Мета: Метою даного заняття є закріплення знань з побудови та роботи машин Тьюринга, які є математичними (формальними) моделями алгоритмів.

Завдання: Відповідно до варіанту написати програму для машини Тьюринга, наприклад Algo2000.exe, або створеної самостійно моделі машини Тьюринга, яка здатна виконувати операції, що необхідні для виконання завдання.

Теоретичні основи:

Машина Тьюринга - це пристрій, що містить друкарську стрічку нескінченної довжини, розбиту на комірки. У кожну комірку може бути записаний один і тільки один символ з вхідного алфавіту $A = \left\{a_0, a_1, ..., a_m\right\}$ машини Тьюринга.

В рамках цього прикладу для простоти будемо розглядати алфавіт $A = \left\{a_0, a_1, a_2\right\}$, де $a_1 = 0$, $a_1 = 1$, а $a_0 = \lambda$ розглядається як порожній "символ" (саме так слід розуміти вміст комірки, в якій не записаний ні 0, ні 1). У початковий момент на стрічку записується довільне слово. Зазвичай це слово представляє собою опис деякої задачі, на яких спеціалізується дана машина Тьюринга. Крім стрічки, у машини є читаюча-друкарська головка, яка дозволяє зчитати символ з комірки або записати в ту комірку, що безпосередньо знаходиться під головкою. Машина Тьюринга працює по тактах. Такти машини Тьюринга жодним чином не прив'язуються до одиниць часу, наприклад, секунд.

"Час", що витрачається машиною Тьюринга на те, щоб виконати обробку вхідного слова, вимірюється в тактах. Доцільно і складність завдання пов’язувати з числом тактів, необхідних для його розв’язування на машині Тьюринга. Машина Тьюринга працює за певними правилами. За своєю суттю такі правила однотипні і зводяться до наступного:
- зчитати символ з поточної комірки;
- за поточним станом машини і зчитаним символом визначити новий стан, в який переходить машина;
- записати в комірку новий символ
  або зрушити стрічку на одну комірку вліво,
  або зрушити стрічку на одну комірку вправо,
  або залишити стрічку в попередньому положенні,
- перейти до наступного такту. Машина завершує роботу, якщо вона потрапляє в кінцевий стан.

Наведені правила найзручніше представляти у вигляді таблиці. Візьмемо як приклад таблицю 1:

	0	1	λ
q_1	$\lambda L q_1$	$0LQ_0$	UUQ_1

У цій таблиці використані такі позначення стану машини Тьюринга: q_1, q_0. Початковий стан позначимо як q_1. Стан q_0 в нашому прикладі є кінцевим станом. У верхньому рядку записані символи вхідного алфавіту машини. Таких символів три: 0, 1, λ. Порожній символ λ теж вважається символом, хоча реально не відповідає жодному символу взагалі.

Розглянемо рядок q_1. Нехай у поточній комірці записано 0. На перетині стовпця "0" і рядка q_1 стоїть дія машини - $\lambda L q_1$. Цю дію слід розуміти таким чином:
-перший символ λ вказує на те, що в комірку писати нічого не потрібно,
-другий символ L означає зрушити стрічку на одну комірку вліво,
-третій символ q_1 означає, в який стан машина переходить (у нашому випадку машина залишається в старому стані, значення комірки не змінюється, стрічка зсувається на крок вліво).

Розглянемо тепер стовпець 1. На перетині стовпця 1 і рядка q_1 представлена така дія: $0Lq_1$. За цією дією в поточну комірку буде записаний 0 замість 1; все інше - як і в попередньому прикладі.

Нарешті, остання дія: $\lambda\lambda q_0$. Цю дію треба розуміти так: нічого в комірку не писати (перше λ), стрічку не зрушувати (друге λ), перейти в кінцевий стан q_0.

Виконуючи дані команди, легко помітити, що машина Тьюринга даного типу може виконувати єдину дію - обнуляти числа, тобто просто замінювати 1 на 0.

На практиці машини Тьюринга використовуються для аналізу алгоритмів. Наприклад, питання типу: "Чи можна вирішити цю задачу?" є по суті питанням

"Чи можна для цього завдання побудувати машину Тьюринга?" Тепер можна сформулювати завдання для лабораторної роботи.

Узагальнена структура машини Тьюринга

Існує низка варіантів детермінованих машин Тьюринга: однострічкова, багатострічкова, універсальна та ін. Відмінність цих варіантів не принципова, вони зумовлені пошуком способів зменшення часової складності.

Модель однострічкової детермінованої МТ задається шісткою:

$$M = \left(A, Q, q_1, q_0, a_0, p\right),$$

де A – скінченна множина символів зовнішнього алфавіту,
Q – скінченна множина символів внутрішнього алфавіту,
q_0 – початковий стан, $q_0 \in Q$,
q_f – кінцевий стан, $q_f \in Q$,
a_0 – позначення порожньої комірки стрічки,
p – програма, яка складається з набору команд: $A \times Q \to A\left\{L, R, E\right\}Q$,
де L – зсувати головку вліво,
R – зсувати головку вправо,
E – головка залишається на місці.

Особливості роботи МТ не суперечать властивостям алгоритму. Кроки МТ дискретні і детерміновані, мають властивість масовості. Єдина властивість, яка приймається умовно – це елементарність кроку. У машині Тьюринга крок алгоритму супроводжується декількома операціями: читання символу в комірці стрічки, пошук необхідної команди, виконання команди – операція зі змістом комірки (залишити попередній символ, стерти його, записати новий), операція переміщення головки (залишити на місці, зсунути ліворуч чи праворуч). Всі ці операції, що складають крок алгоритму, є загальнозрозумілими.

Послідовність розв’язання задач на МТ

1. Розміщуються дані на стрічці
2. Визначається необхідність використання додаткових символів і місця їх розташування
3. Розробляється стратегія розв’язання задачі (слід машини Тьюринга)
4. Будується таблиця програми.
5. У відповідності до сліду машини Тьюринга розробляється набір команд, які розміщуються в клітинах таблиці.
6. Мінімізується кількість станів (команд), не змінюючи стратегії розв’язання задачі

1. Виконати операцію Y = (X mod 3), де X, Y – двійкові числа. L=20

			x7	x6	x5	x4	x3	x2	x1	x0	m	y1	y0										

2. Виконати операціюY = (X mod 3), де X, Y – двійкові числа з мінімальною часовою складністю L=10

					y1	y0	m	x7	x6	x5	x4	x3	x2	x1	x0								

3. Виконати операцію Y = (X mod 3), де X, Y – десяткові числа

								x2	x1	x0	m	y											

4. Виконати операцію додавання двох двійкових чисел: Z= (X+Y)

			x3	x2	x1	x0	+	y3	y2	y1	y0	=	z4	z3	z2	z1	z0					

Без збереження вхідних даних.

5. Виконати операцію додавання двох двійкових чисел: Z= (X+Y)

		Z4	z3	z2	z1	z0	=	y3	y2	y1	y0	+	x3	x2	x1	x0						

Без збереження вхідних даних.

6. Виконати операцію додавання двох двійкових чисел: Z= (X+Y)

			x3	x2	x1	x0	+	y3	y2	y1	y0	=	z3	z2	z1	z0						

Зі збереженням вхідних даних.

7. Виконати операцію додавання двох двійкових чисел: Z= (X+Y)

			z3	z2	z1	z0	=	y3	y2	y1	y0	+	x3	x2	x1	x0						

Зі збереженням вхідних даних.

8. Виконати операцію додавання двох двійкових чисел: Z= (X+Y)

								y3	y2	y1	y0	+	x3	x2	x1	x0						

Результат розташувати на місці вхідних даних

---

9. Виконати операцію додавання двох двійкових чисел:Z= (X+Y) з мінімальною часовою складністю

Розташування даних довільне.

---

10. Виконати операцію додавання двох десяткових чисел, Z=(X+Y)

							x1	x0	+	y1	y0	=	z2	z1	z0							

Без збереження вхідних даних.

---

11. Виконати операцію додавання двох десяткових чисел: Z= (X+Y)

				z2	z1	z0	=	y1	y0	+	x1	x0										

Без збереження вхідних даних.

---

12. Виконати операцію додавання двох десяткових чисел: Z= (X+Y)

							x1	x0	+	y1	y0	=	z2	z1	z0							

Зі збереженням вхідних даних.

---

13. Виконати операцію додавання двох десяткових чисел: Z= (X+Y)

				z2	z1	z0	=	y1	y0	+	x1	x0										

Зі збереженням вхідних даних.

---

14. Виконати операцію додавання двох десяткових чисел: Z= (X+Y)

								y3	y2	y1	y0	+	x3	x2	x1	x0						

Результат розташувати на місці вхідних даних

---

15. Виконати операцію віднімання двох двійкових чисел, Z=(X-Y)

			x3	x2	x1	x0	-	y3	y2	y1	y0	=	z4	z3	z2	z1	z0					

Без збереження вхідних даних. Числа представлені в прямому коді.

16. Виконати операцію віднімання двох двійкових чисел: Z= (X-Y)

		Z4	z3	z2	z1	z0	=	y3	y2	y1	y0	-	x3	x2	x1	x0						

Без збереження вхідних даних. Числа представлені в прямому коді.

---

17. Виконати операцію віднімання двох двійкових чисел: Z= (X-Y)

			x3	x2	x1	x0	-	y3	y2	y1	y0	=	z3	z2	z1	z0						

Зі збереженням вхідних даних. Числа представлені в прямому коді.

---

18. Виконати операцію віднімання двох двійкових чисел: Z= (X-Y)

			z3	z2	z1	z0	=	y3	y2	y1	y0	-	x3	x2	x1	x0						

Зі збереженням вхідних даних. Числа представлені в прямому коді.

---

19. Виконати операцію віднімання двох десяткових чисел, Z=(X-Y)

							x1	x0	-	y1	y0	=	z1	z0								

X>=Y

---

20. Виконати операцію віднімання двох десяткових чисел, Z=(X-Y)

							x1	x0	-	y1	y0	=		z1	z0							

X<Y

---

21. Виконати операцію переводу формата числа із десяткового в унарний $X_{(10)} \rightarrow Y_{(1)}$,

								x0	=	I	I	.	.	.	I							

---

22. Виконати операцію переводу формата числа із десяткового в двійковий $X_{(10)} \rightarrow Y_{(2)}$,

							x1	x0	=	yn	.	.	.	y0								

23. Виконати операцію переводу формата числа із двійкової в десяткову $X_{(2)} \rightarrow Y_{(10)}$,

				xn	.	.	.	x0	=	yk	.	.	.	y0								

24. Виконати операцію кон'юнкції (AND) двох двійкових чисел: Z= (X v Y),

			x3	x2	x1	x0	v	y3	y2	y1	y0	=	z3	z2	z1	z0						

25. Виконати операцію диз'юнкції (OR) двох двійкових чисел:Z= (X^ Y) з мінімальною часовою складністю

Розташування даних довільне.

26. Представити число X ($X_n X_{n-1} \ldots X_1 X_0$) в двійково-інверсному коді ($X_0 X_1 \ldots X_{n-1} X_n$)

									xn	.	.	.	x1	x0									

Розташування даних довільне.

27. Представити число X ($X_n X_{n-1} \ldots X_1 X_0$) в двійково-інверсному коді ($X_0 X_1 \ldots X_{n-1} X_n$)

					xn	.	.	.	xn	x0	->	x0	x1	.	.	.	xn						

28. Виконати операцію зсуву двійкового числа X вліво на 4 розряди

								x3	x2	x1	x0	L	4									

29. Виконати операцію зсуву двійкового числа X вліво на Y розрядів

								x3	x2	x1	x0	L	y1	y0								

30. Виконати операцію циклічного зсуву двійкового числа X вліво на Y розрядів

								x3	x2	x1	x0	Cl	y1	y0								

31. Виконати операцію циклічного зсуву двійкового числа X вправо на Y розрядів

								x3	x2	x1	x0	Cr	y1	y0								

32. Виконати операцію множення двох десяткових чисел X*Y

									x	*	y											

X<10, Y<10. Розташування результату довільне.

- початкове положення рухомої головки задано

- виберіть початкове положення рухомої головки таким чином, щоб часова складність була найменшою.

Вимоги до програмного забезпечення:

1. Уведення даних із клавіатури і з зовнішнього файлу;
2. Перевірка коректності введених даних;
3. Меню.

Зміст звіту:

1. Титульний лист;
2. Тема завдання;
3. Завдання;
4. Блок-схеми алгоритмів;
5. Роздруківка тексту програми;
6. Роздруківка результатів виконання програми;
7. Аналіз результатів.

Контрольні питання.

1. Дати визначення машини Тьюринга.
2. За якими правилами працює машина Тьюринга?
3. Для чого використовують машини Тьюринга?
4. Перерахуйте множини та інші дані, що однозначно описують довільну машину Тьюринга?
5. Назвіть операції, що складають крок алгоритму машини Тьюринга.

Лабораторна робота № 3

Тема: «Інтерполяція функцій».

Мета: Ознайомлення з інтерполяційними формулами Лагранжа, Ньютона, рекурентним співвідношенням Ейткена, методами оцінки похибки інтерполяції.

Завдання: Закріплення, поглиблення і розширення знань студентів при вирішенні практичних обчислювальних завдань. Оволодіння обчислювальними методами і практичними методами оцінки похибки обчислень. Придбання умінь і навичок при програмуванні та налагодженні обчислювальних завдань на комп'ютері.

Теоретичні основи:

Інтерполяція функцій є одним із фундаментальних розділів обчислювальної математики. До появи комп'ютерів для багатьох практичних обчислень застосовувалися таблиці елементарних функцій (синусів, логарифмів і т. ін.). Для отримання досить точних результатів при значеннях аргументів, розташованих між вузловими точками, для яких дані табличні значення функції, вирішувалося завдання інтерполяції (у перекладі - «між полюсами»). У найбільш простому випадку сусідні точки графіка цієї функції з'єднувалися відрізком прямої (лінійна інтерполяція). Власне, густота точок таблиці і вибиралася з огляду на інтерполяцію. Наприклад, вираз «чотиризначні таблиці» означає, що для будь-якого значення аргументу, а не тільки для зазначених в таблиці в якості вузлових, шляхом інтерполяції, (як правило, лінійної) можна отримати табличне значення функції з точністю до чотирьох значущих цифр. Основоположне значення задачі інтерполяції пояснюється також тим, що багато методів розв'язання задач чисельного диференціювання, інтегрування, розв'язування диференціальних та інтегральних рівнянь зводяться до диференціювання і інтегрування інтерполяційного многочлена. Після появи комп'ютерів значення задачі інтерполяції функцій, заданих таблицею, не втратило актуальності, оскільки в результаті чисельного вирішення складних задач отримують ряд значень шуканої функції при різних значеннях вхідного параметра. Одержання значної кількості таких значень пов'язане з великими витратами машинного часу. Застосування інтерполяції в цьому випадку дозволяє істотно зменшити ці витрати. Однак, на відміну від задачі інтерполяції відомої функції, в цьому випадку інформація про шукану функцію обмежується таблицею її значень. Ця задача є некоректною, оскільки існує нескінченна множина функцій, що мають задане кінцеве число відомих значень. З подібними ж проблемами доводиться стикатися і при розв'язуванні диференціальних та інтегральних рівнянь. Тому можна сформулювати таку тезу: в обчислювальній математиці не існує коректних задач. Існують тільки коректно поставлені завдання, тобто штучно придумані умови, які на практиці, як правило, не виконуються у зв'язку з браком інформації про те, що є шуканим. У зв'язку з цим задача інтерполяції в реальних умовах є найважливішою проблемою обчислювальної математики, розв’язок якої дозволяє знайти ключ до вирішення багатьох інших завдань, необхідних для практики.

Постановка завдання

Нехай деяка функція $f\left(x\right)$ задана своїми значеннями $y_j = f\left(x_j\right)$ на дискретній множині точок $x_j,\ j = 0,...,m$. Потрібно наближено визначити аналітичний вигляд цієї функції і тим самим отримати можливість обчислити її значення в проміжних точках $x \in \left(x_j, x_{j+1}\right)$. Графік, що ілюструє дану задачу, зображений на рис. 1.

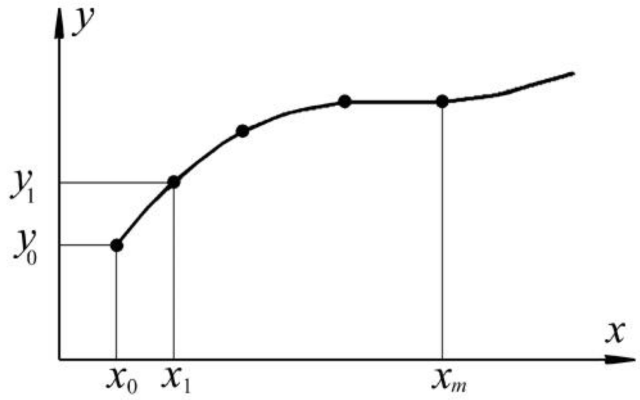


Рис. 1. Інтерполяція функцій

Інтерполюючу функцію будемо шукати у вигляді алгебраїчного многочлена.

$$P_n(x) = \sum_{i=0}^{n} a_i x^i \,. \tag{1.1}$$

Оскільки многочлен $P_n(x)$ у вузлових точках повинен збігатися з заданими значеннями функції, то завдання зводиться до вирішення системи лінійних алгебраїчних рівнянь

$$\sum_{i=0}^{n} a_i x_j^i = y_j, \quad j = k,\dots,k+n \tag{1.2}$$

щодо невідомих a_1 (k - номер початкової вузлової точки, використовуваної в даному розрахунку). Ця система рівнянь має єдине рішення (якщо $m \geq n+k$, і всі x_j різні), оскільки визначник цієї системи – визначник Вандермонда – не дорівнює нулю.

Методи розв’язування задач.
Інтерполяційний многочлен Лагранжа

Розв’язування системи рівнянь (1.2) можна представити у формі інтерполяційного многочлена Лагранжа:

$$P_n\left(x\right) = L_n\left(x\right) = \sum_{j=k}^{k+n} y_j \prod_{\substack{i=k \\ i\neq j}}^{k+n} \frac{x - x_i}{x_j - x_i}. \tag{1.3}$$

В окремому випадку $n = 1$ (лінійна інтерполяція)

$$L_1\left(x\right) = \frac{x - x_{k+1}}{x_k - x_{k+1}} y_k + \frac{x - x_k}{x_{k+1} - x_k} y_{k+1}$$

а при $n = 2$

$$L_2\left(x\right) = \frac{\left(x - x_{k+1}\right)\left(x - x_{k+2}\right)}{\left(x_k - x_{k+1}\right)\left(x_k - x_{k+2}\right)} y_k + \frac{\left(x - x_k\right)\left(x - x_{k+2}\right)}{\left(x_{k+1} - x_k\right)\left(x_{k+1} - x_{k+2}\right)} y_{k+1} +$$

$$+ \frac{\left(x - x_k\right)\left(x - x_{k+1}\right)}{\left(x_{k+2} - x_k\right)\left(x_{k+2} - x_{k+1}\right)} y_{k+2}$$

Неважко помітити, що структура цих формул така, що для кожної вузлової точки $x = x_j$ з вхідного в набору використовуваних формулою вузлових точок, тільки один доданок відмінний від нуля і саме той, в який входить y_j. Крім того, дріб, що входить в цей відмінний від нуля доданок, при $x = x_j$ дорівнює одиниці. Тому $L_n\left(x_j\right) = y_j$.

Інтерполяційний многочлен Ньютона

Спочатку необхідно дати кілька визначень. Для спрощення запису введемо позначення: $f_k = f\left(x_k\right)$. Кінцевою різницею першого порядку функції f в точці x_k називається величина

$$\Delta f_k = f_{k+1} - f_k$$

а кінцевою різницею n-го порядку $\left(n-1\right)$ величина

$$\Delta^n f_k = \Delta^{n-1} f_{k+1} - \Delta^{n-1} f_k \tag{1.4}$$

Звідси, зокрема, випливає, що різниця другого порядку

$$\Delta^2 f_k = \Delta f_{k+1} - \Delta f_k = (f_{k+2} - f_{k+1}) - (f_{k+1} - f_k) = f_k - 2f_{k+1} + f_{k+2}$$

Розділеними різницями нульового порядку називаються значення функції f_k . Розділеною різницею першого порядку називається величина

$$f\left(x_k, x_{k+1}\right) = f\left(x_{k+1}, x_k\right) = \frac{f_{k+1} - f_k}{x_{k+1} - x_k} = \frac{f_k}{x_k - x_{k+1}} + \frac{f_{k+1}}{x_{k+1} - x_k}$$

Розділена різниця n-го порядку визначається через розділені різниці $\left(n-1\right)$-го порядку за рекурентною формулою

$$f\left(x_k, x_{k+1}, x_{k+n}\right) = \frac{f\left(x_{k+1}, x_{k+2}, x_{k+n}\right) - f\left(x_k, x_{k+1}, x_{k+n-1}\right)}{x_{k+n} - x_k} \qquad (1.5)$$

Інший вираз розділеної різниці n -го порядку

$$f\left(x_k, x_{k+1}, x_{k+n}\right) = \sum_{j=k}^{k+n} f_j \left(\prod_{\substack{i=k \\ i \neq j}}^{k+n} \left(x_j - x_i\right) \right)^{-1} \qquad (1.6)$$

Інтерполяційним многочленом Ньютона називається алгебраїчний многочлен

$$l_n\left(x\right) = f\left(x_k\right) + \left(x - x_k\right) f\left(x_k, x_{k+1}\right) + \left(x - x_k\right)\left(x - x_{k+1}\right) f\left(x_k, x_{k+1}, x_{k+2}\right) + \ldots$$
$$\ldots + \left(x - x_k\right)\left(x - x_{k+1}\right)\ldots\left(x - x_{k+n-1}\right) f\left(x_k, x_{k+1}, \ldots, x_{k+n}\right) \qquad (1.7)$$

Цей многочлен тотожно дорівнює многочлену n-го ступеня, записаному в формі Лагранжа або в будь-якій іншій формі в силу єдиності інтерполяційного многочлена. Проте така форма запису дозволяє при необхідності збільшення ступеня многочлена не перебудовувати весь многочлен заново, а тільки додавати додаткові доданки.

Методи оцінки похибки інтерполяції

Оцінка похибки методу

Теоретична оцінка похибки інтерполяції.

Справедлива наступна оцінка похибки інтерполяції

$$f\left(x\right) = P_n\left(x\right) + \frac{\prod_{j=k}^{k+n}\left(x - x_j\right)}{\left(n+1\right)!} f^{\left(n+1\right)}\left(\xi\right) \qquad (1.8)$$

де x_j - вузли сітки $\xi \in \left[x_k, x_{n+k}\right]$, x - значення аргументу, де оцінюється похибка інтерполяції. Для безпосереднього застосування цієї формули необхідно мати верхню оцінку модуля $\left(n+1\right)$- ї похідної функції $f\left(x\right)$. Якщо йдеться про інтерполяцію відомої функції по її табличних значеннях, то така оцінка може бути отримана аналітично. Наприклад, похідна будь-якого порядку від функцій $\sin x$ і $\cos x$ по модулю не перевищує одиниці.

Необхідно відзначити, що значення $\omega_n\left(x\right)$ між вузлами x_j поблизу кінців інтервалу інтерполяції істотно більше (по модулю), ніж у середині. Крім того, при збільшенні n значення $\omega_n\left(x\right)$ швидко зростають. Звідси випливає, що підвищення степеня многочлена може призвести до збільшення похибки інтерполяції, якщо зі збільшенням порядку похідної досить швидко збільшується її величина.

Практична оцінка похибки інтерполяції за результатами чисельного експерименту. У разі, коли інтерпольована функція є результатом чисельного розв’язку деякої задачі, вся інформація про шукану функцію вичерпується її значеннями у вузлових точках. Задача інтерполяції при цьому є некоректною, оскільки може існувати скільки завгодно функцій, графіки яких проходять через дані точки (рис 1.2а). Тобто, рішення задачі може бути отримано тільки з точністю до довільної адитивної складової, що має нульові значення у всіх заданих вузлових точках.

Однак, якщо сітка вибирається довільно, то існування функції, рівної нулю саме в вузлових точках цієї сітки, малоймовірно. Можна також вказати різні способи використання декількох сіток для підвищення надійності одержуваних результатів. У випадку, розглянутому на рис. 1.2б, функція має різкий сплеск на одному з часткових відрізків. При цьому інтерполяційна формула може просто «не помітити» цього сплеску, оскільки у вузлових точках його вплив може бути дуже малим. «Відчути» такий сплеск можна тільки при уточненні результату (наприклад, шляхом підвищення степеня інтерполяційного многочлена, згущенням сітки).

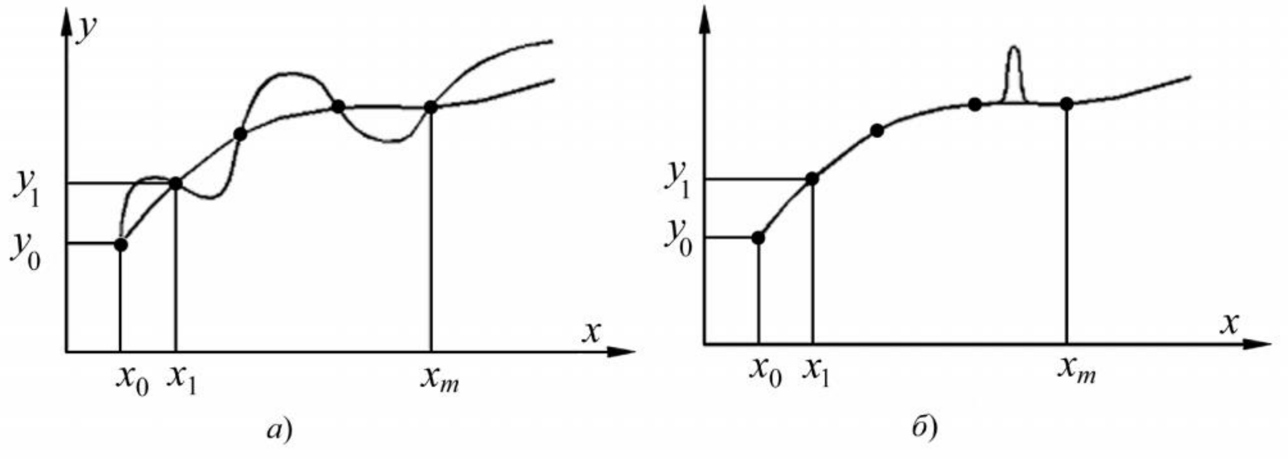


Рис.2. Некоректність задачі інтерполяції

Таким чином, хоча повністю виключити можливість помилки (неправильної оцінки похибки результату), пов’язаної з некоректністю завдання, не можна, але є шляхи зменшення такої можливості.

Розглянемо спосіб оцінки похибки інтерполяції, що не потребує використання будь-якої іншої інформації, крім значень функції у вузлових точках. Для цього на підставі (1.8) представимо математичну модель похибки інтерполяції у наступному вигляді

$$P_n^1\left(x\right)-f\left(x\right)=c\prod_{j=k_1}^{k_1+n}\left(x-x_j^1\right)+\delta_1\left(x\right) \tag{1.9}$$

Тут x_j^1- вузли деякої сітки; $j=0,\dots,N_1$, c - величина, що незалежна від положення вузлів; k_1 - номер початкового вузла, використовуваного інтерполяційною формулою; $\delta_1\left(x\right)$ - додаткова частина похибки, яку вважають малою величиною в порівнянні з першим доданком.

Тепер змінимо сітку, використовуючи нові вузли x_j^2, $j=0,\dots,N_2$. Тоді отримаємо друге рівняння для знаходження невідомих c і $f\left(x\right)$.

$$P_n^2(x) - f(x) = c\prod_{j=k_2}^{k_2+n}\left(x - x_j^2\right) + \delta_2(x) \tag{1.10}$$

Віднімаючи (1.9) з (1.10) і нехтуючи малими, знайдемо c

$$c = \frac{P_n^2(x) - P_n^1(x)}{\Pi_2 - \Pi_1},\ \Pi_i = \prod_{j=k_i}^{k_i+m}\left(x - x_j^i\right) \tag{1.11}$$

оцінку похибки інтерполяції

$$P_n^1(x) - f(x) = \frac{\left(P_n^2(x) - P_n^1(x)\right)\Pi_1}{\Pi_2 - \Pi_1} \tag{1.12}$$

і більш точне значення функції

$$f(x) \approx \frac{P_n^1(x)\Pi_2 - P_n^2(x)\Pi_1}{\Pi_2 - \Pi_1}. \tag{1.13}$$

Формувати різні сітки можна різними способами (наприклад, зменшенням кроку в 2 рази, вибором закону розподілу вузлів). У тому числі для оцінки інтерполяції можна використовувати значення функції в інших вузлах тієї ж самої сітки. Останнє може виявитися більш зручним з практичної точки зору. Спосіб вибору вузлів також може бути різним.

Розглянемо випадок, коли другий набір x_j^2 складається з вузлів x_j^1 з номерами від $k+1$ до $n+k+1$ (тобто $k_1 = k$, $k_2 = k+1$). У цьому випадку згідно (1.12) похибка оцінюється за формулою

$$P_n^1(x) - f(x) \approx \frac{\left[P_n^2(x) - P_n^1(x)\right]\prod_{j=k}^{k+n}\left(x - x_j\right)}{\prod_{j=k+1}^{k+n+1}\left(x - x_j\right) - \prod_{j=k}^{k+n}\left(x - x_j\right)} =$$

$$= -\left[P_n^2(x) - P_n^1(x)\right]\frac{x - x_k}{x_{k+n+1} - x_k}, \tag{1.14}$$

а (1.13) має вигляд

$$f(x) \approx \frac{x_{k+n+1} - x}{x_{k+n+1} - x_k}P_n^1(x) + \frac{x - x_k}{x_{k+n+1} - x_k}P_n^2(x) = P_{n+1}(x) \tag{1.15}$$

Функція (1.15) в дійсності представляє собою інтерполяційний многочлен степеня $n+1$, оскільки:

- $P_{n+1}(x)$ є алгебраїчним многочленом степеня $n+1$;
- у вузлах з номерами від $i = k+1$ до $i = k+n$ обидва многочлени $P_n^1(x_i)$ і, $P_n^2(x_i)$, а, отже, і $P_{n+1}(x_i)$, збігаються з $f(x_i)$;

$$P_{n+1}(x_k) = P_n^1(x_k) = f(x_k);$$

$$P_{n+1}\left(x_{n+k+1}\right) = P_n^2\left(x_{n+k+1}\right) = f\left(x_{n+k+1}\right)$$

Рекурентна формула (1.15) використовується при інтерполяції за схемою Ейткена.

Таким чином, даний спосіб оцінки похибки інтерполяції зводиться до побудови інтерполяційного многочлена $P_{n+1}\left(x\right)$ і порівнянні $P_n\left(x\right)$ значень з $P_{n+1}\left(x\right)$ як з більш точними.

Оцінка похибок вихідних даних та округлення

Крім похибки інтерполяції необхідно враховувати похибку, яка обумовлена помилками самих використовуваних значень функції. Цю похибку, згідно (1.3), можна оцінити за формулою

$$\Delta_n\left(x\right) = \sum_{j=k}^{k+n} \sigma_j A_j, \quad A_j = \left| \prod_{\substack{i=k \\ i\neq j}}^{k+n} \frac{x - x_i}{x_j - x_i} \right| \tag{1.16}$$

або за рекурентним співвідношенням,

$$\Delta_{n+1}\left(x\right) = \left| \frac{x_{k+n+1} - x}{x_{k+n+1} - x_k} \Delta_n^1\left(x\right) \right| + \left| \frac{x - x_k}{x_{k+n+1} - x_k} \Delta_n^2\left(x\right) \right| \tag{1.17}$$

$$\Delta_0^1\left(x\right) = \sigma_k, \qquad \Delta_0^2\left(x\right) = \sigma_{k+1}$$

де σ_j- відомі оцінки похибки значень y_j. Якщо $y_j = f\left(x_j\right)$ - обчислені значення відомої функції, то

$$\sigma_j \le \left|y_j\right| \cdot 10^{-M+1}, \tag{1.18}$$

(M - число десяткових розрядів мантиси машинного слова).

Помилку округлення при застосуванні інтерполяційної формули Лагранжа можна оцінити таким способом. Якщо при програмній реалізації цього способу інтерполяції проводиться додавання методом накопичення, то величина похибки округлення приблизно в n разів більше, ніж та, яка випливає з (1.16) з урахуванням (1.18).

При застосуванні рекурентної формули (1.15) відбувається попарне додавання та накопичення часткових сум по схемі бінарного дерева. Якщо при кожному додаванні доданки приблизно дорівнюють один одному, то накопичення похибки округлення, пов’язаної з вирівнюванням порядків доданків, що суттєво відрізняються між собою, не відбувається. Загальну похибку, пов’язану з машинним представленням чисел, можна тоді оцінити за формулою (1.17).

Відзначимо, що в практичних розрахунках степень інтерполяційного многочлена, як правило, не перевищує 10, тому похибка округлення не перевищує набагато похибку вихідних даних. Однак існує можливість того, що доданки суми мають великі за модулем величини і різні знаки, так що сума має істотно менше значення. Тоді відносна похибка округлення може виявитися дуже великою.

Критерій якості оцінки похибки

Оскільки оцінки (1.12) - (1.15) виведені з припущенням, що величини $\delta_i\left(x\right)$ малі, то необхідна перевірка справедливості цього припущення. Це можна зробити наступним чином. Оцінка похибки за формулою (1.15) зводиться до порівняння значення $P_n\left(x\right)$ із значенням, отриманим при інтерполяції многочленом $\left(n+1\right)$-го степеня $P_{n+1}\left(x\right)$. Тому процес збільшення степеня можна продовжити і отримати значення $P_{n+2}\left(x\right)$. $\Delta_n = P_n\left(x\right) - P_{n+1}\left(x\right)$ представляє собою оцінку похибки інтерполяції значення $P_n\left(x\right)$. Різниця $\Delta_{\Delta n} = P_{n+1}\left(x\right) - P_{n+2}\left(x\right)$ є оцінкою похибки оцінки похибки (рис. 1.3). Відношення $\delta_n = \left|\Delta_{\Delta n} / \Delta_n\right|$ змістовно визначає відносну розмитість оцінки похибки. Якщо $\delta_n \ll 1$, то це означає, що відносна розмитість оцінки мала, і такій оцінці можна довіряти. Якщо ж $\delta_n > 0.3 - 0.4$, то ширина області розмитості порівняна з Δ_n і таку оцінку слід відкинути.

Чисельний експеримент

Застосуємо цей спосіб оцінки до конкретної задачі інтерполяції. Нехай

$$f\left(x\right) = \sin x,\ x_j = \frac{j}{m}\frac{\pi}{2}, \quad y_j = f\left(x_j\right), \quad j = 0,\ldots,m.$$

Результати інтерполяції і оцінки похибки зручно представляти на графіку у вигляді залежності $-\lg\left|P_n - P_{n+1}\right|$ (десяткового логарифма правої частини (1.14)) від $\bar{x} = \left(x - x_j\right) / \left(x_{j+1} - x_j\right)$. На рис. 3 різні криві відповідають різним n (при $j = 2$).

Відзначимо, що збільшення ординати кривої на одиницю при збільшенні степеня інтерполяційного многочлена означає зменшення похибки в 10 разів. Зближення кривих означає те, що за рахунок подальшого збільшення степеня точність підвищити не вдається.

Попарне зближення кривих пояснюється тим, що функція $\sin x$ – непарна, і в її розкладанні за степенями x присутні тільки непарні члени.

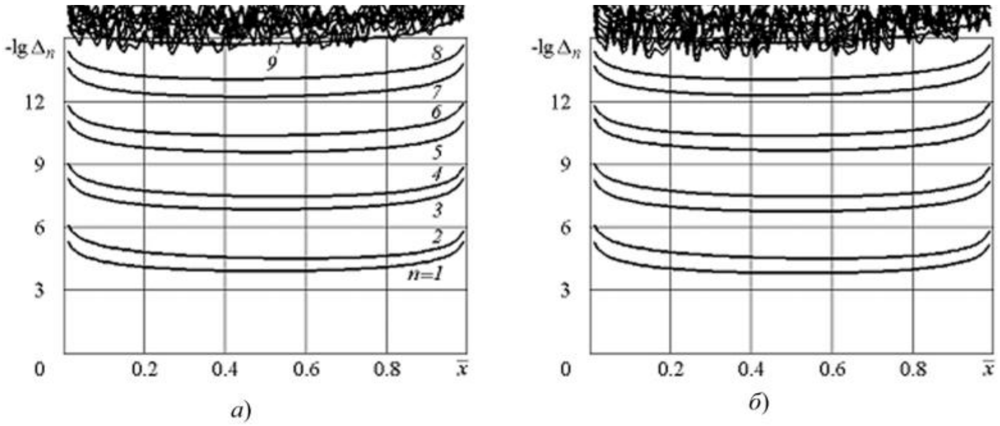


Рис. 3. Результати інтерполяції

З рис. 3а видно, що в результаті інтерполяції даних цього прикладу при *m* = 20 можуть бути отримані значення з похибкою порядку 10-13 з відносною розмитістю близько 0.01. На рис. 3б зображені криві, аналогічні наведеним на рис. 3а, тільки для оцінки похибки використані точні значення функції $\sin x$. Видно, що відмінність графіків на обох малюнках незначна, що говорить про високу точність оцінки похибки за цим методом.

На рис. 4а наведені оцінки похибки (1.16), яка викликана помилками використовуваних значень функції $\sigma_j = \sin x_j \cdot 10^{-15}$ (тут використана подвійна точність). Ця похибка, як неважко помітити, істотно перевищує значення σ_j. При інтерполяції на відрізках, близьких до середини таблиці, ця похибка значно менше (рис. 4б).

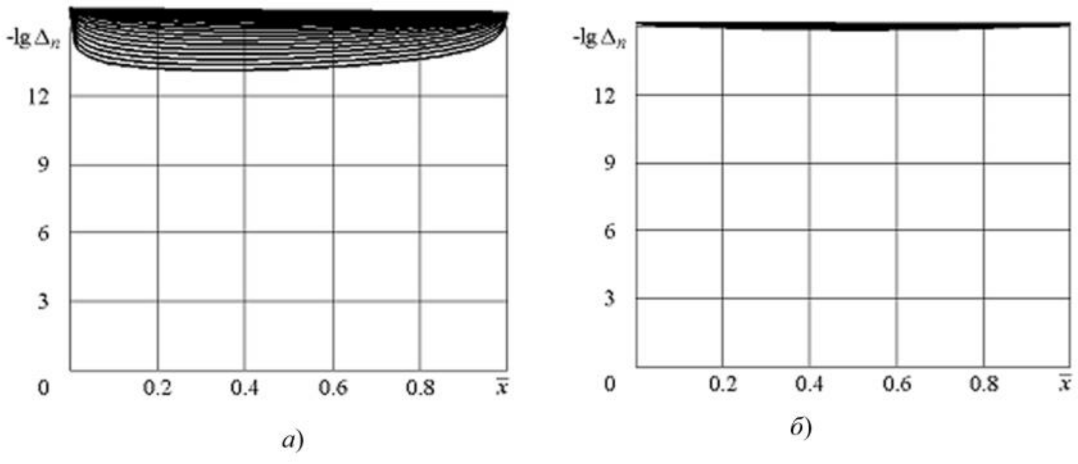


Рис. 4. Вплив похибки початкових даних при інтерполяції

У табл. 1.1 подано результати розрахунків для цього ж прикладу для точки, розташованої посередині між вузлами. Величина $\Delta_n = P_n\left(x\right) - P_{n+1}\left(x\right)$ є оцінена за формулою (1.14) похибка інтерполяції; Δ_n^{exact}– різниця між інтерпольованим і точним значеннями; $k_\Delta = 1 - \Delta_n^{exact} \big/ \Delta_n$– є коефіцієнтом уточнення інтерпольованого значення і він же рівний частковій оцінці похибки оцінки похибки (1.14), тобто фактичної розмитості оцінки (1.14).

Таблиця 1.

n	Δ_n	Δ_n^{exact}	k_Δ
1	$-1.2{\cdot}10^{-4}$	$-1.5{\cdot}10^{-4}$	0.25
2	$-3.0{\cdot}10^{-5}$	$-3.0{\cdot}10^{-5}$	0.01
3	$-1.4{\cdot}10^{-7}$	$-1.7{\cdot}10^{-7}$	0.25
4	$-3.4{\cdot}10^{-8}$	$-3.4{\cdot}10^{-8}$	0.01
5	$-2.7{\cdot}10^{-10}$	$-2.2{\cdot}10^{-10}$	-0.16
6	$-4.3{\cdot}10^{-11}$	$4.4{\cdot}10\text{-}^{11}$	0.01
7	$6.1{\cdot}10^{-13}$	$5.2{\cdot}10^{-13}$	-0.15

n	Δ_n	Δ_n^{exact}	k_Δ
8	$-9.0{\cdot}10^{-14}$	$-9.1{\cdot}10^{-14}$	0.02
9	$-1.8{\cdot}10^{-15}$	$-1.6{\cdot}10^{-15}$	-0.13
10	$1.9{\cdot}10^{-16}$	$2.4{\cdot}10^{-16}$	0.22
11	$5.6{\cdot}10^{-17}$	$4.3{\cdot}10^{-17}$	-0.22
12	$2.8{\cdot}10^{-17}$	$-1.2{\cdot}10^{-17}$	-1.44
13	$8.3{\cdot}10^{-17}$	$-4.0{\cdot}10^{-17}$	-1.48

З таблиці видно, що при $k_\Delta = 0.01$ значення Δ_n і Δ_n^{exact} практично збігаються. При $0.2 < k_\Delta < 0.3$ значення Δ_n і Δ_n^{exact} помітно різняться, але при оцінці похибки такі відмінності можуть вважатися допустимими. При $k_\Delta > 0.3$ значення Δ_n і Δ_n^{exact} розрізняються суттєво, і оцінку похибки за таких умов не можна вважати задовільною.

Таким чином, застосування розглянутого способу оцінки похибки інтерполяції дозволяє не лише з високою точністю оцінити цю похибку (користуючись тільки інформацією, закладеною в табличних даних), але і наближено визначити частку похибки, що міститься в цій оцінці. Це дозволяє судити про якість оцінки, і в разі незадовільного результату відкинути таку оцінку.

Порядок розв'язування задачі на ЕОМ

1) За вказівкою викладача вибрати метод інтерполяції (многочлени Лагранжа (1.3), Ньютона (1.7) або рекурентне співвідношення Ейткена (1.15)).

2) Скласти програму, що обчислює значення заданої функції $y_i = f\left(x_i\right)$ у вузлах інтерполяції $x_i = a + hi$, де $h = \dfrac{\left(b-a\right)}{10}$, $i = 0,1,\ldots,10$, на відрізку $\left[a,b\right]$.

3) Передбачити в програмі оцінку похибки на основі порівняння значень, отриманих за допомогою інтерполяційних многочленів різного степеня.

4) Оцінити розмитість оцінки похибки.

5) Налагодити програму шляхом інтерполяції функції $\sin x$ (див. «Чисельний експеримент»).

6) Застосувати програму для інтерполяції функції, з таблиці 2 за номером у списку.

7) Результат оцінки похибки представити у вигляді графіка (рис. 3, 4) і для одного з значень x у вигляді таблиці 1.

Таблиця 2

Варіанти завдань

№ варианта	$f(x)$	$[a, b]$	№ варианта	$f(x)$	$[a, b]$
1	$\sin x^2$	[0, 2]	**9**	$x \cdot \cos\left(x + \ln\left(1 + x\right)\right)$	[1, 5]
2	$\cos x^2$	[0, 2]	**10**	$10 \cdot \ln 2x \big/ \left(1 + x\right)$	[1, 5]
3	$e^{\sin x}$	[0, 5]	**11**	$\sin x^2 \cdot e^{-\left(x/2\right)^2}$	[0, 3]
4	$1 \big/ \left(0.5 + x^2\right)$	[0, 2]	**12**	$\cos\left(x + \cos^3 x\right)$	[0, 2]
5	$e^{-\left(x + \sin x\right)}$	[2, 5]	**13**	$\cos(x + e^{\cos x})$	[3, 6]
6	$1 \big/ \left(1 + e^{-x}\right)$	[0, 4]	**14**	$\cos\left(2x + x^2\right)$	[0, 1]
7	$\sin\left(x + e^{\sin x}\right)$	[0, 3]	**15**	$e^{\cos x} \cos x^2$	[0, 2]
8	$e^{-\left(x + 1/x\right)}$	[1, 3]			

Вимоги до звіту по лабораторній роботі

Звіт по лабораторній роботі повинен містити:

1) файл вихідного тексту програми;
2) файли результатів для тестового прикладу і для інтерполяції заданої функції;
3) опис алгоритму розрахунку (в текстовій формі та у вигляді блок-схеми) в електронному та роздрукованому вигляді;
4) роздруківку файлів з коментарями;
5) загальні висновки за результатами роботи, що включають результати тестування, отримані оцінки похибки результатів і обґрунтування цих оцінок.

Контрольні питання.

1) Переваги та недоліки різних методів інтерполяції.
2) Оцінка ефективності різних способів оцінки похибки інтерполяції з точки зору їх надійності та практичної застосовності.
3) Вплив похибки початкових даних та округлення на результат інтерполяції.
4) Способи зменшення похибок при інтерполяції.
5) Способи підвищення надійності оцінки похибки інтерполяції.

Лабораторна робота №4

Тема: «Розв’язання нелінійних рівнянь на ЕОМ»

Мета: Метою даного заняття є ознайомлення з методиками та вивчення різних алгоритмів розв’язання нелінійних рівнянь на ЕОМ.

Завдання: Закріплення знань студентів при вирішенні практичних завдань з розв’язування нелінійних рівнянь. Оволодіння методами і практичними навичками розв’язування нелінійних рівнянь на ЕОМ. Набуття умінь і навичок при програмуванні та налагодженні програм для розв’язування нелінійних рівнянь на комп’ютері.

<u>Теоретичні основи:</u>

Загальні поняття та визначення

При вирішенні практичних інженерних задач часто доводиться зустрічатися з розв’язанням рівнянь виду

$$\phi(x) = g(x), \tag{1}$$

або $f(x) = 0$ (2)

де $\phi(x)$, $g(x)$ та $f(x) = 0$– нелінійні функції, визначені на деякій числовій множині X, яка називається *областю допустимих значень рівняння.*

Рівняння виду (1) або (2) називаються *нелінійними рівняннями*. Всі нелінійні рівняння можна поділити на алгебраїчні та трансцендентні (рис.1)

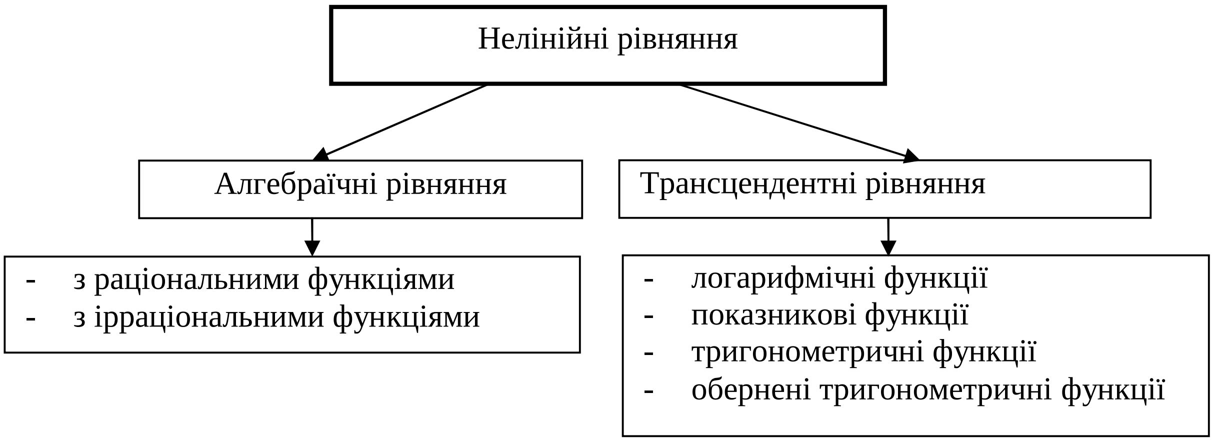


Рис. 1. Класифікація нелінійних рівнянь

Функція називається *алгебраїчною*, якщо для отримання значення функції на заданій множині *X* потрібно здійснити арифметичні операції та піднесення до степеня з раціональним або ірраціональним показником. Рівняння, які містять алгебраїчні функції, називаються *нелінійними алгебраїчними рівняннями*.

До трансцендентних функцій відносять всі неалгебраїчні функції: показникові, логарифмічні, тригонометричні, обернені тригонометричні та інші. Нелінійні рівняння, які містять трансцендентні функції, називаються ***нелінійними трансцендентними рівняннями.***

Розв’язком нелінійного рівняння на ЕОМ називається вектор $\overline{X} = \left\{x_1, x_2, ..., x_n\right\}$, координати якого при підстановці в початкове рівняння перетворюють його в тотожність.

В нелінійному рівнянні виду

$$a_0x^n + a_1x^{n-1} + a_2x^{n-2} + ... + a_{n-1}x + a_n = 0 \tag{3}$$

i-та координата вектора $\overline{X} = \left\{ x_1, x_2, ..., x_n \right\}$ називається i-тим коренем рівняння, а $a_1, a_2, ..., a_m$ – коефіцієнтами рівняння (3).

Метод половинного ділення

Нехай потрібно уточнити єдиний корінь рівняння $f\left(x\right) = 0$, що належить відрізку $\left[a;b\right]$ (відрізок невизначеності).

Точка $c = \dfrac{a+b}{2}$ – середина відрізка $\left[a;b\right]$. Якщо $f\left(c\right) = 0$, то корінь знайдений. В іншому випадку для подальшого розгляду залишаємо ту з половин $\left[a;c\right]$ або $\left[c;b\right]$, на кінцях якої знаки функції $f\left(x\right)$ різні. При цьому виходить послідовність вкладених відрізків, що містять шуканий корінь. На кожному кроці довжина відрізка невизначеності зменшується вдвічі. Метод сходиться завжди. Умовою закінчення пошуку кореня може бути, наприклад,

$$\left|f\left(x\right)\right| < E \text{ або } \frac{\left|b-a\right|}{2^n} < E,$$

де E - точність, $\left[a;b\right]$ - початковий відрізок, n - число ітерацій.

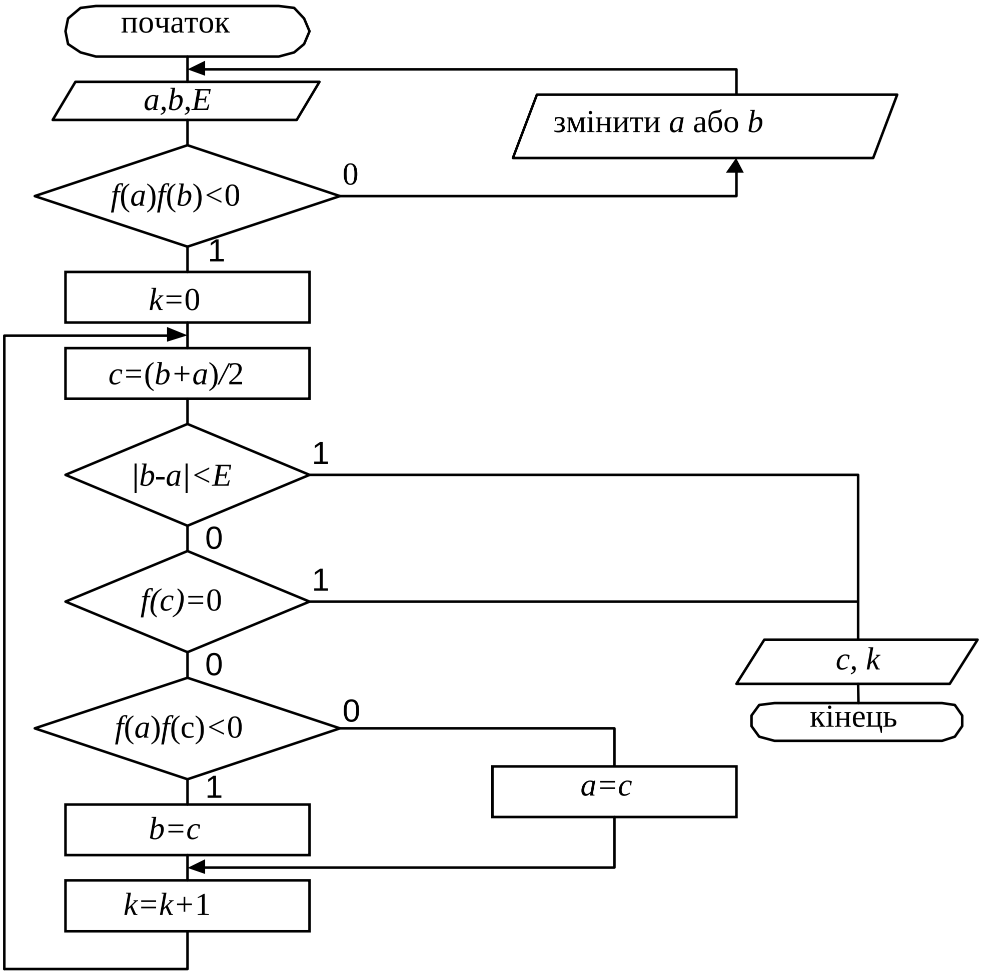


Рис. 2. Схема алгоритму розв’язання нелінійного рівняння методом половинного ділення

Метод хорд

Метод хорд є одним з найбільш поширених методів розв’язання алгебраїчних і трансцендентних рівнянь. В літературі він також зустрічається під назвою "метод лінійного інтерполювання" і "метод пропорційних частин".

Постановка задачі

Розглянемо рівняння $f(x) = 0$, де $f(x)$ неперервна нелінійна функція, яка на відрізку $\left[a,b\right]$ монотонна, диференційована і має єдиний корінь ξ (тобто $f(a) \cdot f(b) < 0$). Потрібно знайти наближене значення кореня ξ з заданою похибкою ε.

Суть метода хорд полягає в тому, що на достатньо малому відрізку $\left[a,b\right]$ дуга функції $f(x)$ замінюється хордою ab, яка її стягує. За наближене значення кореня приймається точка x_1 перетину хорди з віссю Ox (рис.3а).

Рівняння хорди, яка проходить через точки, має вигляд

$$\frac{y - f(x)}{f(b) - f(a)} = \frac{x - a}{b - a} \tag{4}$$

Знайдемо значення x_1, для якого $y = 0$, тобто для нерухомого кінця:

$$x_1 = a - \frac{f(a) \cdot (b - a)}{f(b) - f(a)} \tag{5}$$

Ця формула називається *формулою методу хорд*. Тепер корінь ξ знаходиться всередині відрізка $\left[x_1,b\right]$. Значення кореня x_1 можна уточнити за допомогою метода хорд на відрізку $\left[x_1,b\right]$, тоді нове наближене значення кореня x_2 знаходиться за формулою

$$x_2 = x_1 - \frac{f(x_1) \cdot (b - x_1)}{f(b) - f(x_1)}.$$

Аналогічно для всякого $(i+1)$-го наближення до точного значення кореня ξ даного рівняння використовується формула:

$$x_{i+1} = x_i - \frac{f(x_i) \cdot (b - x_i)}{f(b) - f(x_i)} \tag{6}$$

Процес стягування хордою продовжується багаторазово доти, поки не одержано наближений корінь із заданим ступенем точності

$$\left|x_{i+1} - x_i\right| < \varepsilon \tag{7}$$

де x_{i+1}, x_i – наближені значення коренів рівняння $f(x) = 0$, відповідно на $(i+1)$ і i-му ітераційному кроці; ε – задана точність обчислень. Слід відмітити, що розглянутий випадок (рис. 3а) перетину функцією $f(x)$ відрізка

$\left[a,b\right]$ не є єдиним. Існує ще три варіанти перетину функції, кожний з яких відрізняється напрямком побудови хорд і, відповідно, рухомими кінцями відрізка. Наприклад, на рис. 3а,б рухомий кінець відрізка *a*, а на рис. 3в,г рухомий кінець – b і, відповідно, формула (5) для нього має вигляд:

$$x_1 = b - \frac{f(b)\cdot(b-a)}{f(b)-f(a)}$$

Для автоматизації цього алгоритму необхідно розробити правило для автоматичного вибору рухомого кінця хорди і, відповідно, формули для обчислення наближеного значення кореня. Існує два правила визначення рухомого кінця хорди.

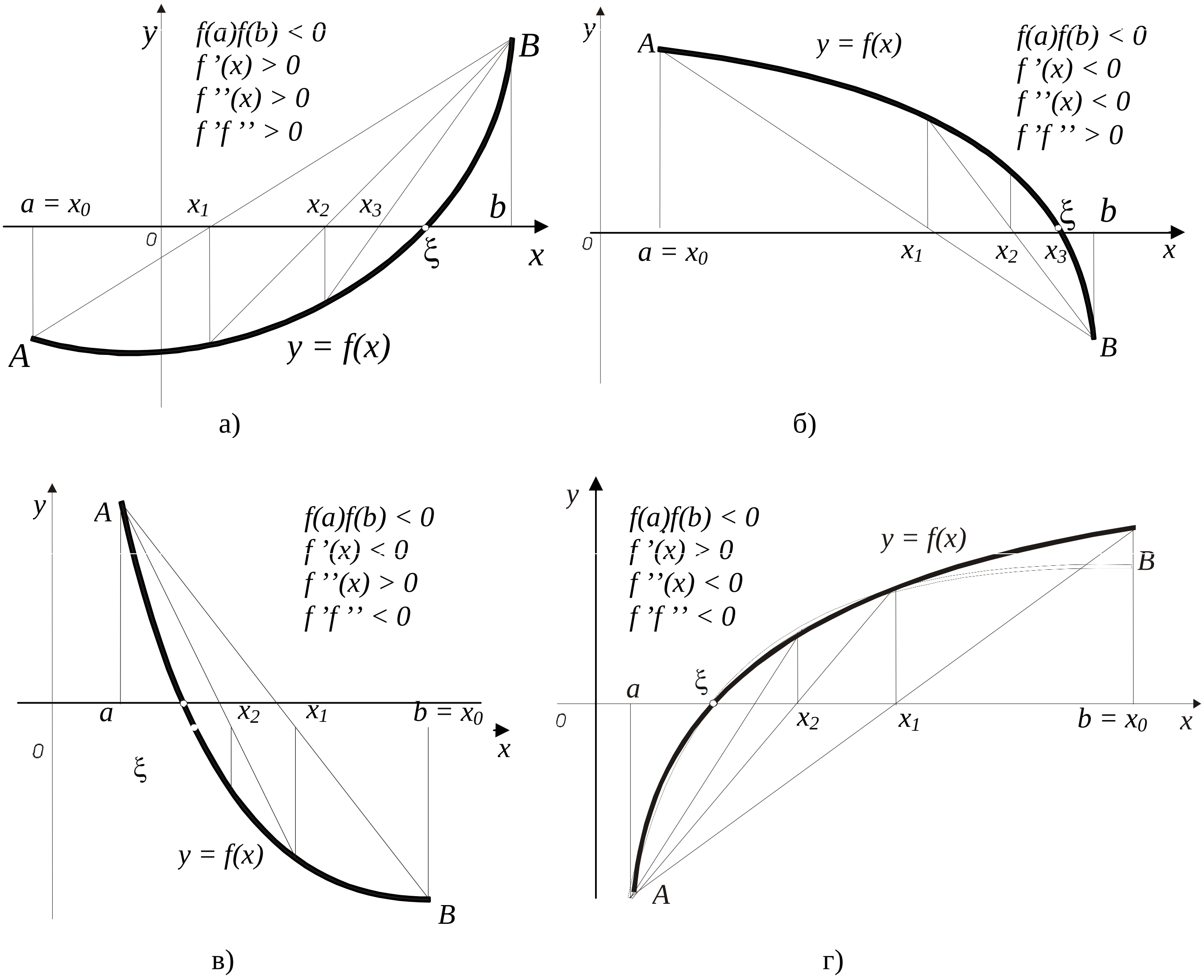


Рис.3. Графічна інтерпретація методу хорд і процедури визначення рухомого кінця хорди

Правило 1.

Нерухомим кінцем відрізка є той, для якого знак функції співпадає із знаком другої похідної. Якщо $f(b)\cdot f''(x) > 0$, то нерухомим є кінець b, а всі наближення до кореня ξ лежать зі сторони кінця a. Якщо $f(a)\cdot f''(x) > 0$, то нерухомим є кінець a, а всі наближення до кореня ξ лежать зі сторони кінця b (рис. 3а,б,в,г).

Правило 2.

Якщо добуток першої на другу похідну функції $f(x)$ більший за нуль: $f'f''>0$, то рухомий кінець a; якщо добуток першої на другу похідну менший за нуль: $f'f''<0$, то рухомий кінець b.

Схема алгоритму розв’язання нелінійного рівняння методом хорд представлена на рис. 4.

Особливості розробки функцій, які реалізують алгоритм методу

1. Метод половинного ділення та метод хорд розробляються як незалежні підпрограми-функції з вхідними параметрами: *a, b,* £ та вихідними: *x, k,* де *x* – наближене значення кореня, *k*– кількість ітерацій.

2. В цих підпрограмах-функціях необхідно передбачити перевірку вхідних даних, наприклад, чи дійсно відрізок вибраний так, що функція на його кінцях має різні знаки.

3. Обчислення перших та других похідних здійснюється за допомогою спеціальних функцій, в яких заданий математичний вигляд похідної.

4. Перед викликом підпрограми-функції, яка реалізує метод, необхідно аналітично визначити кількість коренів. Аналітично чи програмно відокремити корені і в циклі по кількості коренів викликати функцію, яка реалізує метод уточнення коренів так, щоб на екран були виведені всі відрізки, корені на кожному з цих відрізків та кількість ітерацій, за яку був отриманий кожен корінь.

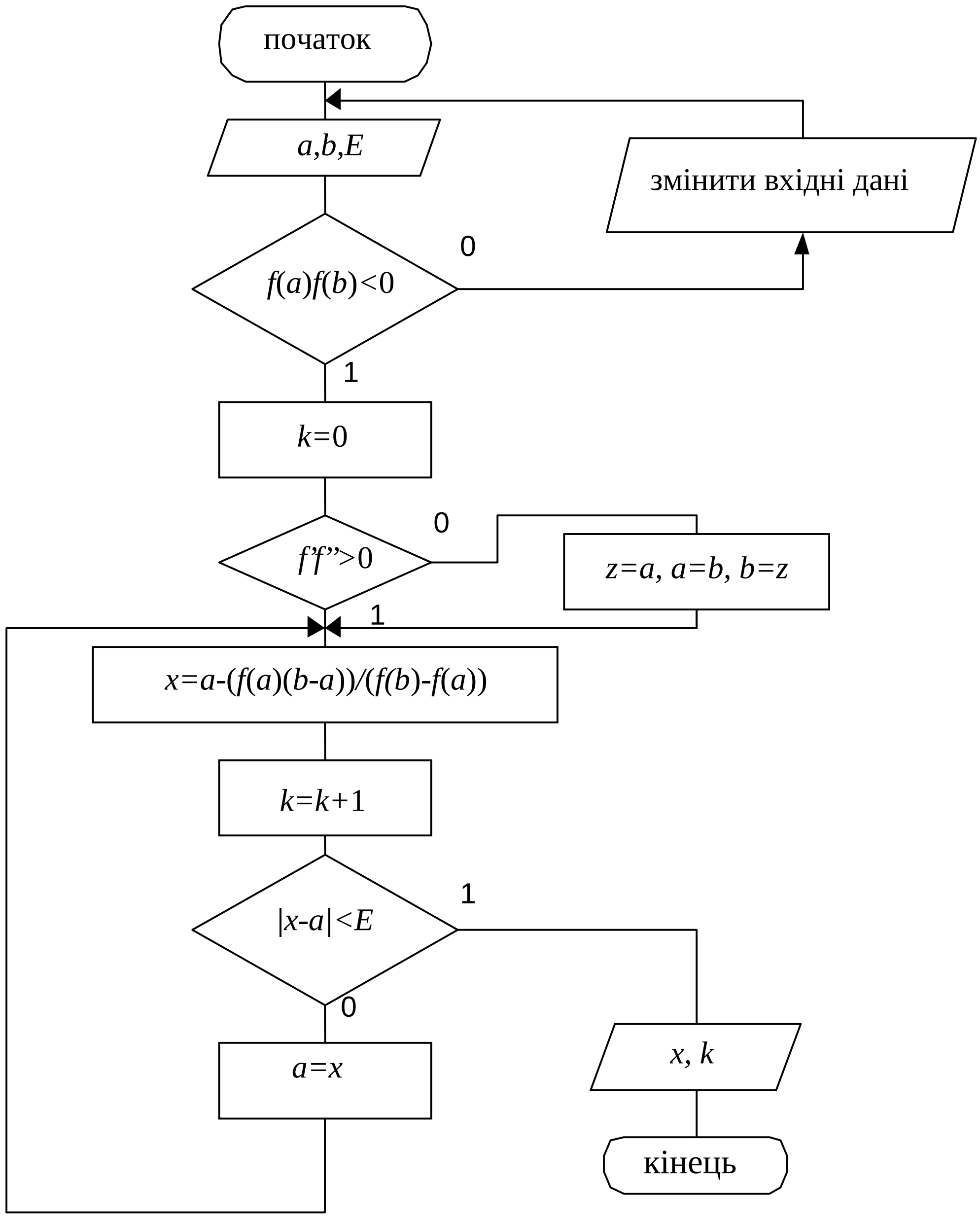


Рис. 4. Схема алгоритму розв'язання нелінійного рівняння методом хорд

Метод Ньютона (метод дотичних)

Метод послідовних наближень, розроблений Ньютоном, дуже широко використовується при побудові ітераційних алгоритмів. Його популярність обумовлена тим, що на відміну від двох попередніх методів замість інтерполяції по двох значеннях функції в методі Ньютона здійснюється екстраполяція за допомогою дотичної до кривої в одній точці.

Постановка задачі

Нехай корінь рівняння $f\left(x\right)=0$ відокремлений на відрізку $\left[a,b\right]$, на якому нелінійна функція $f\left(x\right)$ монотонна і має різні знаки на кінцях відрізка, причому похідні $f'(x)$ та $f''(x)$ неперервні та зберігають постійні знаки на всьому відрізку $\left[a,b\right]$. Потрібно знайти наближене значення кореня ξ з заданою похибкою ε.

Геометричний зміст метода Ньютона полягає в тому, що дуга кривої $y=f(x)$ на відрізку $\left[a,b\right]$ замінюється дотичною до цієї кривої, а наближене значення кореня визначається як точка перетину дотичної з віссю Ox, проведеної з одного з кінців досліджуваного відрізка. Рівняння дотичної має вигляд:

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

Перший випадок. Нехай $f\left(a\right)<0$, $f\left(b\right)>0$, $f'\left(x\right)>0$, $f''\left(x\right)>0$ (рис. 5а) або $f\left(a\right)>0, f\left(b\right)<0$, $f'\left(x\right)<0$, $f''\left(x\right)<0$ (рис. 5б). Проведемо дотичну до кривої $y=f(x)$ в точці $B_0\left(b;f\left(b\right)\right)$ і знайдемо абсцису точки перетину дотичної з віссю Ox. Відомо, що рівняння дотичної в точці $B_0\left(b;f\left(b\right)\right)$ має вид: $y-f\left(b\right)=f'\left(b\right)\cdot\left(x-b\right)$.

Припускаючи $y=0, x=x_1$, отримаємо

$$x_1 = b - \frac{f(b)}{f'(b)} \tag{8}$$

Тепер корінь рівняння знаходиться на відрізку $\left[a,x_1\right]$. Застосовуючи знову метод Ньютона, проведемо дотичну до кривої в точці $B_1\left(x_1;f\left(x_1\right)\right)$ і отримаємо

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)},$$

і так далі (рис. 5).

Даний процес ітераційний, тому формула для будь-якого *n-го* кроку ітерації має вигляд:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{9}$$

В результаті отримана послідовність наближених значень $x_1, x_2, \ldots, x_n, \ldots$, кожний наступний член якої ближчий до кореня ¶, ніж попередній. Однак всі x_n залишаються більше істинного кореня ¶, тобто x_n – наближене значення кореня ¶ з надлишком. Процес визначення кореня продовжується багаторазово доти, поки не одержано наближений корінь із заданим ступенем точності

$$\left|x_{i+1}-x_i\right|<\varepsilon,$$

де x_{i+1}, x_i – наближені значення коренів рівняння $f(x)=0$ відповідно на $(i+1)$ і i-му ітераційному кроці; ε – задана похибка обчислень.

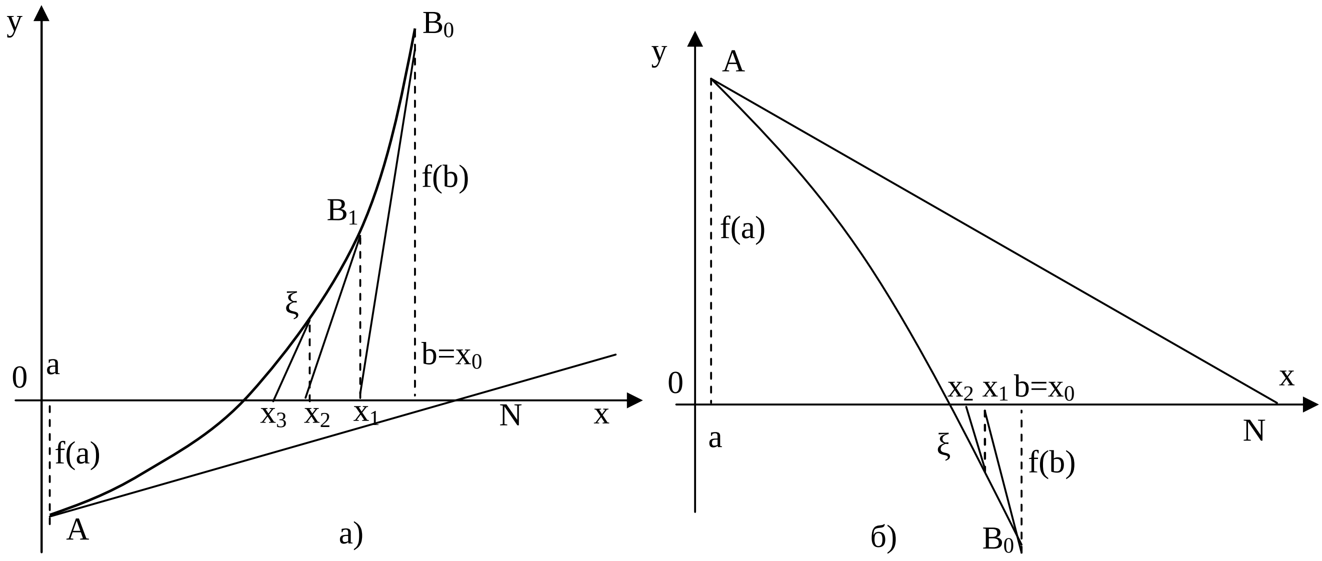


Рис. 5. Геометричний зміст метода Ньютона для випадків, коли
а) функція, яка досліджується, увігнута $\left(f'\left(x\right) > 0, f''\left(x\right) > 0\right)$
б) функція, яка досліджується, опукла $\left(f'\left(x\right) < 0, f''\left(x\right) < 0\right)$

Другий випадок. Нехай $f\left(a\right) < 0$, $f\left(b\right) > 0$, $f'\left(x\right) > 0$, $f''\left(x\right) < 0$ (рис. 6а) або $f\left(a\right) > 0$, $f\left(b\right) < 0$, $f'\left(x\right) < 0$, $f''\left(x\right) > 0$ (рис. 6б). Якщо провести дотичну до кривої $y = f(x)$ в точці *B*, то вона перетне вісь абсцис в точці, яка не належить відрізку $\left[a,b\right]$. Тому проведемо дотичну в точці $A_0\left(a;f\left(a\right)\right)$ і запишемо її рівняння для даного випадку: $y - f\left(a\right) = f'\left(a\right)\left(x - a\right)$.

Припускаючи, що $y = 0, x = x_1$, отримаємо

$$x_1 = a - \frac{f(a)}{f'(a)} \tag{10}$$

Корінь ¶ знаходиться тепер на відрізку $\left[x_1;b\right]$. Застосовуючи знову метод Ньютона, проведемо дотичну до кривої в точці $A_1\left(x_1;f\left(x_1\right)\right)$ і отримаємо

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)},$$

і загалом

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{11}$$

В результаті отримаємо послідовність наближених значень $x_1, x_2, \ldots, x_n, \ldots$ кожний наступний член якої ближчій до істинного кореня ξ, ніж попередній, тобто, x_n – наближене значення кореня ξ з недостачею.

Порівнюючи формули (10), (11) з раніше виведеними, а також враховуючи випадки, які показано на рис. 6а,б помічаємо, що вони відрізняються одна від одної тільки вибором початкового наближення: в першому випадку за x_0 приймався кінець b відрізка, в другому – кінець a.

При виборі початкового наближення кореня необхідно використовувати наступне правило: *за початкову точку слід вибирати той кінець відрізка* $\left[a;b\right]$, *в якому знак функції співпадає зі знаком другої похідної*. В першому випадку $f\left(b\right)\cdot f''\left(x\right) > 0$ і початкова точка $b = x_0$, в другому $f\left(a\right)\cdot f''\left(x\right) > 0$ і в якості початкового наближення беремо $a = x_0$.

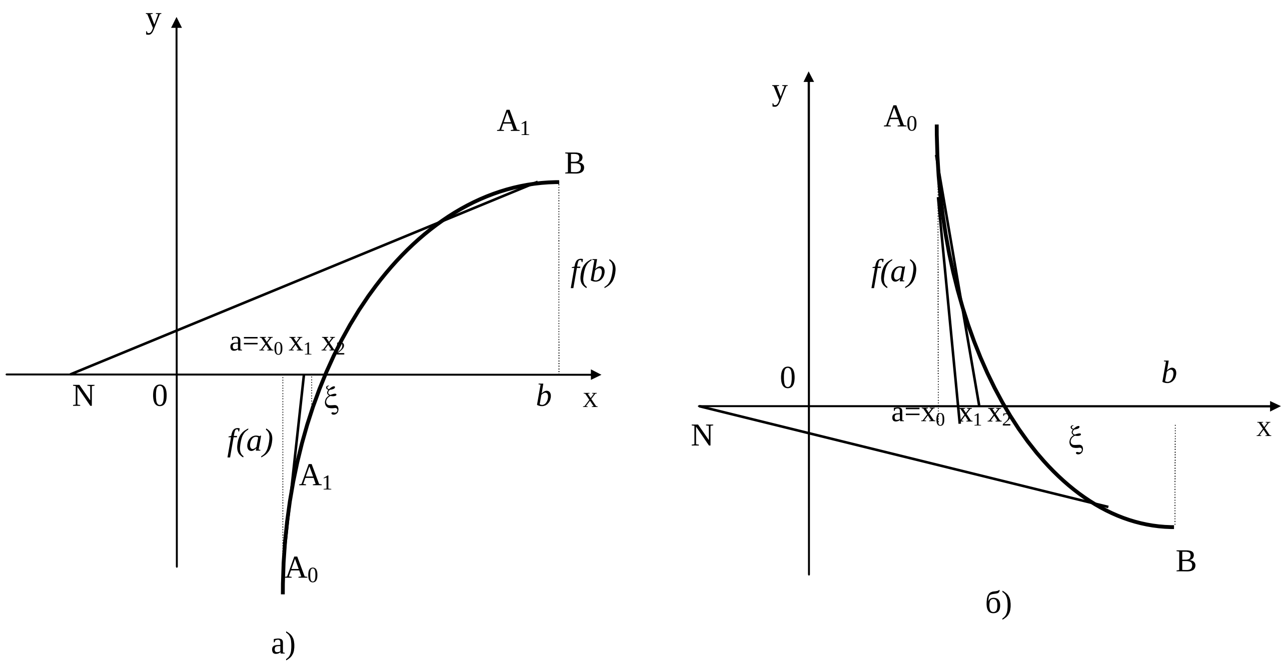


Рис. 6 Геометричний зміст методу Ньютона для випадків, коли
а) функція, яка досліджується, опукла $\left(f'\left(x\right) > 0, f''\left(x\right) < 0\right)$,
б) функція, яка досліджується, увігнута $\left(f'\left(x\right) < 0, f''\left(x\right) > 0\right)$.

Для оцінки похибки можна користуватися загальною формулою

$$\left|\xi - x_n\right| \le \frac{\left|f(x_n)\right|}{m}, \qquad (12)$$

де $m = \min\limits_{[a,\ b]}\left| f'(x) \right|$ (ця формула підходить і до метода хорд).

В тому випадку, коли відрізок $\left[a,b\right]$ настільки малий, що на ньому виконується умова $M_2 < 2m_1$, де $M_2 = \min\limits_{[a,\ b]}\left| f''(x) \right|$, а $m_1 = \min\limits_{[a,\ b]}\left| f'(x) \right|$, точність наближення на n-му кроці інтерполяційного процесу оцінюється наступним чином: якщо $\left|x_n - x_{n-1}\right| < \varepsilon$, $|\xi - x_n| < \varepsilon^2$.

Якщо похідна $f'\left(x\right)$ мало змінюється на відрізку $\left[a,b\right]$, то для спрощення обчислень можна користуватися формулою

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)}, \qquad (13)$$

тобто значення похідної в початковій точці достатньо обчислити тільки один раз.

Процес побудови дотичної продовжується багаторазово доти, поки $\left|x_{i+} - x_i\right| < \varepsilon$, де ε – задана точність обчислень; x_{i+1}, x_i – наближені значення кореня рівняння $f(x) = 0$, відповідно на $(i+1)$ та i - тому ітераційному кроці.

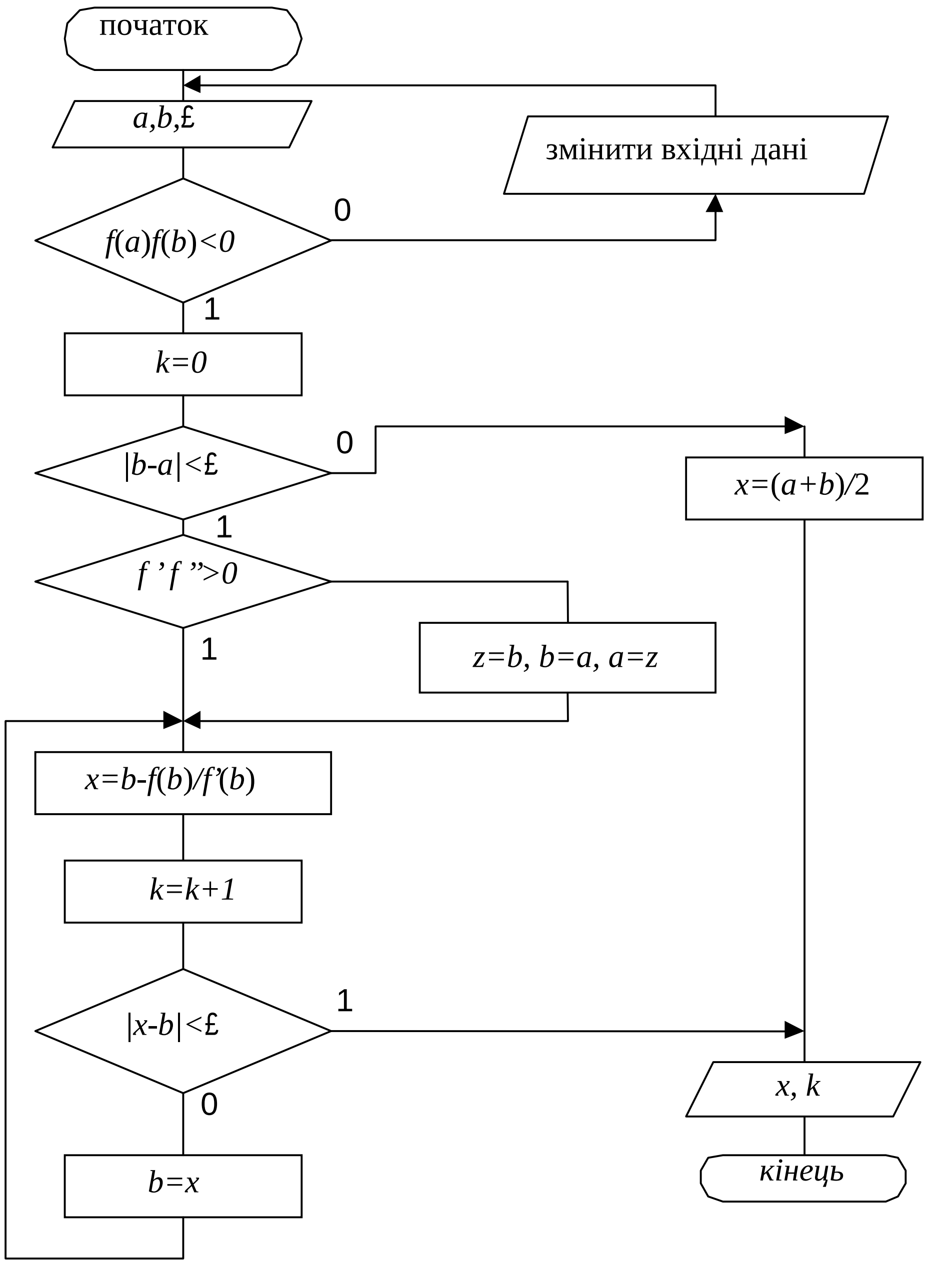


Рис. 7. Схема алгоритму розв’язання нелінійного рівняння методом дотичних

Комбінований метод

Методи хорд і дотичних дають наближення кореня з різних сторін відрізка $\left[a,b\right]$. Тому їх часто використовують в поєднанні один з одним, і процес уточнення кореня ξ нелінійного рівняння (1) проходить швидше .

Постановка задачі

Нехай дано рівняння $f(x)=0$, де $f(x)$ – неперервна нелінійна функція, яка на відрізку $\left[a,b\right]$ монотонна, диференційована і має єдиний корінь ξ (тобто $f(a)\cdot f(b)<0$). Потрібно знайти наближене значення кореня ξ з заданою похибкою ε.

Використаємо комбінований метод хорд і дотичних з урахуванням поведінки функції на відрізку $\left[a,b\right]$. Якщо $f'\left(x\right)f''\left(x\right)>0$, то метод хорд дає наближення кореня з недостачею, а метод дотичних – з надлишком (рис.7.а,б).

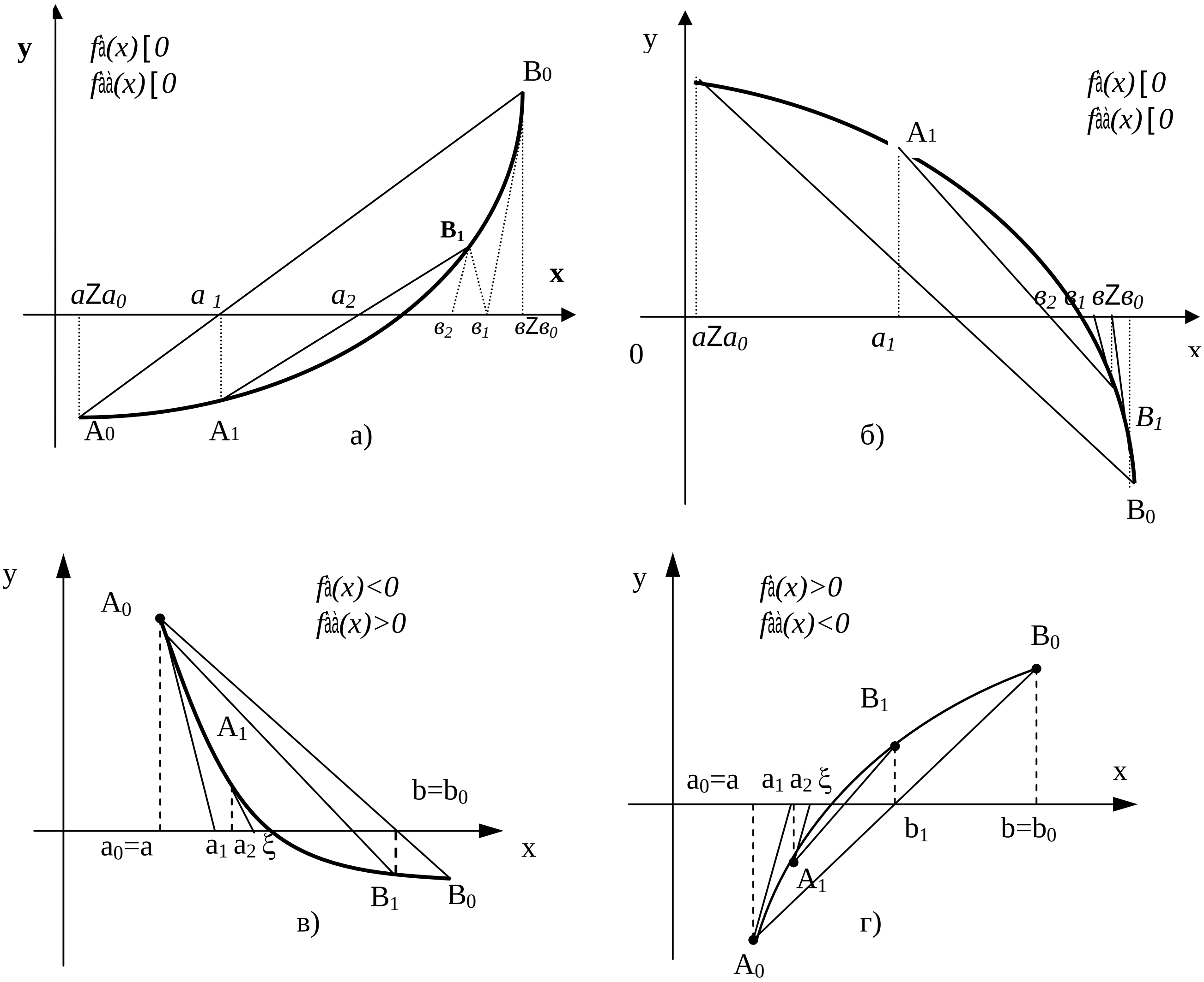


Рис. 7. Геометричний зміст комбінованого методу

Якщо ж $f'\left(x\right)f''\left(x\right)<0$, то методом хорд отримуємо значення кореня з надлишком, а методом дотичних – з недостачею (рис.7.в,г). Однак в усіх випадках справжній корінь ξ знаходиться між наближеними коренями, які отримані за методом хорд і методом дотичних, тобто виконується нерівність $a<x_n<\xi<\overline{x}_n<b$, де x_n – наближене значення кореня з недостачею, $\overline{x}_n$ – з надлишком.

Суть методу полягає в тому, що на досить малому відрізку $\left[a,b\right]$ (отриманому при відокремлені коренів) дуга функції $f(x)$ з одного кінця відрізка стягується хордою, а з другого – дотичною. Тобто, якщо сумістити обидва методи, то після знаходження коренів відрізок $[a,b]$ на кожному кроці ітерації звужується шляхом переносу кінців відрізка $[a,b]$ в точки перетину хорди та дотичної з віссю Ox.

Наближене значення кореня нелінійного рівняння визначається відповідно до таких правил:

Правило 1. Якщо добуток першої на другу похідну функції $f(x)$ більший за нуль: $f'(x)f''(x) > 0$, (рис. 7 а,б) то рухомим для методу хорд є кінець a, і наближене значення кореня з боку кінця a обчислюється за формулою хорд:

$$\overline{x}_{n+1} = a_n - \frac{f(a_n) \cdot (b_n - a_n)}{f(b_n) - f(a_n)}. \tag{14}$$

Для методу дотичних рухомим є кінець b, і наближене значення кореня обчислюється за формулою дотичних:

$$\overline{\overline{x}}_{n+1} = b_n - \frac{f(b_n)}{f'(b_n)}. \tag{15}$$

Правило 2. Якщо добуток першої на другу похідну функції $f(x)$ менший за нуль: $f'(x)f''(x) < 0$ (рис. 7 в, г), то рухомим для методу хорд є кінець b, і наближене значення кореня з боку кінця b обчислюється за формулою хорд:

$$\overline{\overline{x}}_{n+1} = b_n - \frac{f(b_n) \cdot (b_n - a_n)}{f(b_n) - f(a_n)}. \tag{16}$$

Для методу дотичних рухомим є кінець a, і наближене значення кореня обчислюється за формулою дотичних:

$$\overline{x}_{n+1} = a_n - \frac{f(a_n)}{f'(a_n)}. \tag{3.17}$$

Комбінований метод дуже зручний при оцінці похибки обчислень. Ітераційний процес продовжується доти, поки не стане виконуватися нерівність $\left|\overline{\overline{x}}_n - \overline{x}_n\right| < \varepsilon$. За наближене значення кореня приймають $\xi = \frac{1}{2}\left(\overline{x}_n + \overline{\overline{x}}_n\right)$, де $\overline{x}_n$ і $\overline{\overline{x}}_n$ – наближені значення кореня відповідно з недостачею та з надлишком.

Схема алгоритму методу представлена на рис. 9.

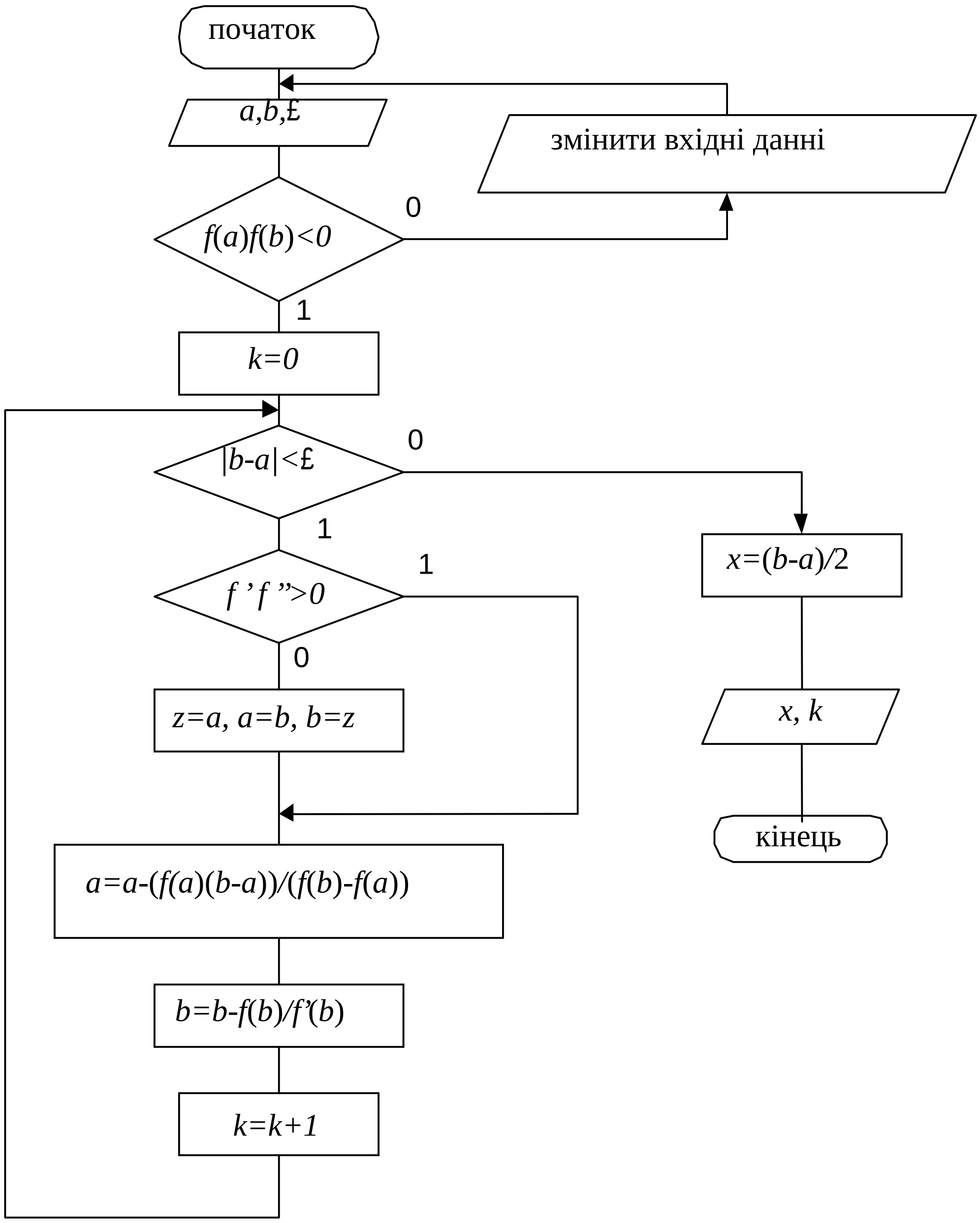


Рис.9. Схема алгоритму розв’язання нелінійного рівняння комбінованим методом

Метод ітерацій (метод послідовних наближень)

Суть методу полягає у заміні початкового рівняння

$$f(x) = 0 \tag{18}$$

еквівалентним йому рівнянням

$$x = \phi(x), \tag{19}$$

Постановка задачі

Нехай задано рівняння $f(x) = 0$, де $f(x)$ – неперервна нелінійна функція. Потрібно визначити корінь ξ цього рівняння, який знаходиться на відрізку $\left[a, b\right]$ з заданою похибкою ε.

Виберемо довільним способом $x_0 \in \left[a, b\right]$ і підставимо його в праву частину рівняння (10); тоді отримаємо $x_1 = \phi(x_0)$. Потім це значення x_1

підставимо знову в праву частину рівняння (9) і отримаємо $x_2 = \phi(x_1)$ (рис. 10 а,б). Повторюючи цей процес, отримаємо послідовність чисел $x_n = \phi(x_{n-1})$. При цьому можливі два випадки:

- послідовність $x_{0,} x_1, \ldots, x_n, \ldots$ *збігається,* тобто має границю, і тоді ця границя буде коренем рівняння (10).
- послідовність $x_{0,} x_1, \ldots, x_n, \ldots$ *розбігається,* тобто не має границі.

Приведемо без доказу теорему, яка виражає умову, за якої ітераційний процес розв'язку нелінійного рівняння методом ітерацій на ЕОМ збігається.

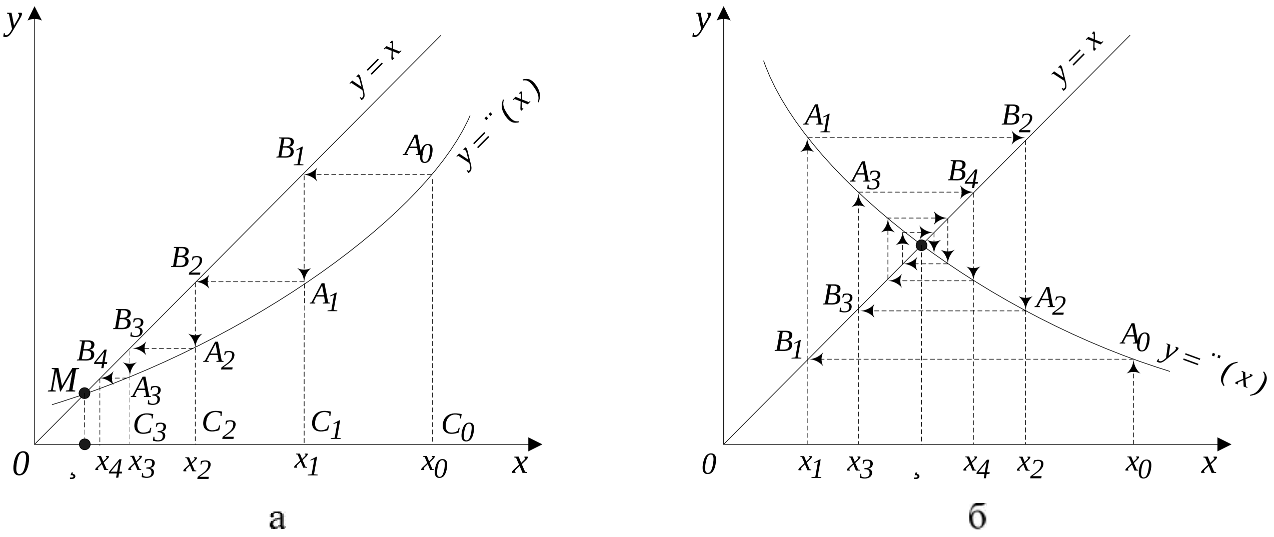


Рис. 10. Геометрична інтерпретація методу ітерацій

Теорема. Нехай на відрізку $\left[a, b\right]$ знаходиться єдиний корінь рівняння $x = \phi(x)$ та у всіх точках цього відрізка похідна $f'(x)$ задовольняє нерівності $\left|\phi'(x)\right| \leq q < 1$. Якщо при цьому виконується і умова $a \leq \phi(x) \leq b$, то ітераційний процес збігається, а за нульове наближення x_0 можна взяти число з відрізка $\left[a, b\right]$.

Розв'яжемо один етап ітерацій. Виходячи із заданого на попередньому кроці значення x_{n-1}, обчислюємо $y = \phi(x_{n-1})$. Якщо $\left|y - x_{n-1}\right| > \varepsilon$, покладемо $x_n = y$ і виконаємо наступну ітерацію. Якщо ж $\left|y - x_{n-1}\right| < \varepsilon$, то обчислення закінчують, за наближене значення кореня приймають величину $x_n = y$.

При використанні методу простих ітерацій основною операцією є вибір функції $\phi(x)$ в рівнянні $x = \phi(x)$, яку слід підібрати так, щоб $\left|\phi'(x)\right| \leq q < 1$ і швидкість збіжності послідовності $\left\{x_n\right\}$ до кореня ξ тим вища, чим менше число q. Схема алгоритму методу ітерацій представлена на рис. 11.

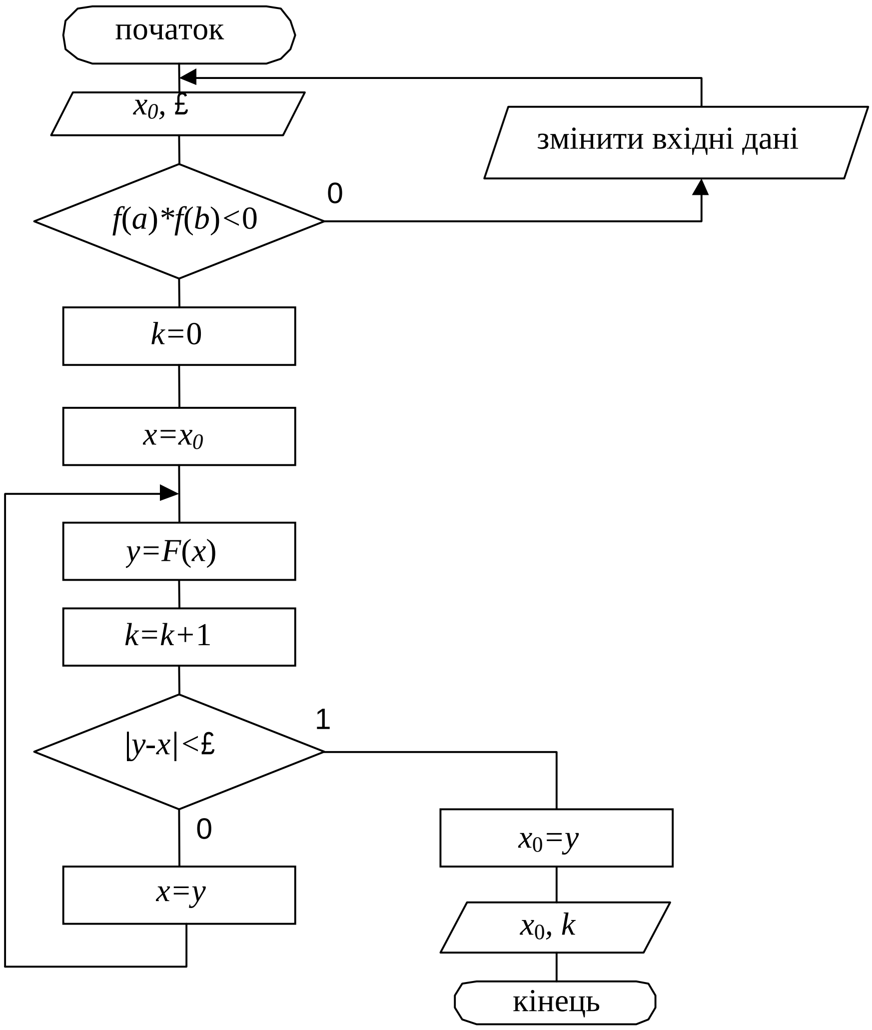


Рис. 11. Схема алгоритму розв’язання нелінійного рівняння методом ітерацій

Порядок виконання роботи

1. Ознайомитись з методикою розв’язання нелінійних рівнянь на ЕОМ.
2. Конкретний метод розв’язання рівняння визначає викладач.
3. Скласти схему алгоритму розв’язання нелінійного рівняння, вибраного з таблиці 1 за вказівкою викладача.
4. Скласти програму розв’язання нелінійного рівняння, користуючись схемою алгоритму.
5. Відокремити корені в рівнянні, що досліджується. (Відокремлення коренів виконати на ЕОМ, розробивши для цього програму).
6. Вибрати значення точності обчислень ε.
7. На кожному проміжку уточнити корені, користуючись розробленою програмою. Початкові дані (проміжок, точність обчислень ε) і результати роздрукувати.
8. Використовуючи блок-схему алгоритму, написати програму розв’язування нелінійного рівняння на мові програмування Pascal.

9. Оформити звіт, що відповідає таким вимогам.

Звіт по лабораторній роботі повинен містити:

1) файл вихідного тексту програми;

2) файли результатів для тестового прикладу і для інтерполяції заданої функції;

3) опис алгоритму розрахунку (в текстовій формі та у вигляді блок-схеми) в електронному та роздрукованому вигляді;

4) роздруківку файлів з коментарями;

5) загальні висновки за результатами роботи, що включають результати тестування, отримані оцінки похибки результатів і обґрунтування цих оцінок.

Таблиця 1 – Варіанти завдань

Метод	Номер варіанту	Рівняння	Примітка
Метод половинного ділення	1	$x^3 - x + 1 = 0$	-1.325
	2	$x^3 + 2x - 4 = 0$	1.180
	3	$x^4 + 5x - 3 = 0$	-1.876; 0.578
	4	$2.2x - 2^x = 0$	0.781; 2.401
	5	$2^x - 2x^2 - 1 = 0$	0.0;0.399;6.352
	6	$2^x - 4x = 0$	0.310; 4.0
Метод хорд	7	$x^3 - x - 3 = 0$	1.213
	8	$x^3 + 8x - 6 = 0$	0.703
	9	$x^3 + 10x - 9 = 0$	0.841
	10	$x^2 - \cos \pi x = 0$	-0.438; 0.438
	11	$x^2 - \sin \pi x = 0$	0.0; 0.787
	12	$\lg x - \frac{1}{x^2} = 0$	1.897
	13	$x^3 - 6x^2 + 9x - 3 = 0$	-4.071; 0.466; 0.993
	14	$x^3 - 12x - 8 = 0$	-0.695; -3.067;3.757
	15	$2\lg x - \frac{x}{2} + 1 = 0$	0.398; 4.682
	16	$x^2 - 20\sin x = 0$	0.0; 2.753

	17	$x - \cos x = 0$	0.739
Комбінований метод	18	$x^3 + 6x - 5 = 0$	0.760
	19	$x^3 - 2x + 7 = 0$	-2.258
	20	$x^3 - 2x^2 + x + 1 = 0$	-0.465
	21	$1.8x^2 - \sin 10x = 0$	-0.567;-0.335; 0.0
	22	$\lg x - \frac{7}{(2x+6)} = 0$	3.473
	23	$2x \ln x - 1 = 0$	1.422
Метод ітерацій	24	$\ln x + (x+1)^3 = 0$	0.187
	25	$x + \lg x = 0.5$	0.672
	26	$tg 1.5x - 2.3x = 0$	
	27	$5 \sin 5x - x = 0$	
	28	$0.83e^{-0.54x} - x = 0$	

Контрольні запитання

1. Дати визначення таким термінам: алгебраїчне та трансцендентне рівняння, розв’язок рівняння, корінь рівняння.
2. Класифікація рівнянь.
3. Суть відокремлення коренів нелінійного рівняння.
4. Умова єдиності кореня рівняння *f(x)*=0 на деякому відрізку $\left[a, b\right]$.
5. Умова відокремлення коренів.
6. Суть методів уточнення коренів.

7. Порівняйте алгоритми методів хорд та комбінованого методу. Який з алгоритмів швидший? Ефективніший?

8. Геометричний зміст комбінованого методу.
9. Геометричний зміст методу Ньютона.
10. Правила визначення рухомого та нерухомого кінця відрізка за допомогою методу хорд.
11. Постановка задачі методу хорд.
12. Постановка задачі методу послідовних наближень.
13. Схема алгоритму розв’язання нелінійного рівняння методом ітерацій.
14. Схема алгоритму розв’язання нелінійного рівняння методом половинного ділення.
15. Схема алгоритму розв’язання нелінійного рівняння комбінованим методом.

16.Відокремити корені рівнянь:

а) $x^3 + 3x^2 - 3 = 0$;

б) $x^3 + 3x^2 - 24x + 1 = 0$;

в) $x^3 - 6x^2 + 9x - 3 = 0$;

г) $x^3 - x - 3 = 0$

д) $x - \cos x = 0$

Лабораторна робота №5

Тема: «Розв’язання систем лінійних алгебраїчних рівнянь».

Мета: Вивчити алгоритми методів розв’язання систем лінійних алгебраїчних рівнянь на ЕОМ

Завдання: Відповідно до варіанту завдання розробити блок-схеми обчислення виразів для лінійного алгоритму, алгоритму, що розгалужується та циклічного алгоритму. У відповідності до блок-схеми створити програму обчислення виразу на алгоритмічній мові Pascal.

Теоретичні основи:

Основні поняття та визначення

Системою лінійних алгебраїчних рівнянь (*СЛАР*) називають систему виду:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1m}x_m = b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2m}x_m = b_2 \\ a_{31}x_1 + a_{32}x_2 + ... + a_{3m}x_m = b_3 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nm}x_m = b_n \end{cases} \qquad (1)$$

де x_i, $(i = \overline{1,n})$ – невідомі; b_i, $(i = \overline{1,n})$ – вільні члени системи; a_{ij}, $(i,j = \overline{1,n})$ – коефіцієнти системи.

В матричному вигляді рівняння (1) має вигляд:

$$\mathbf{A} \times \mathbf{X} = \mathbf{B},$$

де $\mathbf{X}=\left(x_1,x_2,....,x_n\right)^T$ – вектор невідомих; $\mathbf{B}=\left(b_1,b_2,....,b_n\right)^T$ – вектор вільних членів; $\mathbf{A}=\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1m} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2m} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nm} \end{pmatrix}$ – матриця коефіцієнтів СЛАР.

Розв’язком системи лінійних алгебраїчних рівнянь (1) називають вектор $\mathbf{X}$, координати якого $x_1,x_2,...,x_n$ при підстановці у систему, що розв’язують, перетворюють кожне рівняння системи в тотожність.

Кількість невідомих *m* в системі називають *порядком* СЛАР. Систему лінійних алгебраїчних рівнянь називають *сумісною*, якщо вона має хоча б один ненульовий розв’язок. В протилежному випадку СЛАР називають *несумісною.* СЛАР називається *визначеною,* якщо вона має тільки один розв’язок (випадок, коли $m=n$). Систему називають *невизначеною,* якщо вона має безліч розв’язків $\left(m \neq n\right)$. Система називається *виродженою,* якщо головний визначник системи дорівнює нулю. Система називається *невиродженою*, якщо головний визначник системи не дорівнює нулю.

Дві системи називаються *еквівалентними*, якщо ці системи сумісні, визначені і мають однаковий розв’язок.

СЛАР можна розв’язати на ЕОМ чисельними методами, якщо вона сумісна, визначена, невироджена.

До наближених методів відносяться методи, які дозволяють розв’язок системи отримати як границю послідовних k розв’язків системи (1) при $k \to \infty$ виду:

$$\overline{x}=\lim\{\overline{x}^0,\overline{x}^1,\overline{x}^2...\overline{x}^k\},$$

де $\overline{x}^0$– вектор розв’язку 0-го наближення, $\overline{x}^1$– вектор розв’язку 1-го наближення і т. д., $\overline{x}^k$– вектор розв’язку k -го наближення.

Для розв’язання СЛАР наближеними методами найбільший інтерес представляють такі методи:

- метод послідовних наближень;
- метод Гауса-Зейделя;
- метод верхньої релаксації.

Розглянемо особливості загального підходу до розв’язання СЛАР наближеними методами.

Класифікація методів розв’язання СЛАР на ЕОМ

Для розв’язання СЛАР на ЕОМ традиційно використовують дві групи чисельних методів, що представлені на рис.1:

-точні (метод Гауса, метод Гауса з вибором головного елементу, метод Гауса з одиничною матрицею, метод Гауса з перетвореною матрицею, метод Гауса-Халецького, метод Гауса-Жордана, метод Крамера);

-наближені (метод послідовних ітерацій, метод Гауса-Зейделя, метод векторів зміщень).

До **точних методів** відносять методи, які дозволяють отримати точний розв'язок системи (1) за відповідну кількість операцій перетворення без урахування похибок заокруглення.

До **наближених методів** відносять методи, які дозволяють отримати розв'язок системи (1) у вигляді границі послідовності векторів $\lim\limits_{k\to\infty}\left\{X^0, X^1, X^2, ..., X^k\right\}$, яка збігається до точного розв'язку системи, де:

$$X^0 = \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ . \\ . \\ x_n^{(0)} \end{bmatrix}, \quad X^1 = \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ . \\ . \\ x_n^{(1)} \end{bmatrix}, \quad \ldots \quad X^k = \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ . \\ . \\ x_n^{(k)} \end{bmatrix}$$

Рис. 1. Класифікація чисельних методів

Особливості методів Гауса

Найбільш відомими з точних методів розв’язання системи лінійних алгебраїчних рівнянь (1) є методи Гауса, суть яких полягає в тому, що система рівнянь, яка розв’язується, зводиться до еквівалентної системи з верхньою трикутною матрицею. Невідомі знаходяться послідовними підстановками, починаючи з останнього рівняння перетвореної системи. Алгоритми Гауса складаються із виконання однотипних операцій, які легко формалізуються. Однак, точність результату й витрачений на його отримання час у більшості випадків залежить від алгоритму формування трикутної матриці системи. У загальному випадку алгоритми Гауса складаються з двох етапів:

Прямий хід, в результаті якого СЛАР (1), що розв’язується, перетворюється в еквівалентну систему з верхньою трикутною матрицею коефіцієнтів виду:

$$\begin{cases} a_{11}\cdot x_1 & +a_{12}\cdot x_2 & +\ldots+ & a_{1n}\cdot x_n & =b_1 \\ 0\cdot x_1 & +a_{22}\cdot x_2 & +\ldots+ & a_{2n}\cdot x_n & =b_2 \\ \hdashline 0\cdot x_1 & +0\cdot x_2 & +\ldots+ & a_{nn}\cdot x_n & =b_n \end{cases} \qquad (2)$$

Зворотний хід дозволяє визначити вектор розв‘язку, починаючи з останнього рівняння системи (2) шляхом підстановки координат вектора невідомих, отриманих на попередньому кроці.

Відомо декілька різних алгоритмів отримання еквівалентної системи з верхньою трикутною матрицею. Розглянемо найбільш відомі з них.

Метод Гауса з послідовним виключенням невідомих

Метод Гауса з послідовним виключенням невідомих (базовий метод) засновано на алгоритмі, в основі якого лежить послідовне виключення невідомих вектора $\overline{x}$ з усіх рівнянь, починаючи з $\left(i+1\right)$-го, шляхом елементарних перетворень: перемноження обох частин рівняння на будь-яке число, крім нуля; додавання (віднімання) до обох частин одного рівняння відповідних частин другого рівняння, помножених на будь-яке число, крім нуля.

Суть алгоритму розглянемо на прикладі системи, яка складається з трьох лінійних алгебраїчних рівнянь з трьома невідомими:

$$\begin{cases} a_{11}x_1+a_{12}x_2+a_{13}x_3=b_1 \\ a_{21}x_1+a_{22}x_2+a_{23}x_3=b_2 \\ a_{31}x_1+a_{32}x_2+a_{33}x_3=b_3 \end{cases} \qquad (3)$$

1) Перевіримо, щоб принаймні один із коефіцієнтів a_{11}, a_{21}, a_{31} не дорівнював нулю. Якщо, наприклад, $a_{11}=0$, тоді необхідно переставити рівняння так, щоб коефіцієнт при x_1 у першому рівнянні не дорівнював нулю.

2) Обчислюється множник:

$$M_2 = \frac{a_{21}}{a_{11}}. \qquad (2.4)$$

3) Перше рівняння системи (3) множиться на M_2 і віднімається від другого рівняння системи, отриманої після перестановки рівнянь, якщо вона була необхідною. Результат обчислення має вигляд:

$$(a_{21} - M_2 a_{11})x_1 + (a_{22} - M_2 a_{12})x_2 + (a_{23} - M_2 a_{13})x_3 = b_2 - M_2 b_1, \qquad (5)$$

але
$$a_{21} - M_2 a_{11} = a_{21} - \left(\frac{a_{21}}{a_{11}}\right) a_{11} = 0 \qquad (6)$$

Тоді x_1 виключається із другого рівняння.

Позначимо нові коефіцієнти:

$$a'_{22} = a_{22} - M_2 a_{12}; \quad a'_{23} = a_{23} - M_2 a_{13}; \quad b'_2 = b_2 - M_2 b_1 \qquad (7)$$

Тоді друге рівняння системи (3) набуває вигляду:

$$a'_{22} x_2 + a'_{23} x_3 = b_2. \qquad (8)$$

Далі необхідно звільнитися від коефіцієнта a_{31} при x_1 в третьому рівнянні системи (3) за аналогічним алгоритмом

4) Обчислюється множник для третього рівняння:

$$M_3 = \frac{a_{31}}{a_{11}}. \qquad (9)$$

5) Перше рівняння системи (3) множиться на M_3 і віднімається від третього рівняння. Коефіцієнт при x_1 стає нулем, і третє рівняння набуває вигляду:

$$a'_{32} x_2 + a'_{33} x_3 = b_3, \qquad (2.10)$$

де
$$a'_{32} = a_{32} - M_3 a_{12}, \qquad (11)$$

$$a'_{33} = a_{33} - M_3 a_{13}, \qquad (12)$$

$$b'_3 = b_3 - M_3 b_1. \qquad (13)$$

Перетворена таким чином система рівнянь (3) набуває вигляду:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ 0 * x_1 + a'_{22} x_2 + a'_{23} x_3 = b'_2 \\ 0 * x_1 + a''_{32} x_2 + a''_{33} x_3 = b''_3 \end{cases} \qquad (14)$$

Ця система рівнянь еквівалентна початковій і має певні переваги, оскільки x_1 входить тільки до першого рівняння. Спробуємо тепер виключити x_2 з останнього рівняння. Якщо $a_{22} = 0$, а $a_{32} \neq 0$, тоді переставимо друге й третє рівняння так, щоб $a_{22} \neq 0$. Інакше система вироджена і має безліч розв'язків.

7) Обчислюємо множник

$$M_3'' = \frac{a_{32}}{a_{22}}. \qquad (2.15)$$

8) Друге рівняння системи (11) помножується на M_3'' і віднімається від 3-го рівняння:

$$(a_{32} - M_3 a_{22})x_2 + (a_{33} - M_3 a_{23})x_3 = b_3 - b_2 M_2. \qquad (16)$$

При цьому коефіцієнт біля x_2 дорівнює нулю:

$$a'_{32} - M_3 a'_{22} = 0, \qquad (17)$$

$$a''_{33} = a'_{33} - M_3 a'_{23}, \qquad (18)$$

$$b''_3 = b'_3 - M_3 b'_2, \qquad (19)$$

Отримаємо

$$a''_{33}\, x_3 = b''_3. \qquad (20)$$

Замінивши в системі (14) третє рівняння на (20), отримаємо систему рівнянь виду:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ 0 * x_1 + a'_{22}\, x_2 + a'_{23}\, x_3 = b'_2 \\ 0 * x_1 + 0 * x_2 + a''_{33}\, x_3 = b''_3 \end{cases} \qquad (21)$$

Таку систему називають ***системою з трикутною матрицею коефіцієнтів,*** що еквівалентна СЛАР (3). Процес знаходження такої системи називається ***прямим ходом Гауса***. Знайти розв'язок такої системи просто: із 3-го рівняння знайти x_3, підставити результат у друге і знайти x_2, підставити x_2 і x_3 в 1-е рівняння системи (21) і знайти x_1 за формулами:

$$x_3 = \frac{b''_3}{a''_{33}}, \qquad (22)$$

$$x_2 = \frac{b'_2 - a'_{23}\, x_3}{a'_{22}}, \qquad (23)$$

$$x_1 = \frac{b_1 - a_{12}x_2 - a_{13}x_3}{a_{11}}. \qquad (24)$$

Процес знаходження вектора розв’язку системи (3) називають ***зворотним ходом метода Гауса***. На рис. 2 показана схема алгоритму метода Гауса з послідовним виключенням для розв’язання системи із N рівнянь з N невідомими. Ця схема відповідає розглянутому алгоритму і може бути використана при розробці програми. Блок “Перестановка рівнянь так, щоб $a_{nn} \neq 0$ ” означає деякий алгоритм, який дає змогу не допустити помилки “ділення на 0”. Якщо прямувати до можливого зменшення помилок округлення, то можна використати алгоритм з вибором головного елемента.

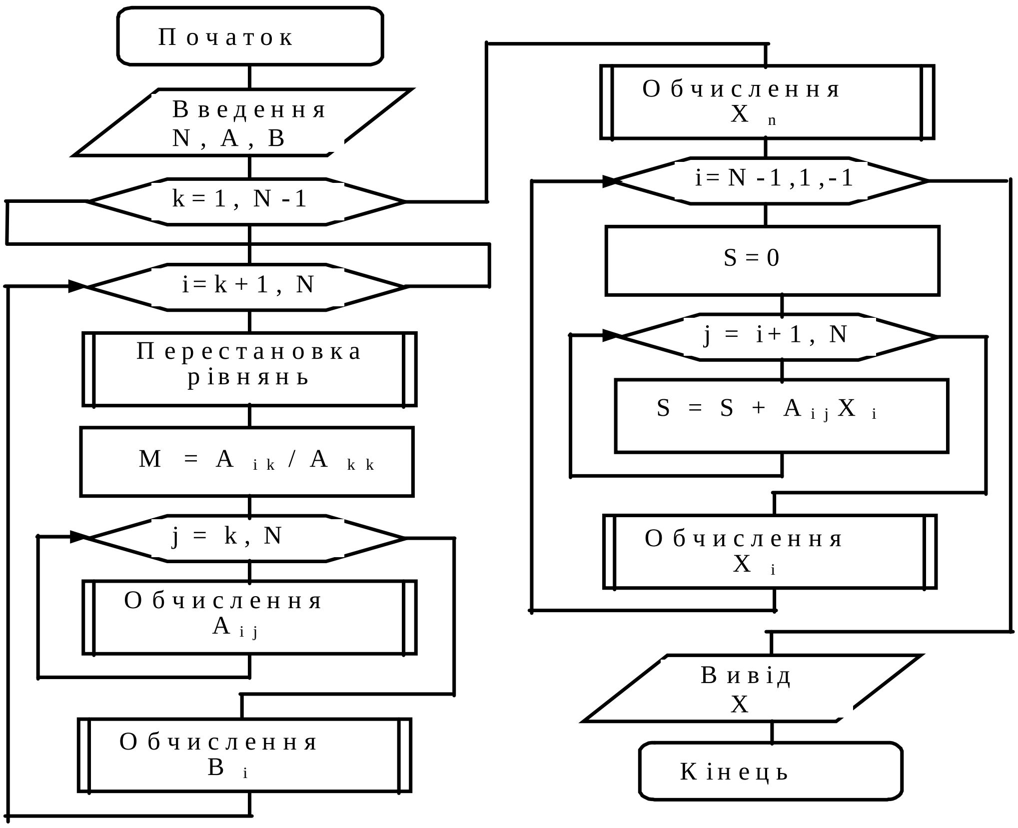


Рис. 2. Схема алгоритму розв’язання системи лінійних алгебраїчних рівнянь методом Гауса.

Призначення індексів в схемі алгоритму (рис. 2):

k – номер рівняння, яке віднімається від інших, а також номер невідомого, яке виключається із залишених *к*-рівнянь;

i – номер рівняння, із якого в даний момент виключається невідоме;

j – номер стовпця.

Метод Гауса з вибором головного елемента

Ідея цього методу виникла у зв'язку з тим, що коефіцієнти СЛАР є параметрами реальних інженерних систем та в більшості є наближеними значеннями, тому що отримані звичайно в результаті вимірювання або як статистичні дані. Для таких систем рівнянь при обчисленні масштабного множника

$$M = \frac{a_{ik}}{a_{kk}} \qquad (25)$$

можлива ситуація при визначені a_{kk}, що ділення наближеного числа a_{ik} на достатньо мале число a_{kk} призводить до різкого збільшення похибки методу. Тому для того, щоб не збільшувати похибку результату, необхідно виконувати такі дії:

1) в системі (1) необхідно знайти з k-го стовпця найбільший за абсолютним значенням коефіцієнт a_{kj} ;

2) переставити k-те рівняння з рівнянням, у якому знаходиться цей максимальний коефіцієнт;

3) масштабний множник буде обчислюватись за формулою (25), де a_{kk} – максимальний коефіцієнт, а тому похибка розв'язання СЛАР у результаті арифметичних операцій не збільшується.

Схема алгоритму метода Гауса з вибором головного елемента (прямий та обернений хід) показана на рис. 3.

Метод Гауса з одиничними коефіцієнтами

В цьому методі зроблена спроба зменшити недоліки перших двох методів, пов'язаних з багаторазовим діленням одного наближеного числа на інше. Для цього перед введенням масштабного множника k-те рівняння системи ділиться один раз на діагональний елемент a_{kk} так, щоб коефіцієнт при $x_k = 1$, а масштабний множник M_i буде дорівнювати a_{kj}. Результатом прямого ходу є система, еквівалентна СЛАР (1), з одиничними коефіцієнтами на головній діагоналі виду:

$$\begin{cases} 1 * x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ 0 * x_1 + 1 * x_2 + ... + a_{2n}x_n = b_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ 0 * x_1 + 0 * x_2 + ... + 1 * x_n = b_n \end{cases} \qquad (26)$$

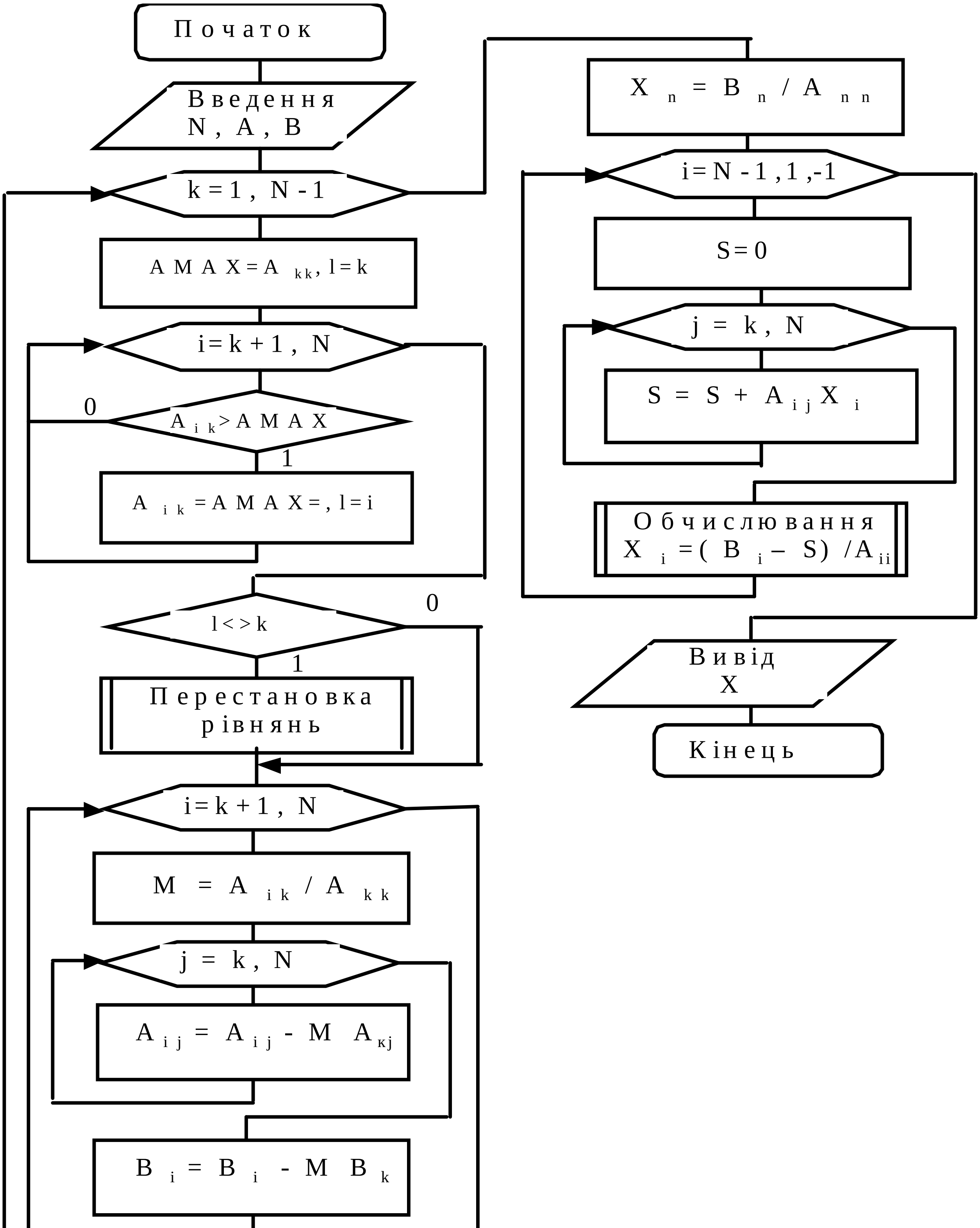


Рис. 3. Схема алгоритму метода Гауса з вибором головного елемента

Дана система схожа на систему (2), яка отримується в результаті прямого ходу базового методу Гауса з послідовним вилученням невідомих і відрізняється від неї тільки діагональними коефіцієнтами. Для отримання такої системи необхідно використовувати алгоритм, який включає в себе наступні етапи:

1. Організація циклу по всіх рівняннях від 1 до *N*-1 (*k* = 1, 2, …, *N*-1).
2. В кожному k-му стовпці визначається номер l-го рівняння з головним елементом (тобто номер l-го рівняння, в якому знаходиться коефіцієнт при x_n зі всіх рівнянь, починаючи з k-го до *N*-го).
3. Якщо номер цього рівняння l не дорівнює k $\left(l <> k\right)$, тоді необхідно переставити місцями l-е рівняння з k-м.
4. Нормування k-го рівняння, тобто ділення всіх коефіцієнтів k-го рівняння на a_{kk} (головний елемент при x_n), включаючи b_n.
5. Перетворення всіх $i-$х рівнянь, починаючи з $\left(k+1\right)$ до N у відповідності з базовим алгоритмом Гауса з метою отримання еквівалентної системи з верхньою трикутною матрицею коефіцієнтів.
6. Кінець циклу по k.

Формула зворотного ходу для систем виду (26) спрощується і має вигляд:

$$\begin{aligned} & x_n = b_n \\ & \dots\dots\dots\dots\dots\dots \\ & x_2 = b_2 - a_{2n}x_n \\ & x_1 = b_1 - a_{1n}x_n - a_{12}x_2 \end{aligned} \tag{27}$$

Схема алгоритму методу Гауса з одиничними діагональними коефіцієнтами наведена на рис. 4.

Загальний підхід до розв'язання СЛАР наближеними методами

Розглянемо систему лінійних алгебраїчних рівнянь виду:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 \\ a_{31}x_1 + a_{32}x_2 + ... + a_{3n}x_n = b_3 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n = b_n \end{cases} \tag{28}$$

Для розв'язання системи (28) методами послідовних наближень необхідно виконати наступні кроки:

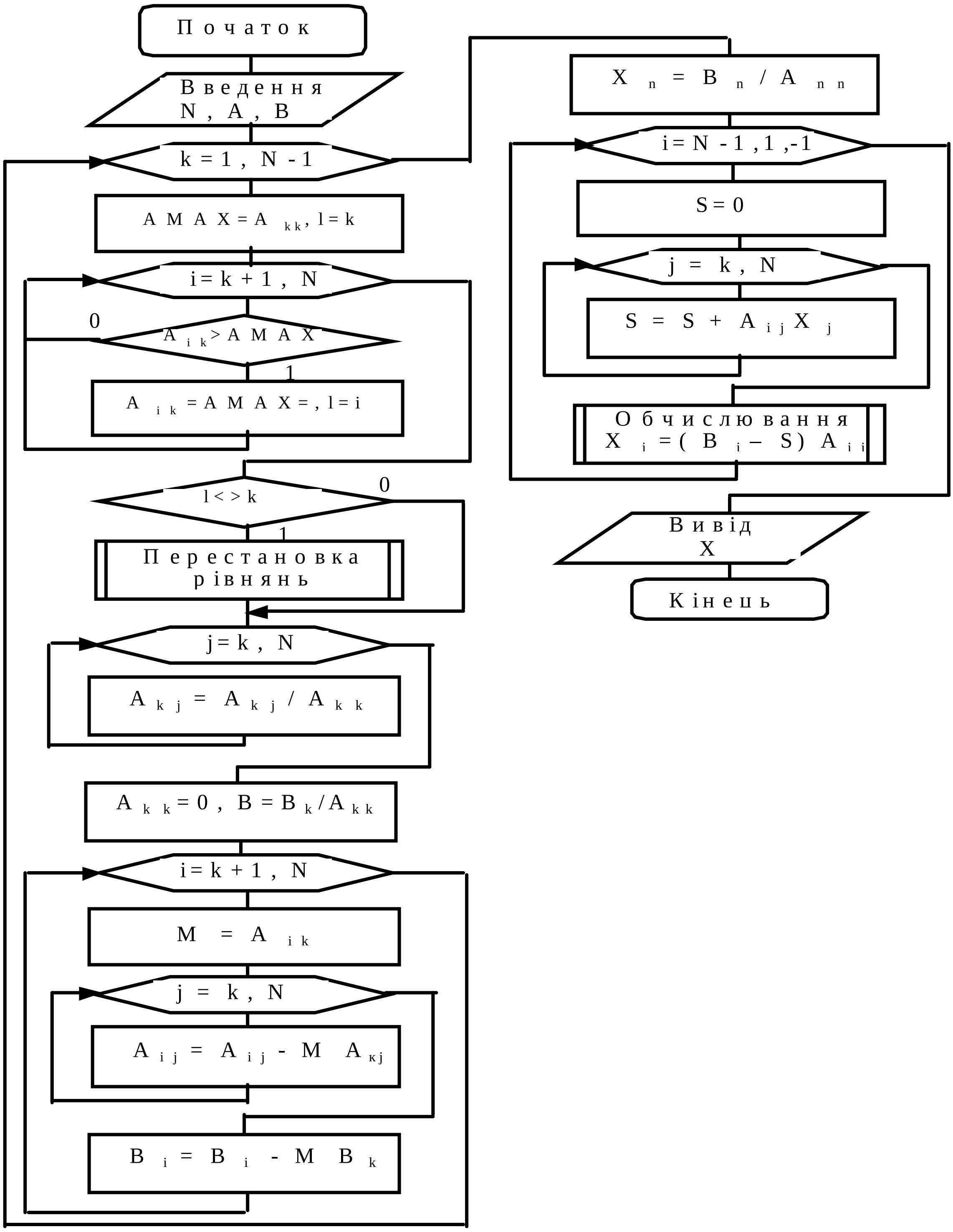


Рис. 4. Схема алгоритму метода Гауса з одиничними коефіцієнтами

1) Кожне рівняння системи розділити на діагональний елемент a_{kk}, де $k = 1,2,...,n$, n – кількість рівнянь в системі, і перетворити кожне рівняння системи відносно координат вектора, індекс якого співпадає з номером рівняння:

$$\begin{cases} x_1 = \dfrac{b_1}{a_{11}} - (\dfrac{a_{12}}{a_{11}}x_2 + \dfrac{a_{13}}{a_{11}}x_3 + ... + \dfrac{a_{1n}}{a_{11}}x_n) \\ x_2 = \dfrac{b_2}{a_{22}} - (\dfrac{a_{21}}{a_{22}}x_1 + \dfrac{a_{23}}{a_{22}}x_3 + ... + \dfrac{a_{2n}}{a_{22}}x_n) \\ x_3 = \dfrac{b_3}{a_{33}} - (\dfrac{a_{31}}{a_{33}}x_1 + \dfrac{a_{32}}{a_{33}}x_2 + ... + \dfrac{a_{1n}}{a_{33}}x_n) \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ x_n = \dfrac{b_n}{a_{nn}} - (\dfrac{a_{n1}}{a_{nn}}x_1 + \dfrac{a_{n2}}{a_{nn}}x_2 + ... + \dfrac{a_{nn-1}}{a_{nn}}x_{n-1}) \end{cases} \qquad (29)$$

2) Нехай $\dfrac{b_k}{a_{kk}} = \beta_k$, а $-\dfrac{a_{ki}}{a_{kk}} = \alpha_{ki}$, де $k = 1,2,...,n$; $i = 1,2,...,n$. Тоді система матиме вигляд:

$$\begin{cases} x_1 = \beta_1 + \alpha_{11}x_2 + \alpha_{12}x_2 + ... + \alpha_{1n}x_n \\ x_2 = \beta_2 + \alpha_{11}x_1 + \alpha_{22}x_2 + ... + \alpha_{2n}x_n \\ x_3 = \beta_3 + \alpha_{11}x_1 + \alpha_{22}x_2 + ... + \alpha_{3n}x_n \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ x_n = \beta_n + \alpha_{11}x_1 + \alpha_{22}x_2 + ... + \alpha_{nn}x_n \end{cases} \qquad (30)$$

Така система називається *зведеною до нормального вигляду*.

3) Представимо систему (30) в матричному вигляді:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ ... \\ x_n \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ ... \\ \beta_n \end{bmatrix} + \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & ... & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & ... & \alpha_{n2} \\ \alpha_{311} & \alpha_{32} & \alpha_{33} & ... & \alpha_{3n} \\ .... & ...... & ..... & ..... & ..... \\ \alpha_{n11} & \alpha_{n2} & \alpha_{n3} & ... & \alpha_{nn} \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ ... \\ x_n \end{bmatrix}, \qquad (31)$$

або векторному $\qquad \bar{x} = \bar{\beta} + \bar{\alpha} * \bar{x}$. $\qquad$ (32)

Якщо деяким чином визначити так званий вектор початкових значень $\bar{x}^{(0)}$, який знаходиться в правій частині (32), то можна отримати певні значення вектора $\bar{x}$.

В якості вектора початкових наближень $\overline{x}^{(0)}$ вибирають:

- вектор, в якого всі координати x_i дорівнюють 0;
- вектор, в якого всі координати x_i дорівнюють 1;
- вектор, координати x_i якого дорівнюють координатам вектора вільних членів β_i;
- координати вектора $\overline{x}$ вибирають в результаті аналізу особливостей об'єкта дослідження та задачі, яка розв'язується.

4) Якщо вектор початкових наближень $\overline{x}^{(0)}$ підставити в праву частину системи (31) або (32), то вона набуває вигляду:

$$\begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \\ \dots \\ x_n^{(1)} \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \dots \\ \beta_n \end{bmatrix} + \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \dots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \dots & \alpha_{n2} \\ \alpha_{311} & \alpha_{32} & \alpha_{33} & \dots & \alpha_{3n} \\ \dots\dots & \dots\dots & \dots\dots & \dots\dots & \dots\dots \\ \alpha_{n11} & \alpha_{n2} & \alpha_{n3} & \dots & \alpha_{nn} \end{bmatrix} * \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ x_3^{(0)} \\ \dots \\ x_n^{(0)} \end{bmatrix}$$

або $\overline{x}^{(1)} = \overline{\beta} + \overline{\alpha} * \overline{x}^{(0)}$,

легко розв'язується, тому що в правій частині містить всі визначені елементи, і дозволяє отримати розв'язок системи, який називається *вектором першого наближення* $\overline{x}^{(1)}$.

5) Перевіряється виконання умови закінчення ітераційного процесу пошуку розв'язку системи (28) виду:

$$|\overline{x}^{(1)} - \overline{x}^{(0)}| \le \varepsilon, \qquad 33),$$

де ε - задана похибка результатів розв'язання задачі.

Якщо умова (33) не виконується, то $x^{(1)}$ підставляється в праву частину (31) або (32) і знаходиться $x^{(2)}$ з системи виду:

$$\begin{bmatrix} x_1^{(2)} \\ x_2^{(2)} \\ x_3^{(2)} \\ \dots \\ x_n^{(2)} \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \dots \\ \beta_n \end{bmatrix} + \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \dots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \dots & \alpha_{n2} \\ \alpha_{311} & \alpha_{32} & \alpha_{33} & \dots & \alpha_{3n} \\ \dots\dots & \dots\dots & \dots\dots & \dots\dots & \dots\dots \\ \alpha_{n11} & \alpha_{n2} & \alpha_{n3} & \dots & \alpha_{nn} \end{bmatrix} * \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \\ \dots \\ x_n^{(1)} \end{bmatrix}$$

або $\overline{x}^{(2)} = \overline{\beta} + \overline{\alpha} * \overline{x}^{(1)}$.

6) Знову перевіряється виконання умови закінчення ітераційного процесу пошуку розв'язку системи (28)

$$| \overline{x}^{(2)} - \overline{x}^{(1)} | \leq \varepsilon .$$

Якщо умова не виконується, то $x^{(2)}$ підставляється в праву частину (31) і знаходиться $x^{(3)}$ з системи виду:

$$\begin{bmatrix} x_1{}^{(3)} \\ x_2{}^{(3)} \\ x_3{}^{(3)} \\ \dots \\ x_n{}^{(3)} \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \dots \\ \beta_n \end{bmatrix} + \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \dots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \dots & \alpha_{n2} \\ \alpha_{311} & \alpha_{32} & \alpha_{33} & \dots & \alpha_{3n} \\ \dots\dots & \dots\dots & \dots\dots & \dots\dots & \dots\dots \\ \alpha_{n11} & \alpha_{n2} & \alpha_{n3} & \dots & \alpha_{nn} \end{bmatrix} * \begin{bmatrix} x_1{}^{(2)} \\ x_2{}^{(2)} \\ x_3{}^{(2)} \\ \dots \\ x_n{}^{(2)} \end{bmatrix}.$$

і т. д.

7) Етапи 4 та 5 повторюються доти, поки на якому-небудь k –ому кроці не виконається умова

$$| \overline{x}^{(k)} - \overline{x}^{(k-1)} | \leq \varepsilon . \tag{2.34}$$

Таким чином, процес пошуку розв'язку системи (28) наближеними методами з заданою похибкою ε є ***ітераційним***, а умовою виходу з цього процесу є умова (34).

Умови збіжності ітераційного процесу

Описаний вище алгоритм дозволяє отримати розв'язок системи (28), близький до точного (з заданою похибкою ε) тільки в тому випадку, коли ітераційний процес пошуку розв'язку СЛАР збігається.

Теорема про збіжність. Ітераційний процес пошуку розв'язку системи лінійних алгебраїчних рівнянь виду (31) наближеними методами збігається, якщо будь-яка канонічна норма матриці $\|\alpha\| < 1$.

Канонічною нормою матриці називається будь-яке дійсне додатне число, яке визначається за такими умовами:

перша канонічна норма – це максимальна з сум модулів елементів матриці коефіцієнтів α по рядках:

$$\|\alpha\|_1 = \max_i \sum_{j=1}^{n} |\alpha_{ij}|, \tag{34}$$

друга канонічна норма – це максимальна з сум модулів елементів матриці коефіцієнтів α по стовпцях:

$$\|\alpha\|_2 = \max_j \sum_{i=1}^{n} |\alpha_{ij}|, \qquad (35)$$

третя канонічна норма – це корінь квадратний з сум квадратів модулів всіх елементів матриці коефіцієнтів α:

$$\|\alpha\|_3 = \sqrt{\sum_i \sum_j |\alpha_{ij}|^2}.$$

Наслідок 1: Ітераційний процес розв'язання системи (2.30) збігається, якщо сума модулів елементів рядків матриці коефіцієнтів α або сума модулів елементів її стовпців менше за одиницю, тобто виконується умова

$$\max_i \sum_{j=1}^{n} |\alpha_{ij}| < 1 \quad \text{або умова} \quad \max_j \sum_{i=1}^{n} |\alpha_{ij}| < 1. \qquad (36)$$

Наслідок 2: Ітераційний процес розв'язання системи (2.30) збігається, якщо елементи головної діагоналі більше суми модулів елементів відповідного рядка, крім діагонального елемента цього рядка, тобто виконується умова:

$$|\alpha_{ii}| > \max_i \sum_{j=1}^{n} |\alpha_{ij}| \quad \text{або умова} \quad |\alpha_{jj}| > \max_j \sum_{i=1}^{n} |\alpha_{ij}|. \qquad (37)$$

Приклад.

Визначити, чи збігається ітераційний процес для системи рівнянь

$$\begin{cases} 8x_1 + x_2 + x_3 = 20 \\ x_1 + 5x_2 - x_3 = 7 \\ x_1 - x_2 + 5x_3 = 7 \end{cases} \quad \text{або} \quad \begin{cases} x_1 = 3{,}25 - 0{,}125x_2 - 0{,}125x_3 \\ x_2 = 1{,}4 - 0{,}2x_1 + 0{,}2x_3 \\ x_3 = 1{,}4 - 0{,}2x_1 + 0{,}2x_2 \end{cases}$$

Матриця $\bar{\alpha}$ даної системи має вигляд:

$$\alpha = \begin{bmatrix} 0 & -0{,}125 & -0{,}125 \\ -0{,}2 & 0 & 0{,}2 \\ -0{,}2 & 0{,}2 & 0 \end{bmatrix}$$

Визначаємо норми:

$$\left\|\alpha\right\|_1 = \max\begin{Bmatrix} \left|\alpha_{11}\right| + \left|\alpha_{12}\right| + \left|\alpha_{13}\right| \\ \left|\alpha_{21}\right| + \left|\alpha_{22}\right| + \left|\alpha_{23}\right| \\ \left|\alpha_{31}\right| + \left|\alpha_{32}\right| + \left|\alpha_{33}\right| \end{Bmatrix} =$$

$$= \max\begin{Bmatrix} 0 + 0{,}125 + 0{,}125 \\ 0{,}2 + 0 + 0{,}2 \\ 0{,}2 + 0{,}2 + 0 \end{Bmatrix} = \max\begin{Bmatrix} 0,25 \\ 0,4 \\ 0,4 \end{Bmatrix} = 0,4 < 1$$

$$\left\|\alpha\right\|_2 = \max\begin{Bmatrix} \left|\alpha_{11}\right| + \left|\alpha_{21}\right| + \left|\alpha_{31}\right| \\ \left|\alpha_{12}\right| + \left|\alpha_{22}\right| + \left|\alpha_{23}\right| \\ \left|\alpha_{31}\right| + \left|\alpha_{32}\right| + \left|\alpha_{33}\right| \end{Bmatrix} = \max\begin{Bmatrix} 0 + 0{,}2 + 0{,}2 \\ 0{,}125 + 0 + 0{,}2 \\ 0{,}125 + 0{,}2 + 0 \end{Bmatrix} =$$

$$= \max\begin{Bmatrix} 0,4 \\ 0,325 \\ 0,325 \end{Bmatrix} = 0,4 < 1 .$$

Таким чином, перша та друга канонічні норми менші за одиницю, тобто ітераційний процес для даної системи збігається.

Розглянемо особливості алгоритмів наближених методів.

Метод послідовних наближень (метод Якобі)

Нехай задана система лінійних алгебраїчних рівнянь виду (28). Метод послідовних наближень (метод Якобі) відноситься до ітераційних методів, тому потребує перетворення даної системи до нормального вигляду (31) та пошуку канонічних норм матриці $\overline{\alpha}$, для того, щоб визначити умови збіжності ітераційного процесу (34)–(36) пошуку розв’язку системи з заданою похибкою ε відповідно до теореми про збіжність. Якщо жодна з умов (34) – (36) не виконується, то дану систему необхідно перетворити за певними правилами та знову перевірити умови збіжності ітераційного процесу (34) – (36). Якщо жодна з умов знову не виконується, то метод послідовних наближень не має сенсу використовувати. Якщо хоча б одна з умов (34)–(36) виконується, то ітераційний процес пошуку розв’язку системи з заданою похибкою ε збігається, і метод послідовних наближень можна використовувати.

По-перше, вибирається певне значення вектора початкових наближень $\overline{x}^{(0)}$, яке підставляється в праву частину системи рівнянь виду:

$$\begin{bmatrix} x_1{}^{(1)} \\ x_2{}^{(1)} \\ x_3{}^{(1)} \\ \dots \\ x_n{}^{(1)} \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \dots \\ \beta_n \end{bmatrix} + \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \dots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \dots & \alpha_{n2} \\ \alpha_{311} & \alpha_{32} & \alpha_{33} & \dots & \alpha_{3n} \\ \dots\dots. & \dots\dots & \dots.. & \dots.. & \dots.. \\ \alpha_{n11} & \alpha_{n2} & \alpha_{n3} & \dots & \alpha_{nn} \end{bmatrix} * \begin{bmatrix} x_1{}^{(0)} \\ x_2{}^{(0)} \\ x_3{}^{(0)} \\ \dots \\ x_n{}^{(0)} \end{bmatrix}, \qquad (38)$$

що легко розв’язується для знаходження вектора розв’язку першого наближення $\overline{x}^{(1)}$, тому що в правій частині містить всі визначені елементи.

По-друге, перевіряється виконання умови закінчення ітераційного процесу виду: $|\ \overline{x}^{(1)} - \overline{x}^{(0)}\ | \leq \varepsilon$,

де ε - задана похибка результатів розв’язання задачі. Якщо умова не виконується, то $x^{(1)}$ підставляється в праву частину системи (31) і знаходиться $x^{(2)}$:

$$\begin{bmatrix} x_1{}^{(2)} \\ x_2{}^{(2)} \\ x_3{}^{(2)} \\ \dots \\ x_n{}^{(2)} \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \dots \\ \beta_n \end{bmatrix} + \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \dots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \dots & \alpha_{n2} \\ \alpha_{311} & \alpha_{32} & \alpha_{33} & \dots & \alpha_{3n} \\ \dots\dots. & \dots\dots & \dots.. & \dots.. & \dots.. \\ \alpha_{n11} & \alpha_{n2} & \alpha_{n3} & \dots & \alpha_{nn} \end{bmatrix} * \begin{bmatrix} x_1{}^{(1)} \\ x_2{}^{(1)} \\ x_3{}^{(1)} \\ \dots \\ x_n{}^{(1)} \end{bmatrix}$$

та знову перевіряється виконання умови закінчення ітераційного процесу виду: $|\ \overline{x}^{(2)} - \overline{x}^{(1)}\ | \leq \varepsilon$.

За аналогією будь-яке $\left(k+1\right)$-е наближення можна обчислити за формулою:

$$x^{(\overline{k}+1)} = \overline{\beta} + \overline{\alpha} * x^{(\overline{k})}, \text{ де } k = 1,2,... \qquad (39)$$

Якщо послідовність $x^{(0)}, x^{(1)}, x^{(2)}, ..., x^{(k)}$, отримана в результаті ітераційного процесу, має границю $x = \lim\limits_{k\to\infty} x^k$, то ця границя є розв’язком системи. Умова закінчення ітераційного процесу має вигляд:

$$|\ x^{(\overline{k}+1)} - x^{(k)}\ | < \varepsilon, \qquad (40)$$

де ε - задана похибка результатів розв’язання системи.

Алгоритмічно перевірка умови (40) представляє собою алгоритм пошуку максимального відхилення між координатами вектора $\overline{x}^{(k)}$ і $\overline{x}^{(k-1)}$ і порівняння його з заданою похибкою ε.

Алгоритм методу послідовних наближень зображено на рис. 5

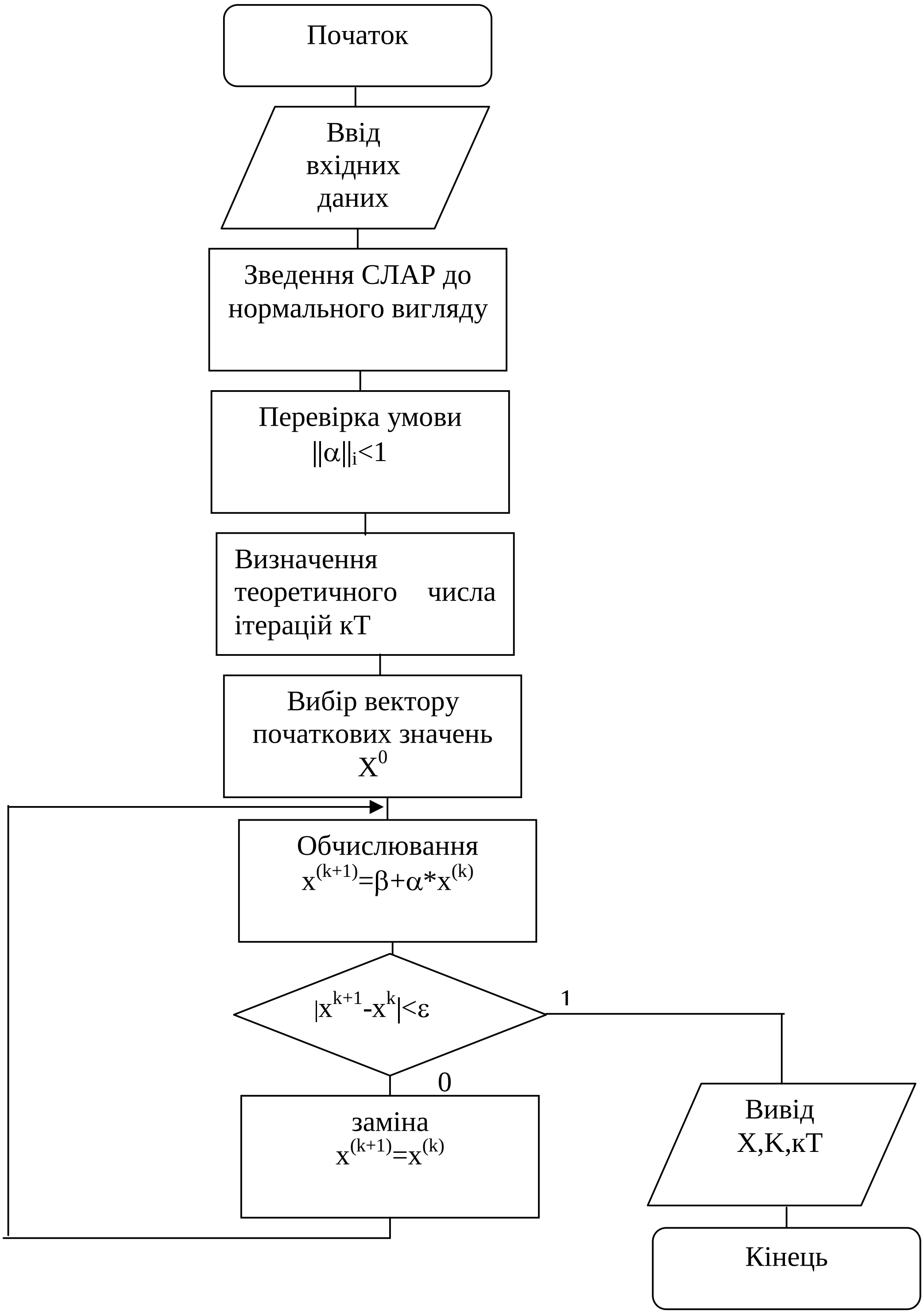


Рис. 5. Схема алгоритму методу послідовних наближень

Оцінка похибки методу Якобі

Якщо задана допустима похибка обчислень ε і x – вектор точного розв’язку системи лінійних рівнянь, а $x_j^{(k)} - k$-те наближення до вектору точного розв’язку, то для оцінки похибки метода послідовних наближень використовується формула:

$$\left\| x_j - x_j^k \right\| \le \frac{\|\alpha\|}{1-\|\alpha\|}^{(k+1)} \|\beta\|, \qquad (41)$$

де $\|\alpha\|$ - одна з 3 норм матриці α; $\|\beta\|$ - аналогічна норма вектора β; k - кількість ітерацій, необхідна для досягнення потрібної точності ε.

Метод Гауса-Зейделя

Метод Зейделя є модифікацію методу послідовних наближень, причому у методі Зейделя при обчисленні i-ої координати вектора розв'язку $(k+1)$-го наближення використовуються значення всіх $\left(i-1\right)$ координат вектора $(k+1)$-е наближення, обчислені раніше. Розглянемо метод більш детально.

Нехай початкова система лінійних алгебраїчних рівнянь приведена до нормального вигляду:

$$\begin{cases} x_1 = \beta_1 + \alpha_{11}x_1 + \alpha_{12}x_2 + ... + \alpha_{1n}x_n; \\ x_2 = \beta_2 + \alpha_{21}x_1 + \alpha_{22}x_2 + ... + \alpha_{2n}x_n; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ x_n = \beta_n + \alpha_{n1}x_1 + \alpha_{n2}x_2 + ... + \alpha_{nn}x_n. \end{cases} \quad (42)$$

Алгоритм Гауса-Зейделя

1. Вибрати значення координат вектора початкових наближень
$$\overline{x} = \left\{x_1^{(0)},...,x_n^{(0)}\right\}$$

2. Визначити значення першої координати $x_1^{(1)}$ вектора першого наближення з першого рівняння системи:
$$x_1^{(1)} = \beta_1 + \alpha_{11}x_1^{(0)} + \alpha_{12}x_2^{(0)} + ... + \alpha_{1n}x_n^{(0)}.$$

3. Підставити в друге рівняння системи значення першої координати $x_1^{(1)}$, яке обчислено на попередньому кроці
$$x_2^{(1)} = \beta_1 + \alpha_{21}x_1^{(1)} + \alpha_{22}x_2^{(0)} + ... + \alpha_{2n}x_n^{(0)}.$$

4. Отримані значення координат першого наближення $x_1^{(1)}$, $x_2^{(1)}$ підставляємо у третє рівняння системи (39)
$$x_3^{(1)} = \beta_1 + \alpha_{21}x_1^{(1)} + \alpha_{22}x_2^{(1)} + \alpha_{23}x_3^{(0)} + ... + \alpha_{2n}x_n^{(0)}$$
для знаходження третьої координати і т. д.

5. Для знаходження останньої координати вектора першого наближення $x_n^{(1)}$ в останнє рівняння системи треба підставити значення всіх $\left(n-1\right)$ координат ($x_1^{(1)}, x_2^{(1)}, x_{3,\ldots,}^{(1)} x_{n-1}^{(1)}$), які отримані на попередніх кроках, та значення координати $x_n^{(0)}$

$$x_n^{(1)} = \beta_n + \alpha_{n1}x_1^{(1)} + \alpha_{n2}x_2^{(1)} + ... + \alpha_{n,n-1}x_{n-1}^{(1)} + \alpha_{nn}x_n^{(0)}.$$

6. Аналогічно будують друге, третє та інші наближення. Так, для вектора $(k+1)$-го наближення за методом Зейделя використовують наступні формули:

$$\begin{cases} x_1^{(k+1)} = \beta_1 + \sum_{j=1}^{n} \alpha_{1j}x_j^{(k)}; \\ x_2^{(k+1)} = \beta_2 + \sum_{j=2}^{n} \alpha_{2j}x_j^{(k)} + \alpha_{21}x_j^{(k+1)}; \\ x_3^{(k+1)} = \beta_3 + \alpha_{31}x_1^{(k+1)} + \alpha_{32}x_2^{(k+1)} + \sum_{j=3}^{n} \alpha_{3j}x_j^{(k)}; \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ x_n^{(k+1)} = \beta_n + \sum_{j=1}^{n-1} \alpha_{nj}x_j^{(k+1)} + \alpha_{nn}x_n^{(k)}. \end{cases} \quad (43)$$

Умови збіжності ітераційного процесу Зейделя

Даний процес розв'язання СЛАР – ітераційний, тому важливим є аналіз умов збіжності ітераційного процесу. Процес Зейделя для системи лінійних рівнянь $\bar{x} = \bar{\beta} + \bar{\alpha}x$ збігається до точного розв'язку з заданою похибкою ε при будь-якому виборі вектора початкових наближень, якщо будь-яка норма матриці $\bar{\alpha}$ менша 1, тобто якщо:

$$\left\|\alpha\right\|_1 = \max \sum_{j=1}^{n} \left|\alpha_{ij}\right| < 1 \quad (44)$$

$$\left\|\alpha\right\|_2 = \max \sum_{i=1}^{n} \left|\alpha_{ij}\right| < 1 \quad (45)$$

$$\left\|\alpha\right\|_3 = \sqrt{\sum_i \sum_j \left|\alpha_{ij}\right|^2} < 1 \quad (46)$$

Відомо, що процес Зейделя сходиться до точного розв'язку СЛАР швидше, ніж метод послідовних наближень.

Приклад. Знайти першу та другу норми та проаналізувати умови збіжності ітераційного процесу для матриці α, яка має вигляд :

$$\bar{\alpha} = \begin{bmatrix} 0{,}24 & \text{-}0{,}05 & \text{-}0{,}24 \\ -022 & 0{,}09 & \text{-}0{,}44 \\ 0{,}13 & \text{-}0{,}02 & 0{,}42 \end{bmatrix}$$

$$\left\|\alpha\ \right\|_1 = \max_j \sum_{j=1}^{n} \left|\alpha_{ij}\right| = \max\left\{0{,}53;\ 0{,}75;\ 0{,}57\right\} = 0{,}75 < 1.$$

Очевидно, що процес ітерації для даної системи сходиться до точного розв'язку, незважаючи на те, що

$$\left\|\alpha\ \right\|_2 = \max_i \sum_{j=1}^{n} \ \left|\alpha_{ij}\right| = \max\{0{,}59;\ 0{,}16;\ 1{,}1\}=1{,}1>1.$$

Оцінка похибки методу Гауса-Зейделя

Якщо $\overline{x}$ – точне значення вектора розв'язку системи лінійних рівнянь; а $\overline{x}^{(k)}$- к-е наближення, обчислене за методом Гауса-Зейделя, то для оцінки похибки цього методу використовується формула:

$$\left\|\overline{x} - \overline{x}^{(k)}\right\|_1 \leq \frac{\left\|\alpha\right\|_1^{(k)}}{1-\left\|\alpha\right\|_1}\left\|\overline{x}^{(1)} - \overline{x}^{(0)}\right\|_1 \qquad (47)$$

Метод верхньої релаксації

В основі методу верхньої релаксації використується алгоритм та обчислювальна схема метода Гауса-Зейделя, але на відміну від нього нові значення координат вектора k -го наближення визначаються за формулами:

$$x_i^{(k+1)} = x_i^{(k)} + \varpi(\overline{x}_i^{(k+1)} - x_i^{(k)}), \qquad (48)$$

де $\overline{x}_i^{(k+1)}$- уточнене значення змінної по методу Гауса-Зейделя, ϖ - параметр релаксації, значення якого визначається з інтервалу $1 \leq \varpi \leq 2$. При ϖ=1 метод тотожний методу Гауса-Зейделя. Швидкість збіжності ітераційного процесу залежить від значення ϖ.

Контрольні запитання

1. Дати визначення термінів: розв'язання СЛАР, визначена і невизначена, вироджена, сумісна і несумісна СЛАР.
2. Класифікація методів розв'язання СЛАР.
3. Суть точних методів розв'язання СЛАР.
4. У яких випадках краще використовувати точні методи?
5. Суть алгоритму методу Гауса розв'язання СЛАР.
6. У чому різниця між алгоритмами методів Гауса з послідовним виключенням невідомих і з вибиранням головного елементу; що спільного?
7. У чому суть методу Гауса за схемою Халецького?
8. Порівняйте алгоритми методів Гауса з вибиранням головного елементу і за схемою Халецького. Обчислення за яким з цих алгоритмів швидше? Який із алгоритмів ефективніший?
9. В чому суть методу Гауса за схемою Халецького?
10. Яку систему отримано в результаті прямого ходу методу Гауса-Жордана?

11. Дати визначення термінів: розв'язання СЛАР, норма матриці, норма вектора, достатня умова збіжності.
12. Привести СЛАР:

$$9{,}9x_1 - 1{,}5x_2 + 2{,}6x_3 = 0;$$
$$0{,}4x_1 + 13{,}6x_2 - 4{,}2x_3 = 8{,}2;$$
$$0{,}7x_1 + 0{,}4x_2 + 7{,}1x_3 = -13;$$

до нормального вигляду.

Порядок виконання роботи

1. Ознайомитись з відомими точними чисельними методами розв'язання систем лінійних алгебраїчних рівнянь (СЛАР).
2. Скласти схему алгоритму розв'язання систем лінійних алгебраїчних рівнянь методом, вказаним викладачем.
3. Скласти програму розв'язання систем лінійних алгебраїчних рівнянь на алгоритмічній мові Pascal.
4. Користуючись даними відповідного варіанту (табл.1), розв'язати СЛАР на ЕОМ (початкові дані та результати роздрукувати).
5. Оформити звіт.

Звіт по лабораторній роботі повинен містити:
1) файл вихідного тексту програми;
2) файли результатів для тестового прикладу і для інтерполяції заданої функції;
3) опис алгоритму розрахунку (в текстовій формі та у вигляді блок-схеми) в електронному та роздрукованому вигляді;
4) роздруківку файлів з коментарями;
5) загальні висновки за результатами роботи, що включають результати тестування, отримані оцінки похибки результатів і обґрунтування цих оцінок.

Варіанти лабораторної роботи приведені в таблиці 1.

Таблиця 1 – Варіанти завдань

Номер варіанту	Матриця коефіцієнтів системи	Стовпець вільних членів	Примітка
1	$\begin{matrix} 1 & 1 & 2 \\ 2 & -1 & 2 \\ 4 & 1 & 4 \end{matrix}$	-1 -4 2	$x_1 = 1$ $x_2 = 2$ $x_3 = -2$
2	$\begin{matrix} 1 & -3 & 2 \\ 3 & -4 & 0 \\ 2 & -5 & 3 \end{matrix}$	1 2 2	$x_1 = 2$ $x_2 = 1$ $x_3 = 1$
3	$\begin{matrix} 1 & -3 & 2 \\ 3 & -4 & 0 \\ 2 & -5 & 3 \end{matrix}$	5 7 9	$x_1 = 5$ $x_2 = 2$ $x_3 = 3$

4	$\begin{matrix} 1 & 1 & 1 \\ 2 & 3 & 1 \\ 1 & -1 & -1 \end{matrix}$	4 9 -2	$x_1 = 1$ $x_2 = 2$ $x_3 = 1$
5	$\begin{matrix} 1 & -3 & 2 \\ 3 & -4 & 0 \\ 2 & -5 & 3 \end{matrix}$	-5 -2 -7	$x_1 = 6$ $x_2 = 5$ $x_3 = 2$
6	$\begin{matrix} 0,63 & 1,00 & 0,71 & 0,34 \\ 1,17 & 0,18 & -0,65 & 0,71 \\ 2,71 & -0,75 & 1,17 & -2,35 \\ 3,58 & 0,21 & -3,45 & -1,18 \end{matrix}$	2,08 0,17 1,28 0,05	$x_1 = 0.4026$ $x_2 = 1.5016$ $x_3 = 0.5862$ $x_4 = -0.2678$
7	$\begin{matrix} 7,09 & 1,17 & -2,23 \\ 0,43 & 1,40 & -0,62 \\ 3,21 & -4,25 & 2,13 \end{matrix}$	-4,75 -1,05 -5,06	$x_1 = 0.2386$ $x_2 = 0.5945$ $x_3 = 3.2019$
8	$\begin{matrix} 1,84 & 2,25 & 2,58 \\ 2,32 & 2,00 & 2,82 \\ 1,83 & 2,06 & 2,24 \end{matrix}$	-6,09 -6,96 -5,52	
9	$\begin{matrix} 2,36 & 2,37 & 2,13 \\ 2,51 & 2,40 & 2,10 \\ 2,59 & 2,41 & 2,06 \end{matrix}$	1,48 1,92 2,16	
10	$\begin{matrix} 6,1 & 0,7 & -0,05 \\ -1,3 & -2,05 & 0,87 \\ 2,5 & -3,12 & -5,03 \end{matrix}$	6,97 0,10 2,04	$x_1 = 1.22$ $x_2 = -0.67$ $x_3 = 0.35$
11	$\begin{matrix} 8,7 & -3,1 & 1,8 & 2,2 \\ 2,1 & 6,7 & -2,2 & 0 \\ 3,2 & -1,8 & -9,5 & -1,9 \\ 1,2 & 2,8 & -1,4 & -9,9 \end{matrix}$	-9,7 13,1 6,9 25,1	$x_1 = -0.72$ $x_2 = 1.88$ $x_3 = -0.92$ $x_4 = -1.94$
12	$\begin{matrix} 2,58 & 2,98 & 3,13 \\ 1,32 & 1,55 & 1,58 \\ 2,09 & 2,25 & 2,84 \end{matrix}$	-6,66 -3,58 -5,01	

13	$\begin{matrix} 1,54 & 1,70 & 1,62 \\ 3,69 & 3,73 & 3,59 \\ 2,45 & 2,43 & 2,25 \end{matrix}$	-1,97 -3,69 -5,98	
14	$\begin{matrix} 7,6 & 0,5 & 2,4 \\ 2,2 & 9,1 & 4,4 \\ -1,3 & 0,2 & 5,8 \end{matrix}$	1,9 9,7 -1,4	$x_1 = 0.248$ $x_2 = 1.114$ $x_3 = -0.224$
15	$\begin{matrix} 8 & 1 & 1 \\ 1 & 5 & -1 \\ 1 & -1 & 5 \end{matrix}$	26 7 7	$x_1 = 3$ $x_2 = 1$ $x_3 = 1$
16	$\begin{matrix} 2 & 1 & 4 \\ 2 & -1 & -3 \\ 3 & 4 & 5 \end{matrix}$	7 -5 -14	$x_1 = 0$ $x_2 = -1$ $x_3 = 2$
17	$\begin{matrix} 11 & 3 & -1 \\ 2 & 5 & -5 \\ 1 & 1 & 1 \end{matrix}$	15 -11 1	$x_1 = 2$ $x_2 = -2$ $x_3 = 1$
18	$\begin{matrix} 1 & -4 & 0 & -1 \\ 1 & 1 & 2 & 3 \\ 2 & 3 & -1 & -1 \\ 1 & 2 & 3 & -1 \end{matrix}$	6 -1 -1 3	$x_1 = 1$ $x_2 = -1$ $x_3 = 1$ $x_4 = -1$
19	$\begin{matrix} 2 & 0 & -1 & -2 \\ 0 & 1 & 2 & -1 \\ 1 & -1 & 0 & -1 \\ -1 & 3 & 2 & 0 \end{matrix}$	-8 -1 -6 7	$x_1 = -1$ $x_2 = 2$ $x_3 = 0$ $x_4 = 3$
20	$\begin{matrix} 1,14 & -2,15 & -5,11 \\ 0,42 & -1,13 & 7,05 \\ -0,71 & 0,81 & -0,02 \end{matrix}$	2,05 0,80 -1,07	$x_1 = 1.12$ $x_2 = -0.341$ $x_3 = -0.008$
21	$\begin{matrix} 0,61 & 0,71 & -0,05 \\ -1,03 & -2,05 & 0,87 \\ 2,5 & -3,12 & 5,03 \end{matrix}$	-0,16 0,50 0,95	$x_1 = 0.008$ $x_2 = -0.231$ $x_3 = 0.042$