



МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ  
ТА СПОРТУ УКРАЇНИ

УКРАЇНСЬКА ДЕРЖАВНА АКАДЕМІЯ  
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ

**МАТЕМАТИЧНІ МЕТОДИ І МОДЕЛІ:  
КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ**

*Підручник*

*Затверджено Міністерством освіти і науки, молоді та  
спорту України як підручник для студентів вищих  
навчальних закладів*

**Харків 2012**

УДК 004.942 (075.8)

ББК 22.12

М 541

*Затверджено Міністерством освіти і науки, молоді та спорту  
України як підручник для студентів вищих навчальних закладів  
(лист № 1/11 – 9813 від 18.06.2012 р).*

**Авторський колектив:** Завгородня Н.М., Панченко С.В.,  
Бантюков С.Є., Меркулов В.С.

**Рецензенти:**

професори Р.В. Вовк,  
Г.І. Загарій,  
В.А. Тихонов

М 541 Математичні методи і моделі: комп'ютерне  
моделювання: Підручник. – Харків: УкрДАЗТ, 2012. –  
185 с., табл. 5, рис. 52.  
ISBN 978-966-2033-83-0

Підручник містить теоретичні питання та математичний апарат, які лежать в основі курсу лекцій з дисциплін “Обчислювальна техніка, програмування, моделювання систем”, “Математичні моделі на ЕОМ”, “Математичне моделювання”, “Математичні методи та моделі в розрахунках на ЕОМ”, “Математичне моделювання БКВРМ”, “Числові методи та моделювання на ЕОМ”, що читаються в Українській державній академії залізничного транспорту.

У підручнику приділяється значна увага питанням методики викладання названих навчальних дисциплін й методам тестового контролю.

Книга призначена для студентів, бакалаврів і магістрів, слухачів факультету підвищення кваліфікації, викладачів, а також фахівців технічних галузей.

УДК 004.942 (075.8)

ББК 22.12

**ISBN 978-966-2033-83-0**

© Українська державна академія  
залізничного транспорту, 2012.  
© Завгородня Н.М., Панченко С.В.,  
Бантюков С.Є., Меркулов В.С., 2012

Підручник

**Завгородня Ніна Миколаївна,  
Панченко Сергій Володимирович,  
Бантюков Сергій Євгенович**  
та ін.

**МАТЕМАТИЧНІ МЕТОДИ І МОДЕЛІ:  
КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ**

Відповідальний за випуск Завгородня Н.М.

Редактор Ібрагімова Н.В.

---

Підписано до друку 25.12.2011 р.

Формат паперу 60x84 1/8. Папір писальний.

Умовн.-друк.арк. 4,0. Тираж 500. Замовлення № .

Видавець та виготовлювач Українська державна академія  
залізничного транспорту,

61050, Харків-50, майдан Фейербаха, 7.

Свідоцтво суб'єкта видавничої справи ДК № 2874 від 12.06.2007 р.

**МАТЕМАТИЧНІ МЕТОДИ І МОДЕЛІ:  
КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ**

*Підручник*

**Харків 2012**

УКРАЇНСЬКА ДЕРЖАВНА АКАДЕМІЯ  
ЗАЛІЗНИЧНОГО ТРАНСПОРТУ

Факультет автоматики, телемеханіки та зв'язку

Кафедра "Обчислювальна техніка та системи управління"

**Н.М. Завгородня, С.В. Панченко,  
С.Є. Бантюков, В.С. Меркулов**

**МАТЕМАТИЧНІ МЕТОДИ І МОДЕЛІ:  
КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ**

**Підручник**

**Харків - 2012**

УДК 004.942 (075.8)

Завгородня Н.М., Панченко С.В., Бантюков С.Є., Меркулов В.С.  
Математичні методи і моделі: комп'ютерне моделювання Підручник. -  
Харків: УкрДАЗТ, 2012 - 185 с.

Авторський колектив: Завгородня Ніна Миколаївна,  
Панченко Сергій Володимирович,  
Бантюков Сергій Євгенович,  
Меркулов Віктор Сергійович

Підручник містить теоретичні питання та математичний апарат, які лежать в основі курсу лекцій з дисциплін “Обчислювальна техніка, програмування, моделювання систем”, “Математичні моделі на ЕОМ”, “Математичне моделювання”, “Математичні методи та моделі в розрахунках на ЕОМ”, “Математичне моделювання БКВРМ”, “Числові методи та моделювання на ЕОМ”, що читаються в Українській державній академії залізничного транспорту.

У підручнику приділяється значна увага питанням методики викладання названих навчальних дисциплін й методам тестового контролю.

Книга призначена для студентів, бакалаврів і магістрів, слухачів факультету підвищення кваліфікації, викладачів, а також фахівців технічних галузей.

Іл. , табл. , бібліогр.: 9 назв.

ISBN 978-966-2033-830

Рецензенти:

професори Р.В.Вовк,  
Г.І. Загарій,  
В.А. Тихонов

Рекомендовано Міністерством освіти і науки України як підручник для студентів вищих навчальних закладів (лист № 1/11 – 9813 від 18.06.2012 р).

## ЗМІСТ

	Вступ .....	5
1.	Особливості побудови математичних моделей .....	10
2.	Комп'ютерне моделювання й обчислювальний експеримент .....	15
3.	Чисельні методи розв'язання нелінійних рівнянь ....	18
3.1.	Метод половинного ділення .....	21
3.2.	Метод простих ітерацій .....	24
3.3.	Метод Ньютона (метод дотичних) .....	26
3.4.	Модифікований метод Ньютона (метод січних) .....	29
3.5.	Метод хорд .....	29
4.	Комп'ютерне імітаційне моделювання. Статистичне імітаційне моделювання. Метод Монте-Карло .....	32
5.	Комп'ютерне моделювання і розв'язання лінійних багатовимірних систем .....	41
6.	Моделювання багатовимірних нелінійних систем ...	54
6.1.	Метод простих ітерацій .....	55
6.2.	Розв'язання систем нелінійних рівнянь методом Ньютона .....	59
7.	Комп'ютерне моделювання при обробці дослідних даних .....	66
7.1.	Інтерполяція функцій .....	67
7.2.	Побудова інтерполяційного многочлена в явному вигляді .....	70
7.3.	Інтерполяція за Лагранжем .....	71
7.4.	Програмування формули Лагранжа .....	72
7.5.	Інтерполяція за Ньютоном .....	74
7.6.	Програмування формули Ньютона .....	76
7.7.	Приклад інтерполяції за Ньютоном .....	78
7.8.	Сплайн-інтерполяція .....	80
7.9.	Апроксимація дослідних даних .....	82
8.	Чисельні методи інтегрування .....	90
8.1.	Сутність чисельних методів інтегрування .....	90
8.2.	Метод прямокутників .....	93
8.3.	Метод трапецій .....	96
8.4.	Метод Сімпсона .....	97

9.	Комп'ютерне моделювання динамічних систем .....	100
9.1.	Методи Рунге - Кутта .....	104
9.2.	Метод Рунге - Кутта 1-го порядку (метод Ейлера) ...	104
10.	Деякі відомості з елементарної теорії похибок .....	108
10.1.	Джерела й класифікація похибок результату чисельного розв'язання задач .....	108
10.2.	Наближені числа. Абсолютна й відносна похибки ...	109
	Бібліографічний список .....	125
	Додаток 1. Основні відомості лінійної алгебри .....	126
	Додаток 2. Тестові питання .....	132
	Додаток 3. Моделювання роботи пункту перестановки вагонів .....	169

## ВСТУП

Сьогодні практично немає такої сфери людської діяльності, де не застосовується обчислювальна техніка. ЕОМ зараз широко використовується в процесі створення й дослідження сучасних машин, нових технологічних процесів і пошуку їх оптимальних варіантів; при вирішенні економічних завдань, при вирішенні завдань планування й управління виробництвом на різних рівнях. Створення ж великих об'єктів у космічних дослідженнях, авіабудуванні, суднобудуванні, а також проектування гребель, мостів, будинків та ін. взагалі неможливо без застосування комп'ютерів.

Але для використання ЕОМ при розв'язанні прикладних задач насамперед треба перевести ці задачі на формальну математичну мову, тобто для реального об'єкта, процесу або системи побудувати відповідну *математичну модель*.

Слово "модель" походить від латинського *modus* (копія, образ, обрис). Моделювання - це заміщення деякого об'єкта А іншим об'єктом Б. Інакше кажучи, модель - це об'єкт-замінник об'єкта-оригіналу, що забезпечує вивчення деяких властивостей оригіналу.

Метою моделювання є одержання, обробка, представлення й використання інформації про об'єкти, які взаємодіють між собою з зовнішнім середовищем, а модель тут виступає як засіб пізнання властивостей і закономірностей поведінки об'єкта.

Моделювання широко використовуються в різних сферах людської діяльності, особливо у сферах проектування й управління, де особливими є процеси прийняття ефективних рішень на підставі одержуваної інформації.

Модель завжди будується з певною метою, яка впливає на те, які властивості об'єктивного явища виявляються істотними, а які - ні. Модель являє собою ніби проекцію об'єктивної реальності під певним кутом зору. Іноді, залежно від мети, можна отримати ряд проекцій об'єктивної реальності, що вступають у протиріччя. Це характерно, як правило, для складних систем, у яких кожна проекція виділяє істотне для певної мети з безлічі несуттєвого.



Всі моделі можна поділити на два класи:

- 1) дійсні;
- 2) ідеальні.

У свою чергу дійсні моделі можна поділити:

- 1) на натурні;
- 2) фізичні;
- 3) математичні.

Ідеальні моделі можна поділити:

- 1) на наочні;
- 2) знакові;
- 3) математичні.

Дійсні натурні моделі - це реальні об'єкти, процеси й системи, над якими виконуються експерименти наукові, технічні й виробничі.

Дійсні фізичні моделі - це макети, муляжі, що відтворюють фізичні властивості оригіналів (кінематичні, динамічні, гідравлічні, теплові, електричні, світлові моделі).

Дійсні математичні - це аналогові, структурні, геометричні, графічні, цифрові й кібернетичні моделі.

Ідеальні наочні моделі - це схеми, карти, креслення, графіки, графи, аналоги, структурні й геометричні моделі.

Ідеальні знакові моделі - це символи, алфавіт, мови програмування, упорядкований запис, топологічний запис.

Ідеальні математичні моделі - це аналітичні, функціональні, імітаційні, комбіновані моделі.

Зупинимося на одному з найбільш універсальних видів моделювання - математичному. Воно ставить у відповідність фізичному процесу, що моделюється, систему математичних співвідношень, розв'язання яких дозволяє отримати відповідь на запитання про поведінку об'єкта без створення фізичної моделі, яка часто виявляється дорогою й неефективною.

*Математичне моделювання* - це засіб вивчення реального об'єкта, процесу або системи шляхом їхньої заміни *математичною моделлю*, більш зручною для експериментального дослідження за допомогою ЕОМ.

*Математична модель* є наближеним зображенням реальних об'єктів, процесів або систем, вираженим у математичних термінах, що зберігають суттєві риси оригіналу.

*Математичні моделі* в кількісній формі, за допомогою логіко-математичних конструкцій, описують основні властивості об'єкта, процесу або системи, його параметри, внутрішні й зовнішні зв'язки.

У загальному випадку *математична модель* реального об'єкта, процесу або системи може бути подана у вигляді системи функціоналів

$$\Phi_i (X, Y, Z, t) = 0,$$

де  $X$  - вектор вхідних змінних,  $X = [x_1, x_2, x_3, \dots, x_n]$ ;

$Y$  - вектор вихідних змінних,  $Y = [y_1, y_2, y_3, \dots, y_n]$ ;

$Z$  - вектор зовнішніх впливів,  $Z = [z_1, z_2, z_3, \dots, z_n]$ ;

$t$  - координата часу.

Побудова *математичної моделі* полягає у визначенні зв'язків між тими або іншими процесами і явищами, створенні математичного апарата, що дозволяє виразити кількісно і якісно зв'язок між тими або іншими процесами і явищами, між фізичними величинами, що цікавлять фахівця, і факторами, що впливають на кінцевий результат. Звичайно, їх виявляється настільки багато, що ввести в модель усю їхню сукупність не вдається. При побудові *математичної моделі* необхідно виявити й виключити з розгляду фактори, що несуттєво впливають на кінцевий результат (*математична модель* звичайно включає значно менше число факторів, ніж в реальній дійсності). На підставі даних експерименту висуваються гіпотези про зв'язок між величинами, що виражають кінцевий результат, і факторами, які були введені в *математичну модель*.

Форма й принципи подання *математичної моделі* залежить від багатьох факторів.

За принципами побудови *математичні моделі* поділяють:

- 1) на аналітичні;
- 2) імітаційні.

В аналітичних моделях процеси функціонування реальних об'єктів, процесів або систем записуються у вигляді явних функціональних залежностей.

Аналітичні моделі поділяються на типи залежно від математичної проблеми:

1. Рівняння (алгебраїчні, трансцендентні, диференціальні, інтегральні).
2. Апроксимаційні задачі (інтерполяція, екстраполяція, чисельне інтегрування й диференціювання).
3. Задачі оптимізації.
4. Стохастичні задачі.

З ускладненням об'єкта моделювання побудова аналітичної моделі перетворюється у складну проблему. Тоді дослідник змушений використати *імітаційне моделювання*.

Імітаційне моделювання дозволяє за вихідними даними отримати відомості про стан процесу або системи в певні моменти часу, однак прогнозування поведінку об'єктів, процесів або систем тут важко. Можна сказати, що імітаційні моделі - це проведені на ЕОМ обчислювальні експерименти з *математичними моделями*, що імітують поведінку реальних об'єктів, процесів або систем.

Залежно від характеру досліджуваних реальних процесів і систем *математичні моделі* можуть бути:

- 1) детерміновані;
- 2) стохастичні.

У детермінованих моделях передбачається відсутність будь-яких випадкових впливів, елементи моделі (змінні, математичні зв'язки) достатньо точно встановлені, поведінку системи можна точно визначити. При побудові детермінованих

моделей найчастіше використовуються алгебраїчні рівняння, інтегральні рівняння, матрична алгебра.

Стохастична модель ураховує випадковий характер процесів у досліджуваних об'єктах і системах, який описується методами теорії імовірності й математичної статистики.

За типом вхідної інформації моделі поділяються:

- 1) на безперервні;
- 2) дискретні.

Якщо інформація й параметри є безперервними, а математичні зв'язки стійкими, то модель - безперервна. І навпаки, якщо інформація й параметри - дискретні, а зв'язки нестійкі, то й *математична модель* - дискретна.

За поведінкою моделей у часі вони поділяються:

- 1) на статичні;
- 2) динамічні.

Статичні моделі описують поведінку об'єкта, процесу або системи в який-небудь момент часу.

Динамічні моделі відображають поведінку об'єкта, процесу або системи в часі.

За ступенем відповідності між *математичною моделлю* й реальним об'єктом, процесом або системою *математичні моделі* поділяються:

- 1) на ізоморфні (однакові за формою);
- 2) гомоморфні (різні за формою).

Модель називається ізоморфною, якщо між нею й реальним об'єктом, процесом або системою існує повна поелементна відповідність; гомоморфною - якщо існує відповідність лише між найбільш значними складовими частинами об'єкта й моделі.

# 1. ОСОБЛИВОСТІ ПОБУДОВИ МАТЕМАТИЧНИХ МОДЕЛЕЙ

Для використання ЕОМ при розв'язанні прикладних задач насамперед задача повинна бути "перекладена" на формальну математичну мову, тобто для реального об'єкта, процесу або системи повинна бути побудована його математична модель.

Для *побудови математичної моделі* необхідно:

- 1) ретельно проаналізувати реальний об'єкт або процес;
- 2) виділити його найбільш істотні риси й властивості;
- 3) визначити змінні, тобто параметри, значення яких впливають на основні риси й властивості об'єкта;
- 4) описати залежність основних властивостей об'єкта, процесу або системи від значення змінних за допомогою логіко-математичних співвідношень (рівняння, рівності, нерівності, логіко-математичні конструкції);
- 5) виділити внутрішні зв'язки об'єкта, процесу або системи за допомогою обмежень, рівнянь, рівностей, нерівностей, логіко-математичних конструкцій;
- 6) визначити зовнішні зв'язки й описати їх за допомогою обмежень, рівнянь, рівностей, нерівностей, логіко-математичних конструкцій.

Математичне моделювання, крім дослідження об'єкта, процесу або системи й складання їх математичного опису, також включає:

- побудову алгоритму, що моделює поведінку об'єкта, процесу або системи;
- перевірку адекватності моделі й об'єкта, процесу або системи на підставі обчислювального й натурного експерименту;
- коригування моделі;
- використання моделі.

Математичний опис досліджуваних процесів і систем залежить:

- від природи реального процесу або системи й складається на основі законів фізики, хімії, механіки, термодинаміки, гідродинаміки, електротехніки, теорії пластичності, теорії пружності і т. д.;
- необхідної вірогідності й точності вивчення й дослідження реальних процесів і систем.

На етапі вибору математичної моделі встановлюються лінійність і нелінійність об'єкта, процесу або системи, динамічність або статичність, стаціонарність або нестаціонарність, а також ступінь детермінованості досліджуваного об'єкта або процесу. При математичному моделюванні свідомо абстрагуються від конкретної фізичної природи об'єктів, процесів або систем і в основному зосереджуються на вивченні кількісних залежностей між величинами, що описують ці процеси.

Математична модель ніколи не буває цілком тотожна розглянутому об'єкту, процесу або системі. Заснована на спрощенні, ідеалізації, вона є наближеним описом об'єкта. Тому результати, отримані при аналізі моделі, носять наближений характер. Їхня точність визначається ступенем адекватності (відповідності) моделі й об'єкта.

*Побудова математичної моделі* звичайно починається з побудови й аналізу найпростішої, найбільш грубої математичної моделі розглянутого об'єкта, процесу або системи. Надалі, якщо буде потреба, модель уточнюється, робиться її відповідність об'єкту більш повною. Основні відомості лінійної алгебри, які необхідні для побудови моделі надано в дод. 1.

Розглянемо простий приклад. Потрібно визначити площу поверхні письмового стола. Звичайно для цього вимірюють його довжину й ширину, а потім перемножують отримані числа. Така елементарна процедура фактично означає таке: реальний об'єкт (поверхня стола) замінюється абстрактною математичною моделлю - прямокутником. Прямокутнику приписуються розміри, отримані в результаті виміру довжини й ширини поверхні стола, і площа такого прямокутника приблизно приймається за шукану площу стола.

Однак модель прямокутника для письмового стола - це найпростіша, найбільш груба модель. При більш серйозному підході до завдання перш, ніж скористатися для визначення площі стола моделлю прямокутника, цю модель потрібно перевірити. Перевірки можна здійснити в такий спосіб: виміряти довжини протилежних сторін стола, а також довжини його діагоналей і зрівняти їх між собою. Якщо з необхідним ступенем точності довжини протилежних сторін і довжини діагоналей попарно рівні між собою, то поверхня стола дійсно можна розглядати як прямокутник. А якщо ні, то модель прямокутника доведеться відкинути й замінити моделлю чотирикутника загального вигляду. При більш високій вимозі до точності може виникнути необхідність у подальшому уточненні моделі, наприклад урахувати закруглення кутів стола і т. д.

За допомогою цього простого прикладу можна бачити, що математична модель не визначається однозначно досліджуваним об'єктом, процесом або системою. Для того самого стола ми можемо прийняти або модель прямокутника, або більш складну модель чотирикутника загального вигляду, або чотирикутника з закругленими кутами. Вибір тієї або іншої моделі визначається вимогою точності. З підвищенням точності модель доводиться ускладнювати, враховуючи нові й нові особливості досліджуваного об'єкта, процесу або системи.

Розглянемо інший приклад: дослідження руху кривошипно-шатунного механізму (рис. 1.1).

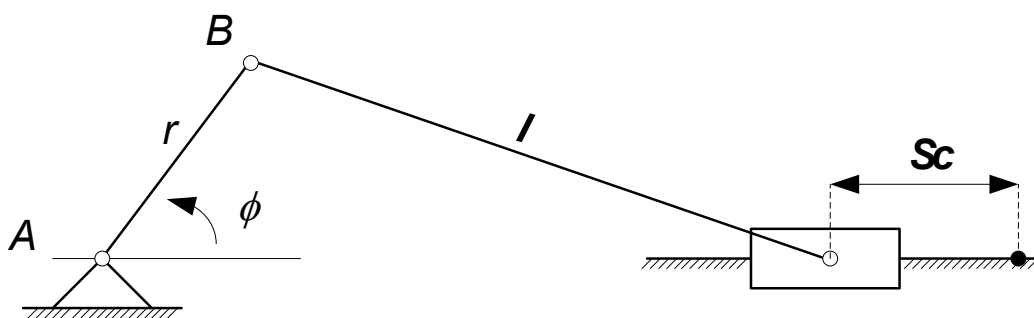


Рис. 1.1

На початку побудуємо кінематичну модель цього механізму.  
Для цього:

- 1) замінюємо механізм його кінематичною схемою, де усі ланки замінені жорсткими зв'язками;
- 2) користуючись цією схемою, виводимо рівняння руху механізму;
- 3) диференціюючи останнє, отримуємо рівняння швидкостей і прискорення, які являють собою диференціальні рівняння 1-го й 2-го порядку.

Запишемо ці рівняння:

$$\begin{cases} S_c = r(1 - \cos \varphi + \frac{\lambda}{2} \sin^2 \varphi); \\ V_c = r(\sin \varphi + \frac{\lambda}{2} \sin 2\varphi); \\ A_c = r(\cos \varphi + \lambda \sin 2\varphi), \end{cases}$$

де  $S_c$  - крайнє праве положення повзуна С;

$r$  - радіус кривошипа АВ;

$l$  - довжина шатуна ВС;

$\varphi$  - кут повороту кривошипа;

$\lambda = r/l$ .

Виходячи з цього можна одержати систему трансцендентних рівнянь, яка являє собою математичну модель руху плоского аксіального кривошипно-шатунного механізму, засновану на таких спрощуючих припущеннях:

- всі тіла механізму ми замінили відрізками прямих. Насправді усі ланки механізму мають масу й досить складну форму, які, звичайно, будуть впливати на рух механізму;

- ми не враховували пружність тіл, що є складовими механізму, тобто всі ланки розглядали як абстрактні абсолютно тверді тіла. У дійсності ж усі тіла, що складають механізм, -



пружні тіла. Вони при русі механізму будуть якось деформуватися, у них можуть навіть виникнути пружні коливання. Це все також буде впливати на рух механізму;

- ми не враховували похибку виготовлення ланок, зазори в кінематичних парах А, В, С і т. д.

Таким чином, важливо ще раз підкреслити, що чим вище вимоги до точності результатів розв'язання задачі, тим більше необхідність враховувати при *побудові математичної моделі* особливості об'єкта, процесу або системи, які досліджуються. При цьому важливо стежити за тим, щоб складна математична модель не перетворилася у задачу, розв'язання якої буде дуже складним.

Найпростіше будується модель, коли добре відомі закони, що визначають поведінку й властивості об'єкта, процесу або системи, і є великий практичний досвід їх застосування.

Більш складна ситуація виникає тоді, коли наші знання про досліджуваний об'єкт, процес або систему недостатні. У цьому випадку при *побудові математичної моделі* доводиться робити додаткові припущення, які носять характер гіпотез, така модель називається гіпотетичною. Висновки, отримані в результаті дослідження такої гіпотетичної моделі, носять умовний характер.

Основним критерієм істинності такої моделі є експеримент, практика в найширшому значенні цього слова.

*Побудова математичної моделі* в прикладних задачах - один з найбільш складних і відповідальних етапів роботи. Досвід показує, що в багатьох випадках правильно вибрати модель - означає вирішити проблему більш ніж наполовину. Труднощі даного етапу полягають у тому, що він вимагає з'єднання математичних і спеціальних знань. Саме тому дуже важливо, щоб фахівці-інженери володіли певною математичною культурою, досвідом проведення досліджень у своїй галузі, знанням ЕОМ і програмування.

З метою здійснення самоперевірки, перевірки знань студентів викладачами передбачено тестові питання (дод. 2).

## 2. КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ Й ОБЧИСЛЮВАЛЬНИЙ ЕКСПЕРИМЕНТ

Комп'ютерне моделювання як метод наукових досліджень ґрунтується:

- 1) на побудові математичних моделей для опису досліджуваних процесів;
- 2) використанні новітніх обчислювальних машин, які мають високу швидкість і здатні вести діалог з людиною.

Суть комп'ютерного моделювання полягає в такому: на підставі математичної моделі за допомогою ЕОМ проводиться серія обчислювальних експериментів, тобто досліджуються властивості об'єктів або процесів, знаходяться їхні оптимальні параметри й режими роботи, уточнюється модель. Наприклад, якщо ми маємо рівняння, що описують протікання того або іншого процесу, можна, змінюючи його коефіцієнти, початкові й граничні умови, дослідити, як при цьому буде поводитися об'єкт. Більш того, можна спрогнозувати поведінку об'єкта в різних умовах.

Обчислювальний експеримент дозволяє замінити дорогий натурний експеримент розрахунками на ЕОМ. Він дозволяє в короткий термін і без значних матеріальних витрат здійснити дослідження великої кількості варіантів проєктованого об'єкта або процесу для різних режимів його експлуатації, що значно скорочує терміни розроблення складних систем і їх впровадження у виробництво.

У деяких процесах, де натурний експеримент небезпечний для життя й здоров'я людей, обчислювальний експеримент є єдиною можливістю (термоядерний синтез, освоєння космічного простору, проєктування й дослідження хімічних та інших виробництв).

Для перевірки адекватності математичної моделі реальному об'єкту, процесу або системі результати досліджень на ЕОМ порівнюються з результатами експерименту на дослідному натурному зразку. Результати перевірки використовуються для коректування математичної моделі або вирішення

питання про застосування побудованої математичної моделі до проектування або дослідження заданих об'єктів, процесів чи систем.

У задачах проектування або дослідження поведінки реальних об'єктів, процесів або систем математичні моделі, як правило, нелінійні, тому що вони повинні відображати реальні фізичні нелінійні процеси, які протікають у них. При цьому параметри (змінні) цих процесів пов'язані між собою фізичними нелінійними законами. Саме тому в задачах проектування або дослідження поведінки реальних об'єктів, процесів чи систем зазвичай використовуються детерміновані безперервні аналітичні математичні моделі.

Після побудови математичної моделі розглянутого об'єкта, процесу або системи, тобто подання прикладної задачі як математичної, важливим є пошук або розроблення методу розв'язання сформульованої математичної задачі. Метод повинен бути зручним для його реалізації на ЕОМ, забезпечувати необхідну якість розв'язання.

Всі методи розв'язання математичних задач можна поділити на 2 групи:

- 1) точні методи розв'язання задач;
- 2) чисельні методи розв'язання задач.

У точних методах розв'язання математичних задач відповідь вдається отримати у вигляді формул. Наприклад, обчислення кореня квадратного рівняння, обчислення похідних табличних функцій або, у ряді випадків, обчислення визначеного інтегралу.

Однак, підставляючи числа у формулу у вигляді кінцевих десяткових дробів, ми все одно одержуємо наближені значення результату за рахунок наближеного подання чисел в ЕОМ.

Для більшості задач, що зустрічаються на практиці, точні методи розв'язання або невідомі, або дають дуже громіздкі формули. Однак вони не завжди є необхідними. Прикладну задачу можна вважати практично розв'язаною, якщо ми зуміємо її розв'язати з потрібним ступенем точності.

Для розв'язання таких задач розроблено чисельні методи, у яких розв'язання складних математичних задач зводиться до послідовного виконання великого числа простих арифметичних операцій. Безпосереднє розроблення чисельних методів належить до обчислювальної математики.

Зазначимо, що в прикладних задачах і при застосуванні точних методів розв'язання, і при застосуванні чисельних методів розв'язання відповідних задач результати обчислень носять наближений характер. Важливо тільки домогтися того, щоб помилки уклалися в рамки необхідної точності.

Чисельні методи розв'язання математичних задач відомі давно, ще до появи ЕОМ, але ними користувалися рідко й тільки в порівняно простих випадках, тому що такі обчислення є надзвичайно трудомісткими. Широке застосування чисельних методів стало можливим тільки завдяки ЕОМ.

### 3. ЧИСЕЛЬНІ МЕТОДИ РОЗВ'ЯЗАННЯ НЕЛІНІЙНИХ РІВНЯНЬ

Якщо закони функціонування моделі нелінійні, а процес або система що моделюються, мають один ступінь свободи (тобто мають одну незалежну змінну), то така модель, як правило, описується одним нелінійним рівнянням.

Необхідність відшукування коренів нелінійних рівнянь зустрічається в розрахунках систем автоматичного управління і регулювання, власних коливань машин і конструкцій, у задачах кінематичного аналізу і синтезу, плоских і просторових механізмів та інших задачах.

Дано нелінійне рівняння

$$f(x) = 0. \quad (3.1)$$

Необхідно розв'язати це рівняння, тобто знайти його корінь  $\bar{x}$  (рис. 3.1).

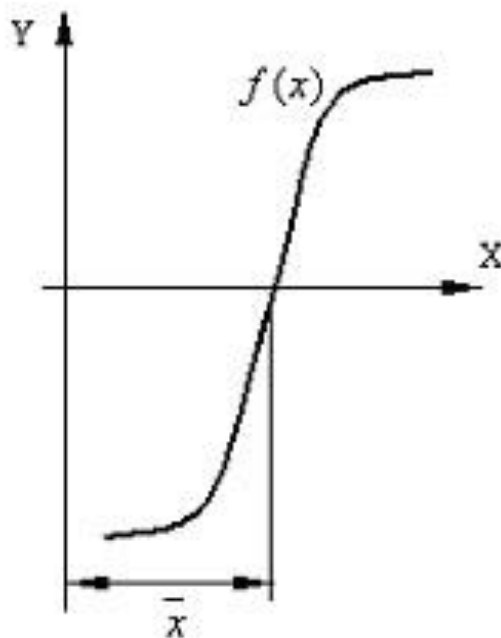


Рис. 3.1

Якщо функція має вигляд многочлена степеня  $m$

$$f(x) = a_0x^m + a_1x^{m-1} + a_2x^{m-2} + \dots + a_{m-1}x + a_m,$$

де  $a_i$  - коефіцієнти многочлена,  $i = \overline{1, m}$ ,

то рівняння  $f(x)=0$  має  $m$  коренів (рис. 3.2).

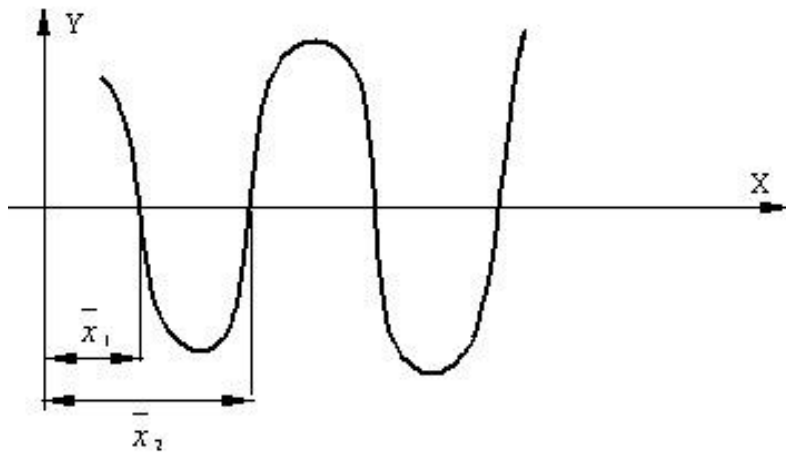


Рис. 3.2

Якщо функція  $f(x)$  містить тригонометричні або експонентні функції від деякого аргументу  $x$ , то рівняння (3.1) називається *трансцендентним*.

Приклади:

$$\arctg x + 1 - x = 0,$$

$$x - e^{\frac{\sin(-x)}{10}} = 0.$$

Такі рівняння звичайно мають нескінченну безліч розв'язків.

Як відомо, не кожне рівняння може бути розв'язано точно. У першу чергу ця властивість стосується більшості *трансцендентних рівнянь*.

Однак точний розв'язок рівняння не завжди необхідний. Задачу відшукування коренів рівняння можна вважати практично розв'язаною, якщо ми зуміємо знайти корінь рівняння з заданим ступенем точності. Для цього використовуються наближені (чисельні) методи розв'язання.

Процес визначення коренів алгебраїчних і *трансцендентних рівнянь* складається з 2 етапів:

1. *Відділення коренів*, тобто визначення для кожного кореня інтервалу ізоляції  $[a,b]$ , всередині якого лежить корінь рівняння.

2. *Уточнення коренів*, тобто звуження інтервалу  $[a,b]$  до розмірів заданого ступеня точності  $\epsilon$ .

Більшість наближених методів, що використовуються для розв'язання рівнянь, є, по суті, способами уточнення кореня. Для їхнього застосування необхідне знання інтервалу ізоляції  $[a,b]$ , у якому лежить корінь, що уточнюється (рис. 3.3).

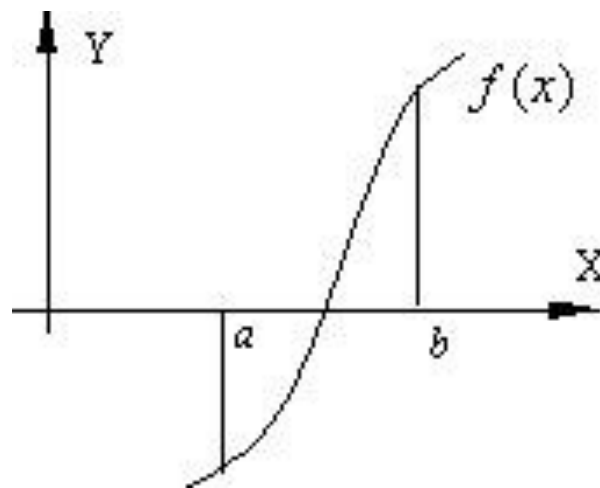


Рис. 3.3

Процес визначення інтервалу ізоляції  $[a,b]$ , що містить тільки один корінь рівняння, називається відділенням цього кореня.

Процес відділення кореня здійснюють виходячи з фізичного смислу прикладної задачі, графічно, за допомогою таблиць значень функції  $f(x)$  або за допомогою спеціальної програми відділення коренів. Процедура відділення коренів базується на відомій властивості безперервних функцій: якщо функція безперервна на замкнутому інтервалі  $[a,b]$  і на його кінцях має різні знаки, тобто  $f(a)*f(b)<0$ , то між точками  $a$  і  $b$  є хоча б один корінь рівняння (3.1). Якщо при цьому знак функції  $f'(x)$  на відрізку  $[a,b]$  не змінюється, то корінь є єдиним на цьому відрізку.

Для алгебраїчних і трансцендентних рівнянь використовують такі методи уточнення наближених значень дійсного кореня:

- 1) *метод половинного ділення (метод дихотомії);*
- 2) *метод простих ітерацій;*
- 3) *метод Ньютона (метод дотичних);*
- 4) *модифікований метод Ньютона (метод січних);*
- 5) *метод хорд та ін.*

### 3.1. Метод половинного ділення

Дане нелінійне рівняння (3.1):

$$f(x) = 0$$

Знайти корінь рівняння, що належить інтервалу  $[a,b]$ , із заданою точністю  $\varepsilon$ .

Для уточнення кореня методом половинного ділення послідовно здійснюємо такі операції:

1. Ділимо інтервал навпіл:

$$t = \frac{a + b}{2} \text{ - координата середини відрізка } [a,b].$$



2. У якості нового інтервалу ізоляції приймаємо ту половину інтервалу, на кінцях якої функція має різні знаки (рис. 3.4).

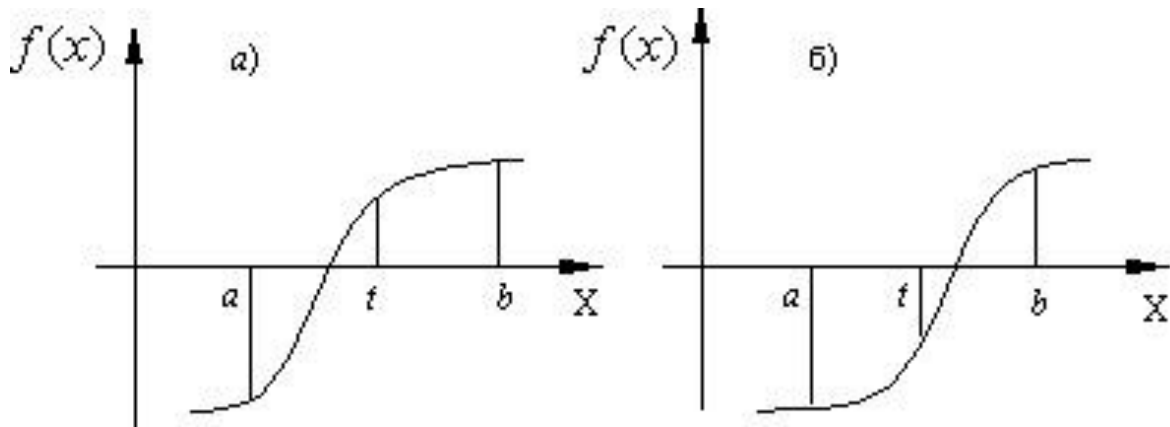


Рис. 3.4

Для цього:

а) обчислюємо значення функції  $f(x)$  у точках "a" і "t";

б) перевіряємо: якщо  $f(a)*f(t) < 0$ , то корінь знаходиться в лівій половині інтервалу  $[a,b]$  (рис. 3.4, а). Тоді відкидаємо праву половину інтервалу й робимо переприсвоєння  $b=t$ ;

в) якщо  $f(a)*f(t) < 0$  не виконується, то корінь знаходиться в правій половині інтервалу  $[a,b]$  (рис. 3.4, б). Тоді відкидаємо ліву половину й робимо переприсвоєння  $a=t$ . В обох випадках ми отримаємо новий інтервал  $[a,b]$ , який в 2 рази менший від попереднього.

3. Процес, починаючи з пункту 1, циклічно повторюємо доти, поки довжина інтервалу  $[a,b]$  не стане рівною або меншою, ніж задана точність, тобто  $|b - a| \leq \varepsilon$ .

Схема алгоритму уточнення коренів за методом половинного ділення зображена на рис. 3.5.

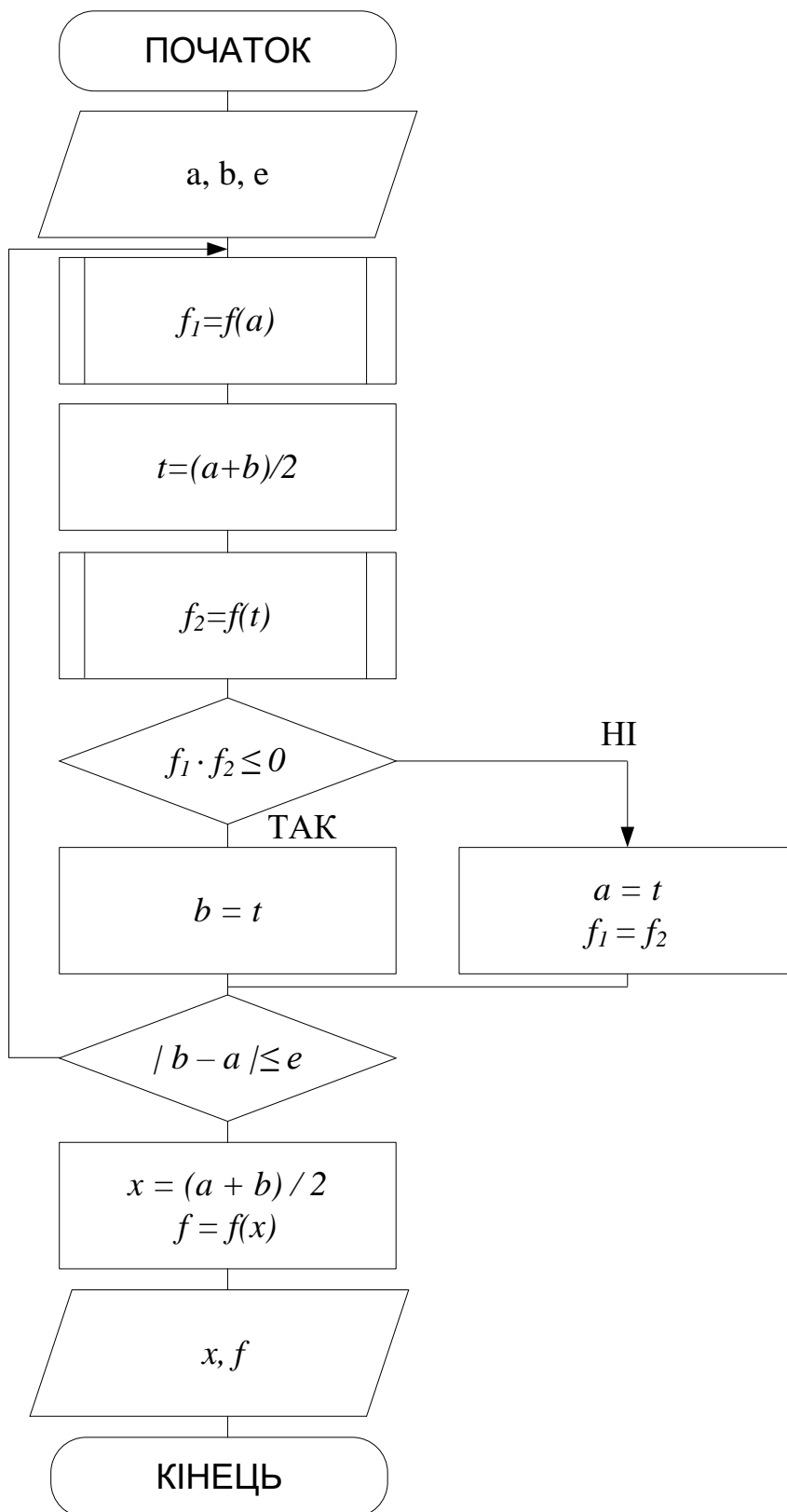


Рис. 3.5. Схема алгоритму уточнення коренів за методом половинного ділення

### 3.2. Метод простих ітерацій

У ряді випадків дуже зручним прийомом уточнення кореня рівняння є метод послідовних наближень (метод ітерацій).

Нехай з точністю  $\varepsilon$  необхідно знайти корінь рівняння  $f(x)=0$ , що належить до інтервалу ізоляції  $[a,b]$ . Функція  $f(x)$  і її перша похідна безперервні на цьому відрізку.

Для застосування цього методу вихідне рівняння  $f(x)=0$  повинно бути приведено до вигляду

$$x = \varphi(x) \quad (3.2)$$

У якості початкового наближення  $x_0$  вибираємо будь-яку точку інтервалу  $[a,b]$ .

Далі ітераційний процес пошуку кореня будується за схемою:

$$\begin{aligned} x_1 &= \varphi(x_0); \\ x_2 &= \varphi(x_1); \\ &\dots\dots\dots \\ x_n &= \varphi(x_{n-1}). \end{aligned} \quad (3.3)$$

Геометричний зміст методу простих ітерацій наведено на рис. 3.6.

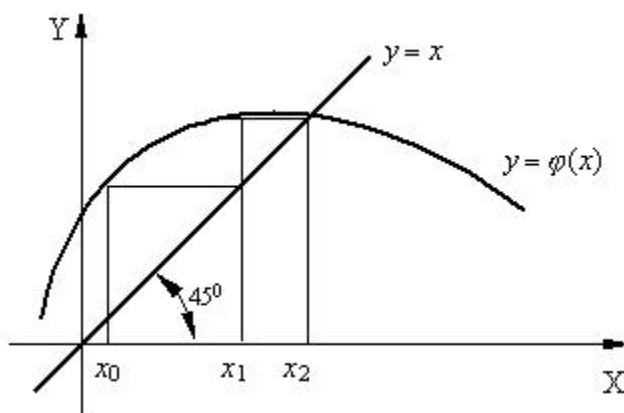


Рис. 3.6. Геометричний зміст методу простих ітерацій

У результаті ітераційний процес пошуку реалізується за рекурентною формулою (3.3). Процес пошуку припиняється, як тільки виконується умова

$$|x_n - x_{n-1}| \leq \varepsilon \quad (3.4)$$

або число ітерацій перевищить задане число  $N$ .

Для того щоб послідовність  $x_1, x_2, \dots, x_n$  наближалася до кореня, пошук якого відбувається, необхідно виконання умови збіжності:

$$|\varphi'(x)| < 1. \quad (3.5)$$

Переходимо до побудови схеми алгоритму (рис. 3.7). Обчислення функції  $\varphi(x)$  оформимо у вигляді підпрограми.

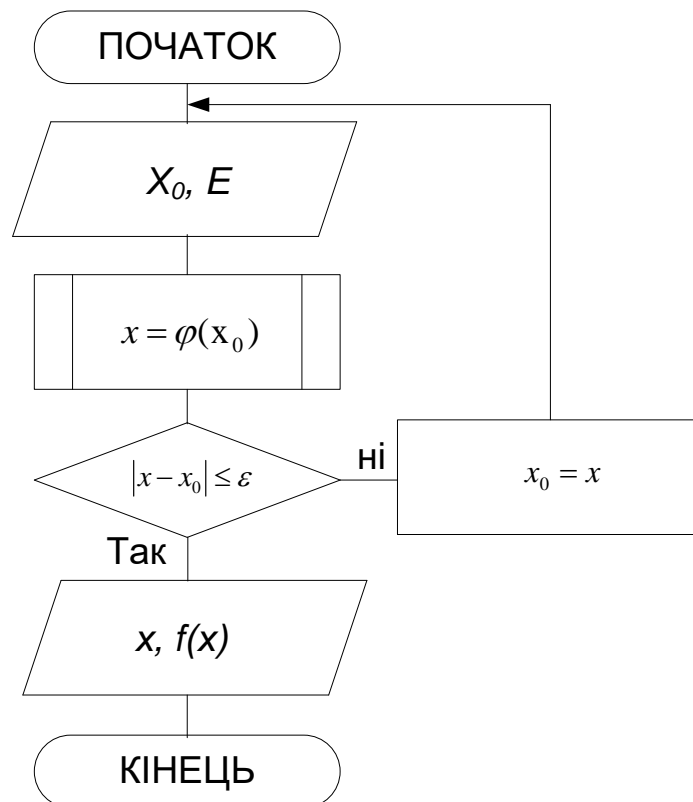


Рис. 3.7. Схема алгоритму уточнення кореня методом ітерацій

### 3.3. Метод Ньютона (метод дотичних)

Розглянуті раніше методи розв'язання нелінійних рівнянь є методами прямого пошуку. У них для знаходження кореня використовується знаходження значення функції в різних точках інтервалу  $[a,b]$ .

*Метод Ньютона* належить до градієнтних методів, у яких для знаходження кореня використовується значення похідної.

Дане нелінійне рівняння (3.1).

Знайти корінь на інтервалі  $[a,b]$  з точністю  $\varepsilon$ .

*Метод Ньютона* заснований на заміні похідної функції  $f(x)$  на кожному кроці пошуку дотичною, проведеною до цієї функції. Перетинання дотичної з віссю  $X$  дає наближення кореня (рис. 3.8).

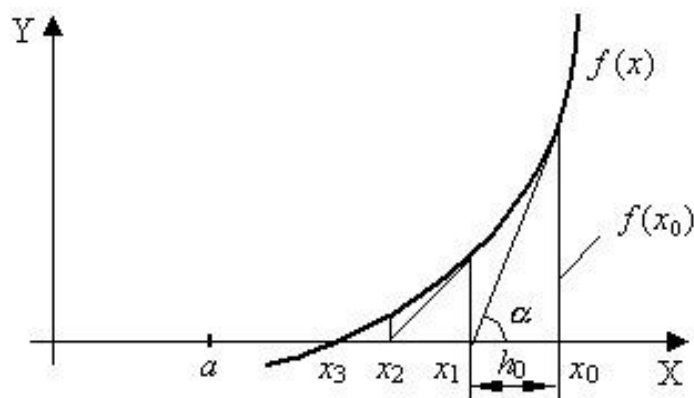


Рис. 3.8.

Виберемо початкову точку  $x_0=b$  (кінець інтервалу ізоляції). Знаходимо значення функції в цій точці і проводимо до неї дотичну, перетинання якої з віссю  $X$  дає нам перше наближення до кореня  $x_1$ .

$$x_1 = x_0 - h_0,$$

де 
$$h_0 = \frac{f(x_0)}{\operatorname{tg}(\alpha)} = \frac{f(x_0)}{f'(x_0)}.$$

Тому

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

У результаті ітераційний процес наближення до кореня реалізується рекурентною формулою

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (3.6)$$

Процес пошуку продовжуємо доти, поки не виконається умова

$$|x_{n+1} - x_n| \leq \varepsilon. \quad (3.7)$$

З урахуванням формули (3.6), одержимо

$$\left| \frac{f(x_n)}{f'(x_n)} \right| \leq \varepsilon \quad (3.8)$$

Метод забезпечує швидку збіжність, якщо виконується умова

$$f(x_0) \cdot f''(x_0) > 0, \quad (3.9)$$

таким чином першу дотичну рекомендується проводити в тій точці інтервалу  $[a,b]$ , де знаки функції  $f(x_0)$  і її кривизни  $f''(x_0)$  збігаються.

Схема алгоритму уточнення кореня *методом Ньютона* приведена на рис. 3.9.

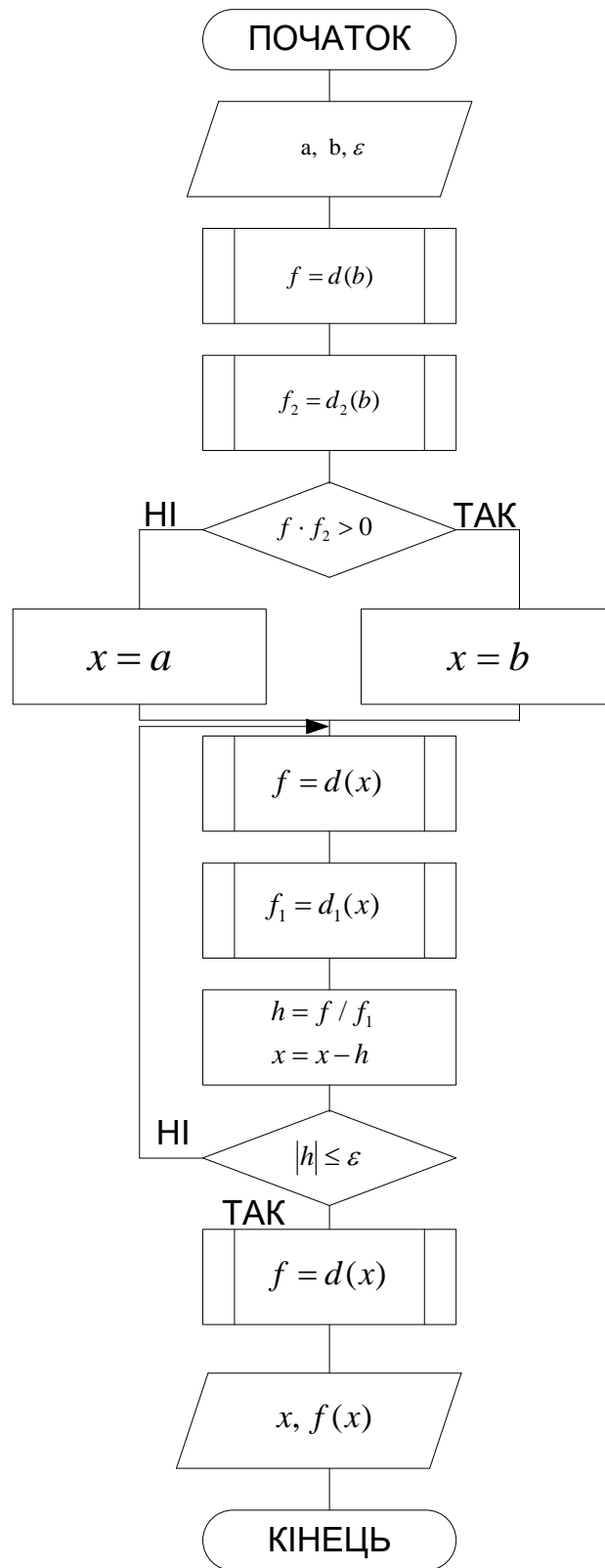


Рис. 3.9. Схема алгоритму уточнення кореня методом Ньютона

### 3.4. Модифікований метод Ньютона (метод січних)

У цьому методі для обчислення похідних на кожному кроці пошуку використовується чисельне диференціювання за формулою

$$f'(x) = \frac{\Delta f(x)}{\Delta x}.$$

Рекурентна формула (3.6) буде мати вигляд:

$$x_{n+1} = x_n - \frac{f(x)}{f'(x)} = x_n - \frac{f(x)\Delta x}{\Delta f(x)} = x_n - \frac{f(x)\Delta x}{f(x + \Delta x) - f(x)}, \quad (3.10)$$

де  $\Delta x \approx \varepsilon$ .

### 3.5. Метод хорд

Метод базується на заміні функції  $f(x)$  на кожному кроці пошуку хордою, перетинання якої з віссю  $X$  дає наближення кореня.

При цьому в процесі пошуку сімейство хорд може будуватися:

а) при фіксованому лівому кінці хорд, тобто  $z=a$ , тоді початкова точка  $x_0=b$  (рис. 3.10, а);

б) при фіксованому правому кінці хорд, тобто  $z=b$ , тоді початкова точка  $x_0=a$  (рис. 3.10, б);



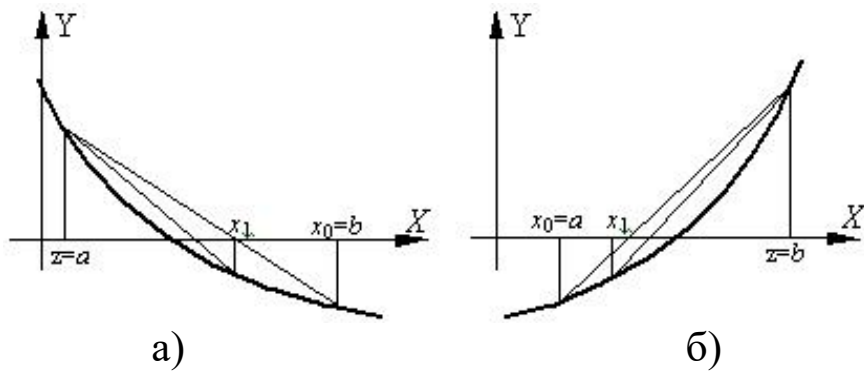


Рис. 3.10

У результаті ітераційний процес наближення до кореня реалізується рекурентною формулою:

для випадку а)

$$x_{n+1} = x_n - \frac{f(x_n)}{f(x_n) - f(a)}(x_n - a); \quad (3.11)$$

для випадку б)

$$x_{n+1} = x_n - \frac{f(x_n)}{f(x_n) - f(b)}(x_n - b); \quad (3.12)$$

Процес пошуку триває доти, поки не виконається умова

$$|x_{n+1} - x_n| \leq \epsilon \text{ або } |h| \leq \epsilon. \quad (3.13)$$

Метод забезпечує швидку збіжність, якщо  $f(x)f''(x) > 0$ , тобто хорди фіксуються в тому кінці інтервалу  $[a, b]$ , де знаки функції  $f(x)$  і її кривизни  $f''(x)$  збігаються.

Схема алгоритму уточнення кореня *методом хорд* наведена на рис. 3.11.

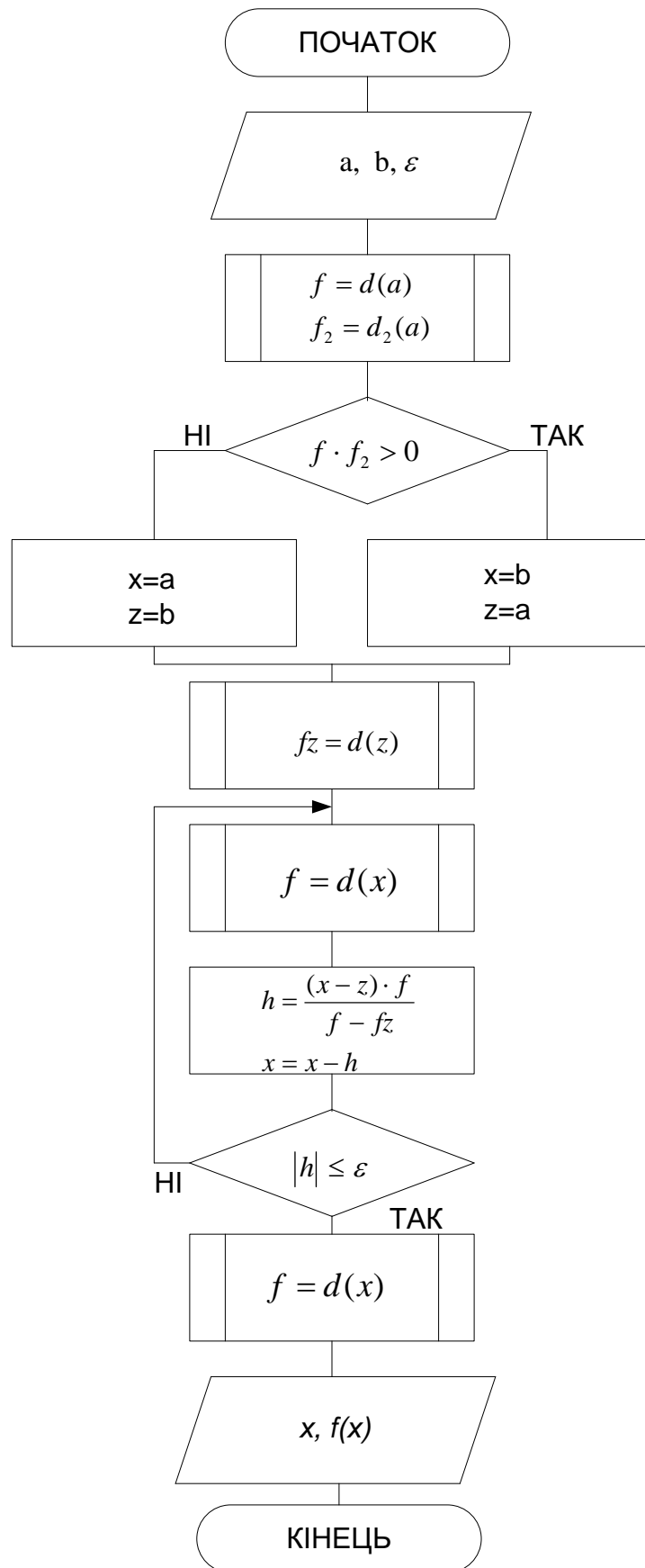


Рис. 3.11. Схема алгоритму уточнення кореня методом хорд

#### 4. КОМП'ЮТЕРНЕ ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ. СТАТИСТИЧНЕ ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ. МЕТОД МОНТЕ-КАРЛО

Як ми вже зазначали, реальні процеси та системи можна досліджувати за допомогою двох типів математичних моделей з погляду їх побудови: аналітичних і імітаційних.

В аналітичних моделях поведінка *реальних процесів і систем (РПС)* задається у вигляді явних функціональних залежностей (рівнянь лінійних або нелінійних, диференціальних або інтегральних, систем цих рівнянь). Однак одержати ці залежності вдається тільки для порівняно простих РПС. Коли явища складні й різноманітні, досліднику доводиться йти на спрощені уявлення складних РПС. У результаті аналітична модель стає занадто грубим наближенням до дійсності. Якщо все-таки для складних РПС вдається одержати аналітичні моделі, то найчастіше вони перетворюються у важку проблему, яку дуже непросто вирішити. Тому дослідник змушений часто використовувати *імітаційне моделювання*.

Зупинимось більш докладно на імітаційному моделюванні.

*Імітаційне моделювання* являє собою чисельний метод проведення на ЕОМ обчислювальних експериментів з математичними моделями, що імітують поведінку реальних об'єктів, процесів і систем у часі в пліні заданого періоду. При цьому функціонування РПС розбивається на елементарні явища, підсистеми й модулі. Функціонування цих елементарних явищ, підсистем і модулів описується набором алгоритмів, які імітують елементарні явища зі збереженням їх логічної структури й послідовності протікання в часі.

"*Імітаційне моделювання*" (ІМ) - це подвійний термін. "Імітація" і "моделювання" - це синоніми. Фактично всі галузі науки й техніки є моделями реальних процесів. Щоб відрізнити математичні моделі, дослідники стали давати їм додаткові назви. Термін "*імітаційне моделювання*" означає, що ми маємо справу з такими математичними моделями, за допомогою яких не можна заздалегідь обчислити або передбачити поведінку системи, а для передбачення

поведінки системи необхідний обчислювальний експеримент (імітація) на математичній моделі при заданих вихідних даних.

Основні переваги ІМ:

- можливість опису поведінки компонент (елементів) процесів або систем на високому рівні деталізації;
- відсутність обмежень між параметрами ІМ і станом зовнішнього середовища РПС;
- можливість дослідження динаміки взаємодії компонент у часі і просторі параметрів системи.

Ці переваги забезпечують імітаційному методу значне поширення.

Рекомендується використовувати ІМ в таких випадках:

1. Якщо не існує закінченої постановки задачі дослідження, і йде процес пізнання об'єкта моделювання. ІМ служить засобом вивчення явища.

2. Якщо аналітичні методи є, але математичні процеси складні і трудомісткі, і ІМ дає більш простий спосіб розв'язання задачі.

3. Коли, крім оцінки впливу параметрів (змінних) процесу або системи, бажано здійснити спостереження за поведінкою компонентів (елементів) *процесу* або *системи* (ПС) протягом певного періоду.

4. Коли ІМ виявляється єдиним засобом дослідження складної системи, через неможливість спостереження явищ у реальних умовах (реакції термоядерного синтезу, дослідження космічного простору).

5. Коли необхідно контролювати протікання процесів або поведінку систем шляхом затримки чи прискорення явищ у ході імітації.

6. При підготовці фахівців нової техніки, коли на ІМ забезпечується можливість придбання навичок в експлуатації нової техніки.

7. Коли вивчаються нові ситуації в РПС. У цьому випадку імітація служить для перевірки нових стратегій і правил проведення натурних експериментів.

8. Коли особливе значення має послідовність подій у проєктованих ПС, і модель використовується для прорахування вузьких місць у функціонуванні РПС.

Однак ІМ поряд з перевагами має такі *недоліки*:

- розроблення гарної ІМ часто обходиться дорожче створення аналітичної моделі й вимагає більших витрат часу;
- може виявитися, що ІМ неточна (що буває часто), і ми не в змозі виміряти ступінь цієї неточності;
- найчастіше дослідники звертаються до ІМ, не уявляючи тих труднощів, з якими вони зустрінуться, і здійснюють при цьому ряд помилок методологічного характеру.

Саме тому, не вважаючи на недоліки, імітаційне моделювання є одним з найбільш широко використовуваних методів при розв'язанні задач синтезу й аналізу складних процесів і систем.

Одним з видів ІМ є статистичне ІМ, що дозволяє відтворювати на ЕОМ функціонування складних випадкових процесів.

При дослідженні складних систем, що піддаються випадковим впливам, використовуються імовірнісні аналітичні моделі й імовірнісні ІМ.

В імовірнісних аналітичних моделях вплив випадкових факторів ураховується за допомогою завдання імовірнісних характеристик випадкових процесів (закони розподілу ймовірностей, спектральні щільності або кореляційні функції). При цьому побудова імовірнісних аналітичних моделей є складною обчислювальною задачею. Тому імовірнісне аналітичне моделювання використовують для вивчення порівняно простих систем.

Відомо, що введення випадкових впливів в ІМ не вносить принципових ускладнень, тому дослідження складних випадкових процесів проводиться, як правило, на ІМ.

В імовірнісному ІМ оперують не характеристиками випадкових процесів, а конкретними випадковими числовими значеннями параметрів ПС. При цьому результати, отримані при відтворенні на імітаційній моделі розглянутого процесу, є випадковими реалізаціями. Тому для знаходження об'єктивних і стійких характеристик процесу потрібно його багаторазове відтворення з наступною статистичною обробкою отриманих даних. Саме тому дослідження складних процесів і систем, підданих випадковим впливам, за допомогою ІМ прийнято називати статистичним моделюванням.

Статистична модель випадкового процесу - це алгоритм, за допомогою якого імітують роботу складної системи, що зазнає випадкових впливів; імітують взаємодію елементів системи, що носить імовірнісний характер.

Метод Монте-Карло - це чисельний метод, що моделює на ЕОМ псевдовипадкові числові послідовності з заданими імовірнісними характеристиками.

Методика статистичного моделювання складається з таких етапів:

1. Моделювання на ЕОМ псевдовипадкових послідовностей із заданою кореляцією та законом розподілу ймовірностей (метод Монте-Карло), що імітують на ЕОМ випадкові значення параметрів при кожному випробуванні.

2. Перетворення отриманих числових послідовностей на *імітаційних математичних моделях*.

3. Статистична обробка результатів моделювання.

Узагальнений алгоритм методу статистичних випробувань наведено на рис. 4.1.

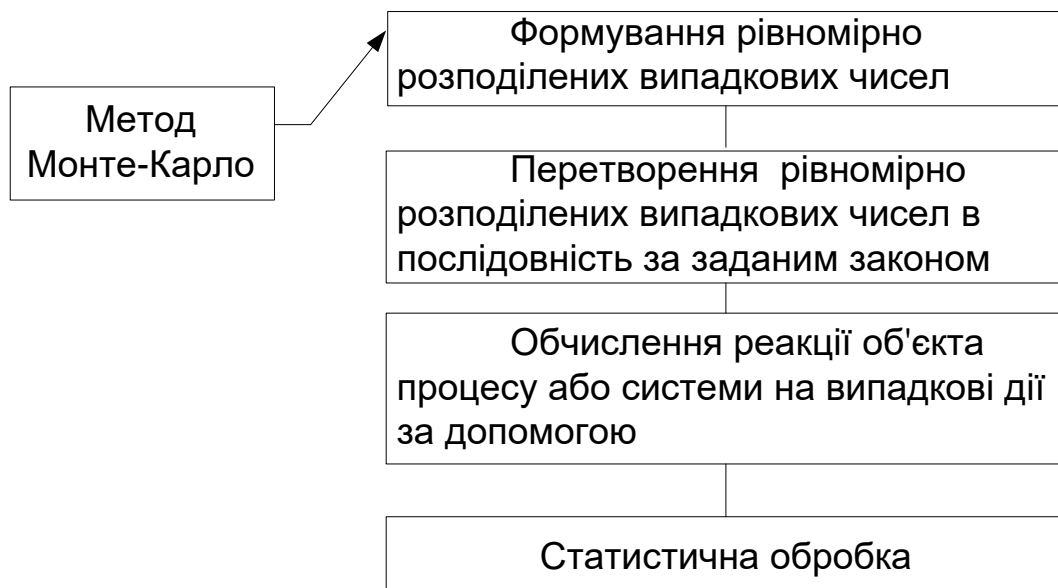


Рис.4.1. Узагальнений алгоритм методу статистичних випробувань

При реалізації на ЕОМ статистичного моделювання виникає задача одержання (генерування) на ЕОМ випадкових числових послідовностей із заданими імовірнісними характеристиками, у яких кожне число - це імітація випадкового значення якого-небудь параметра реального процесу або системи, яке підпадає під випадкове збурення.

У математичній літературі часто використовується терміни "послідовність випадкових чисел" або просто "випадкові числа".

Після створення ЕОМ почалися пошуки ефективних алгоритмів одержання (генерування на ЕОМ) послідовностей випадкових чисел, придатних для програмної реалізації.

Послідовності випадкових чисел, які створено детерміністськими способами (тобто за допомогою спеціальних алгоритмів), називаються псевдовипадковими або квазівипадковими. Надалі ми будемо їх називати просто випадковими послідовностями, розуміючи, що вони просто створюють враження випадкових.

Задачу *генерування випадкових чисел* на ЕОМ із заданим законом розподілу розв'язують у кілька етапів.

Спочатку одержують послідовність рівномірно розподілених на інтервалі  $[0, 1]$  псевдовипадкових чисел.

З рівномірно розподіленої послідовності одержують послідовність псевдовипадкових чисел із заданим законом розподілу в заданому інтервалі.

Рівномірним називається такий розподіл, при якому кожне можливе випадкове число рівновірогідне. Звичайно, якщо спеціально не обумовлений закон розподілу випадкових чисел, то мають на увазі рівномірний розподіл.

Сутність алгоритмічних методів одержання рівномірно розподілених псевдовипадкових чисел полягає в тому, що псевдовипадкові числа одержують за допомогою деякої рекурентної формули  $x_{i+1} = f(x_i)$ , де кожне наступне ( $i+1$ ) значення утворюється з попереднього (або групи попередніх) шляхом застосування деякого алгоритму, що містить логічні й арифметичні операції (дод. В 3).

Відома велика кількість імітації рівномірного розподілу (методи відрахувань, підсумовування, усікання, перемішування). Загальними для всіх цих методів є такі вимоги:

1. Кількість операцій для одержання кожного псевдовипадкового числа повинна бути мінімальною.
2. Випадкові числа генеруються як можна менш кореляційними, а їх розподіл – близьким до рівномірного.

### **Метод середини квадрата**

Перший алгоритмічний метод одержання рівномірно розподілених псевдовипадкових чисел запропонував Джон фон Нейман (один з засновників кібернетики).

*Метод одержав назву "метод середини квадрата".*

Суть методу: попереднє випадкове число підноситься до квадрата, а потім з результату вилучають середні цифри.

Наприклад: нехай  $x_0 = 0.2061$ , тоді  $x_0^2 = 0.04|2477|21$ ;  
 $x_1 = 0.2477$ ,  $x_1^2 = 0.06|1355|29$ ;  
 $x_2 = 0.1355$ ,  $x_2^2 = 0.01|8360|25$ ;

і т. д.



Як ми бачимо, *метод середини квадрата* досить добре повинен "перемішувати" попереднє число. Однак він має такі недоліки:

1. Якщо який-небудь член послідовності виявиться рівним нулю, то всі наступні члени також будуть нулями.
2. Послідовності мають тенденцію "зациклюватися", тобто зрештою утворюють цикл, який повторюється нескінченне число раз.

Властивість "зациклюватися" притаманна всім послідовностям, побудованим за рекурентною формулою  $x_{i+1} = f(x_i)$ .

Цикл, що повторюється, називається періодом. Довжина періоду в різних послідовностях різна. Чим більше, тем краще.

### **Лінійний конгруентний метод**

Найкращі з відомих сьогодні методів імітації випадкових чисел - це окремі випадки схеми, що запропонована в 1948 році Д.Х. Лемером.

Суть методу: вибираємо чотири "магічні числа":

$x_0$  – початкове значення,  $x_0 \geq 0$ ;

$a$  – множник,  $a \geq 0$ ;

$c$  – збільшення,  $c \geq 0$ ;

$m$  – модуль,  $m > x_0$ ,  $m > a$ ,  $m > c$ .

Тоді послідовність випадкових чисел, яка створюється, визначається співвідношенням

$$x_{i+1} = (ax_i + c)(m), n \geq 0, \quad (4.1)$$

таким чином, кожне випадкове число - це залишок від ділення  $(ax_i+c)$  на  $m$  (операція Mod - "визначення залишку", термін узятий від слова "modulo" – у перекладі "залишок").

Послідовність, отримана зі співвідношення (4.1), називається лінійною конгруентною послідовністю.

Приклад:  $x_0 = a = c = 7, m = 10$ .

Тоді послідовність має вигляд: 7, 6, 9, 0, 7, 6, 9, 0,...

Як видно, при обраних значеннях "магічних чисел" послідовність майже відразу "зациклилася", довжина періоду = 4.

Із цього прикладу видно, що "магічні числа" не можна вибирати довільно. Проведено багато досліджень і доведено теореми з питань "Як правильно вибирати "магічні числа".

Метод одержання випадкових чисел при  $c=0$  називається "мультиплікативним конгруентним методом", при  $c \neq 0$  - "змішаний конгруентним методом". При  $c=0$  вироблення послідовностей відбувається швидше, але при цьому зменшується довжина періоду послідовностей.

Спочатку в методі Лемера було прийнято  $c=0$ . Ідея одержання більш довгих послідовностей за рахунок  $c \neq 0$  належить Томпсону і Ротенбергу незалежно один від одного.

Вибір модуля  $m$ . Для одержання довгих послідовностей і для збільшення швидкості обчислення рекомендується  $m$  вибирати рівним розміру машинного слова. Для 32-х розрядного машинного слова  $m = 2^{31} = 2147483648$  (лівий нульовий біт слова відведений під знак числа).

При цьому в 32-х розрядному машинному слові максимальне ціле число, що може бути розміщене в машинному слові, дорівнює  $\omega = 2^{31} - 1 = 2147483647$ .

Тоді  $m = \omega + 1$ .

Значення множника також впливає на довжину періоду послідовностей. З цього питання також проведено багато досліджень.

Лінійні конгруентні послідовності - не єдине із запропонованих джерел випадкових чисел. Його можна узагальнити, перетворивши, наприклад, у квадратичний конгруентний метод:

$$x_{i+1} = (dx_i^2 + a \cdot X_n + c) \text{Mod}(m).$$

Відомий квадратичний метод, запропонований Р. Ковеем:

$$x_{i+1} = x_i(x_i + 1) \text{Mod}(2^0).$$

Відомий метод одержання випадкових чисел, де реалізується послідовність Фібоначчі:

$$x_{i+1} = (x_i + x_{i-1}) \text{Mod}(m).$$

Відомий також метод одержання випадкових чисел, запропонований Гріном:

$$x_{i+1} = (x_i + x_{i-1} - k) \text{Mod}(m),$$

де  $k$  - велике число.

Є ще, так звані, адитивні методи, де не потрібні операції множення й ділення, та інші методи.

## 5. КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ І РОЗВ'ЯЗАННЯ ЛІНІЙНИХ БАГАТОВИМІРНИХ СИСТЕМ

При моделюванні задач управління і планування виробництва, визначення оптимального розміщення устаткування, оптимального плану виробництва, оптимального плану перевезень вантажів (транспортні задачі), розподілу кадрів тощо, може бути покладена в основу гіпотеза лінійного уявлення реального світу.

Математичні моделі таких задач є лінійними рівняннями. Якщо задача багатовимірна, то її математична модель задається системою лінійних рівнянь.

У загальному випадку система лінійних рівнянь має вигляд

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 + \dots + a_{2n}x_n = b_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 + \dots + a_{3n}x_n = b_3, \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + a_{n4}x_4 + \dots + a_{nn}x_n = b_n, \end{cases}$$

де  $a_{ij}$  - коефіцієнти при невідомих системи,

$b_i$  - вільні члени,

$x_j$  - невідомі системи,

$i = \overline{1, n}$  - номер рядка,

$j = \overline{1, n}$  - номер стовпця,

$n$  - порядок системи.

У матричній формі система лінійних рівнянь має вигляд

$$A \cdot \bar{X} = \bar{B},$$

де  $A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$  - матриця коефіцієнтів системи порядку  $(n \times n)$ ;

$$\bar{X} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} - \text{вектор невідомих системи розмірністю } n;$$

$$\bar{B} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix} - \text{вектор вільних членів розмірністю } n.$$

Чисельні методи розв'язання *систем лінійних рівнянь (СЛР)* можна поділити на дві групи:

- 1) точні або прямі методи;
- 2) наближені методи.

Наближені методи реалізують на ЕОМ знаходження кореня з заданою точністю і є ітераційними методами.

Точні методи дозволяють одержати розв'язок системи за кінцеве число ітерацій. До точних методів належать:

- правило Крамера;
- *метод Гаусса*;
- метод прогону.

### **Розв'язання систем лінійних рівнянь методом Гаусса**

*Метод Гаусса* є точним методом. Він дозволяє одержати розв'язок системи за кінцеве число арифметичних дій. В основі методу лежить ідея послідовного виключення невідомих. Метод складається з двох етапів. На першому етапі (прямий хід) система за допомогою послідовного виключення невідомих зводиться до трикутного вигляду. На другому етапі (зворотний хід) із системи трикутного виду послідовно, у зворотному порядку, починаючи з  $n$ -го рівняння, визначають невідомі системи.

Як приклад розглянемо систему 4 порядку.

$$\begin{cases} a_{11}^{(0)} x_1 + a_{12}^{(0)} x_2 + a_{13}^{(0)} x_3 + a_{14}^{(0)} x_4 = b_1^{(0)}, \\ a_{21}^{(0)} x_1 + a_{22}^{(0)} x_2 + a_{23}^{(0)} x_3 + a_{24}^{(0)} x_4 = b_2^{(0)}, \\ a_{31}^{(0)} x_1 + a_{32}^{(0)} x_2 + a_{33}^{(0)} x_3 + a_{34}^{(0)} x_4 = b_3^{(0)}, \\ a_{41}^{(0)} x_1 + a_{42}^{(0)} x_2 + a_{43}^{(0)} x_3 + a_{44}^{(0)} x_4 = b_4^{(0)}, \end{cases} \quad (5.1)$$

Прямий хід. На першому кроці прямого ходу ( $k=1$ ) знаходимо  $x_1$  з першого рівняння системи (5.1).

$a_{11}^{(0)}$  - головний елемент першого рядка.

Якщо  $a_{11}^{(0)} \neq 0$ , то

$$x_1 = -\frac{a_{12}^{(0)}}{a_{11}^{(0)}} x_2 - \frac{a_{13}^{(0)}}{a_{11}^{(0)}} x_3 - \frac{a_{14}^{(0)}}{a_{11}^{(0)}} x_4 + \frac{b_1^{(0)}}{a_{11}^{(0)}}, \quad (5.2)$$

Позначимо

$$\frac{a_{12}^{(0)}}{a_{11}^{(0)}} = a_{12}^{(1)}, \frac{a_{13}^{(0)}}{a_{11}^{(0)}} = a_{13}^{(1)}, \frac{a_{14}^{(0)}}{a_{11}^{(0)}} = a_{14}^{(1)}, \frac{b_1^{(0)}}{a_{11}^{(0)}} = b_1^{(1)}. \quad (5.3)$$

Підставляючи (6.3) в (6.2), одержимо

$$x_1 = a_{12}^{(1)} x_2 - a_{13}^{(1)} x_3 - a_{14}^{(1)} x_4 + b_1^{(1)}, \quad (5.4)$$

де  $a_{1j}^{(1)} = \frac{a_{1j}^{(0)}}{a_{11}^{(0)}}, j = 2, 3, 4 = \overline{(k+1), n}$ ;

$$b_1^{(1)} = \frac{b_1^{(0)}}{a_{11}^{(0)}}.$$

Підставляємо вираз (5.4) в 2, 3 і 4 рівняння системи (5.1), отримаємо

$$\begin{cases} (a_{22}^{(0)} - a_{21}^{(0)} \cdot a_{12}^{(1)})x_2 + (a_{23}^{(0)} - a_{21}^{(0)} \cdot a_{13}^{(1)})x_3 + (a_{24}^{(0)} - a_{21}^{(0)} \cdot a_{14}^{(1)})x_4 = b_2^{(0)} - a_2^{(0)} - a_{21}^{(0)} \cdot b_1^{(1)}, \\ (a_{32}^{(0)} - a_{31}^{(0)} \cdot a_{12}^{(1)})x_2 + (a_{33}^{(0)} - a_{31}^{(0)} \cdot a_{13}^{(1)})x_3 + (a_{34}^{(0)} - a_{31}^{(0)} \cdot a_{14}^{(1)})x_4 = b_3^{(0)} - a_3^{(0)} - a_{31}^{(0)} \cdot b_1^{(1)}, \\ (a_{42}^{(0)} - a_{41}^{(0)} \cdot a_{12}^{(1)})x_2 + (a_{43}^{(0)} - a_{41}^{(0)} \cdot a_{13}^{(1)})x_3 + (a_{44}^{(0)} - a_{41}^{(0)} \cdot a_{14}^{(1)})x_4 = b_4^{(0)} - a_4^{(0)} - a_{41}^{(0)} \cdot b_1^{(1)}, \end{cases}$$

Позначивши коефіцієнти при невідомих отриманої системи через  $a_{ij}^{(1)}$ , а вільні члени через  $b_i^{(1)}$  перепишемо отриману систему:

$$\begin{cases} a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + a_{24}^{(1)}x_4 = b_2^{(1)}, \\ a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 + a_{34}^{(1)}x_4 = b_3^{(1)}, \\ a_{42}^{(1)}x_2 + a_{43}^{(1)}x_3 + a_{44}^{(1)}x_4 = b_4^{(1)}. \end{cases} \quad (5.5)$$

де

$$a_{ij}^{(1)} = a_{ij}^{(0)} - a_{i1}^{(0)} \cdot a_{1j}^{(1)},$$

$$b_i^{(1)} = b_i^{(0)} - a_{i1}^{(0)} \cdot b_1^{(1)},$$

$$i = \overline{(k+1), n}; j = \overline{(k+1), n}.$$

Таким чином, у результаті виконання першого кроку прямого ходу вихідна система (5.1)  $n$ -го порядку перетворена до сукупності рівняння (5.4) і системи лінійних рівнянь (5.5), порядок якої дорівнює  $n-1$ .

На другому кроці прямого ходу ( $k=2$ ) з першого рівняння системи (5.5) знаходимо  $x_2$ .

$a_{22}^{(1)}$  - головний елемент першого рядка системи (5.5).

Якщо  $a_{22}^{(1)} \neq 0$ , то з першого рівняння системи (5.5) маємо

$$x_2 = -a_{23}^{(2)}x_3 - a_{24}^{(2)}x_4 + b_2^{(2)} \quad (5.6)$$

де

$$a_{2j}^{(2)} = a_{2j}^{(1)} / a_{22}^{(1)},$$

$$b_2^{(2)} = b_2^{(1)} / a_{22}^{(1)}.$$

$$j = 3, 4 = \overline{(k+1), n}$$

Підставивши вирази (5.6) у 2 та 3 рівняння системи (5.5), отримаємо нову систему лінійних рівнянь, порядок якої дорівнює  $n-2$ :

$$\begin{cases} a_{33}^{(2)} x_3 + a_{34}^{(2)} x_4 = b_3^{(2)}, \\ a_{34}^{(2)} x_3 + a_{44}^{(2)} x_4 = b_4^{(2)}. \end{cases} \quad (5.7)$$

де

$$\begin{aligned} a_{ij}^{(2)} &= a_{ij}^{(1)} - a_{i2}^{(2)} \cdot a_{2j}^{(2)}, \\ b_i^{(2)} &= b_i^{(1)} - a_{i2}^{(2)} \cdot b_2^{(2)}, \\ i &= \overline{(k+1), n}; \quad j = \overline{(k+1), n}. \end{aligned}$$

Таким чином, у результаті виконання другого кроку прямого ходу вихідна система (5.1) перетворена до сукупності рівнянь (5.4), (5.6) і системи лінійних рівнянь (5.7), порядок якої дорівнює  $n-2$ .

На третьому кроці прямого ходу ( $k=3$ ) із системи (5.7) знаходимо  $x_3$ .

$a_{33}^{(2)}$  - головний елемент системи (5.7).

Якщо  $a_{33}^{(2)} \neq 0$ , то з першого рівняння системи (5.7) маємо

$$x_3 = -a_{34}^{(2)} x_4 + b_3^{(2)}, \quad (5.8)$$

де

$$\begin{aligned} a_{3j}^{(3)} &= a_{3j}^{(2)} / a_{33}^{(2)}, \\ b_3^{(3)} &= b_3^{(2)} / a_{33}^{(2)}, \\ j &= \overline{1 = (k+1), n} \end{aligned}$$

Підставивши вираз (5.8) для  $x_3$  у друге рівняння системи (5.7), отримаємо:



$$a_{11}^{(3)} x_4 = b_4^{(3)}, \quad (5.9)$$

де

$$\begin{aligned} a_{ij}^{(3)} &= a_{ij}^{(2)} - a_{i3}^{(2)} \cdot a_{3j}^{(3)}, \\ b_i^{(3)} &= b_i^{(2)} - a_{i3}^{(2)} \cdot a_{3j}^{(3)}, \\ i &= \overline{(k+1), n}; \quad j = \overline{(k+1), n}. \end{aligned}$$

На останньому кроці прямого ходу. Якщо  $a_{44}^{(3)} \neq 0$ , то з рівняння (5.9) маємо

$$x_4 = b_4^{(4)}, \quad (5.10)$$

де

$$b_4^{(4)} = b_4^{(3)} / a_{44}^{(3)}. \quad (5.11)$$

У результаті виконання всіх кроків прямого ходу вихідна система (5.1) зводиться до системи трикутного вигляду, отриманої об'єднанням рівнянь (5.4), (5.6), (5.8), (5.10):

$$\begin{cases} x_1 = -a_{12}^{(1)} x_2 - a_{13}^{(1)} x_3 - a_{14}^{(1)} x_4 + b_1^{(1)} \\ x_2 = -a_{23}^{(2)} x_3 - a_{24}^{(1)} x_4 + b_2^{(2)}, \\ x_3 = -a_{34}^{(3)} x_4 + b_3^{(3)}, \\ x_4 = b_4^{(4)}. \end{cases} \quad (5.12)$$

При побудові алгоритму прямого ходу обчислення організуємо в циклі по кроках, тобто  $k = \overline{1, (n-1)}$ .

Останній  $n$ -й крок прямого ходу виведемо з циклу, тому що тут реалізується тільки одне обчислення:

$$b_n = \frac{b_n}{a_{nn}} \quad (5.13)$$

У процесі виконання всіх кроків прямого ходу всі перетворення коефіцієнтів і вільних членів проводимо за отриманими раніше рекурентними формулами:

$$\begin{aligned}
 b_k^{(k)} &= b_k^{(k-1)} / a_{k,k}^{(k-1)}, \\
 b_i^{(k)} &= b_i^{(k-1)} / a_{i,k}^{(k-1)} \cdot b_k^{(k)}, \\
 a_{k,j}^{(k)} &= a_{k,j}^{(k-1)} / a_{k,k}^{(k-1)}, \\
 a_{i,j}^{(k)} &= a_{i,j}^{(k-1)} - a_{i,k}^{(k-1)} \cdot a_{k,j}^{(k)}
 \end{aligned}
 \tag{5-14}$$

де  $k = \overline{1, (n-1)}$  – номер кроку прямого ходу;

$i = \overline{(k+1), n}$  – номер рівняння систем (5.5), (5.7);

$j = \overline{(k+1), n}$ .

У процесі зворотного ходу з системи (5.12) невідомі визначаються у зворотному порядку. Значення кореня  $x_4$  знаходять із останнього рівняння системи (5.12). Далі  $x_4$  використовується для відшукування кореня  $x_3$  з 3-го рівняння, далі  $x_3$  і  $x_4$  використовуються для відшукування  $x_2$  з 2-го рівняння системи (5.12), і, нарешті,  $x_2$ ,  $x_3$  і  $x_4$  використовуються для відшукування  $x_1$  з 1-го рівняння системи (5.12).

Всі обчислення зворотного ходу проводимо в циклі по  $i$ , де  $i = \overline{(n-1), 1}$  (за рекурентними формулами),  $b_i = b_i - x_i \cdot a_{i,j}$ ,  $i = \overline{(n-1), 1}$ ,  $j = \overline{(i+1), n}$ ,  $x_i = b_i$ .

Розглянутий вище найпростіший варіант *методу Гаусса*, називаний схемою єдиного розподілу, має такий недолік: якщо ведучий елемент  $a_{kk}$  якого-небудь рядка виявиться рівним нулю, то цей метод формально непридатний, хоча система може мати єдиний розв'язок. Із цих міркувань у схему алгоритму додано пошук ненульового головного елемента.

На рис 5.1 зображена укрупнена схема алгоритму (блок-схема) *методу Гаусса*. На рис. 5.2 - 5.6 зображено алгоритми окремих блоків методу.

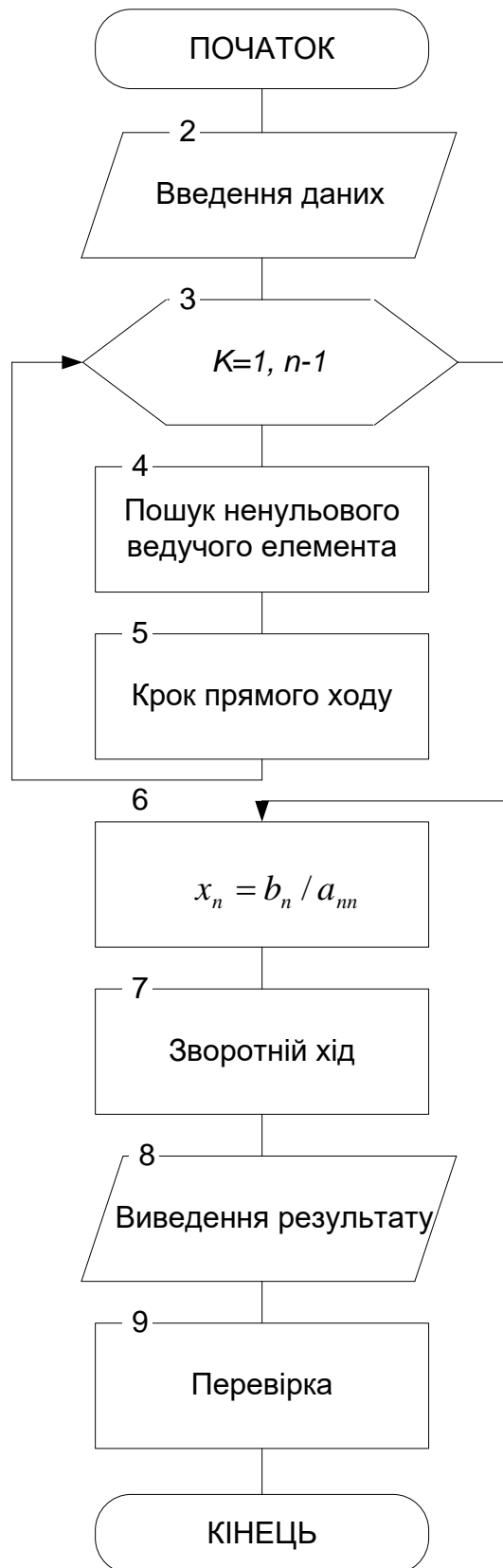


Рис. 5.1. Укрупнена схема алгоритму (блок-схема) методу Гаусса

Блок 2. За допомогою двох вкладених циклів з керівними змінними  $i=1,n$  і  $j=1,k$  організуємо введення коефіцієнтів  $a_{ij}$  і вільних членів  $b_i$  вихідної системи. Для того щоб надалі можна було виконати в блоці 9 перевірку результату, в алгоритмі передбачено збереження значень  $a_{ij}$  і  $b_i$  вихідної системи за допомогою переприсвоєння:  $c_{ij}=a_{ij}$  і  $d_i=b_i$ .

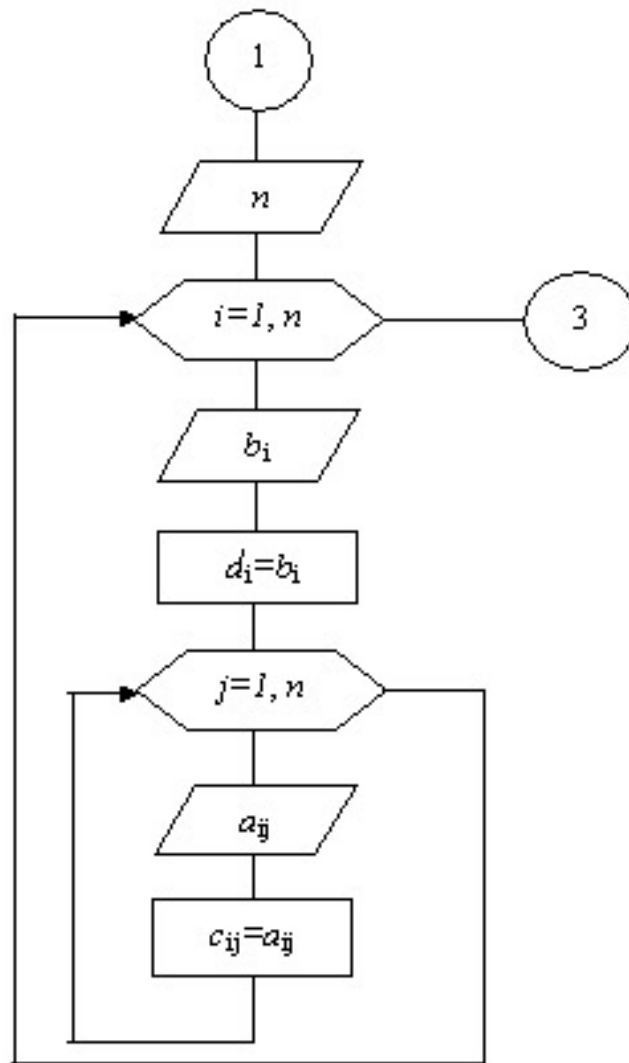


Рис. 5.2

Блок 3. Організуємо цикл по  $k$ , у середині якого проводиться обчислення по всіх кроках прямого ходу. Останній  $n$ -й крок прямого ходу виводимо з циклу.

Блок 4. На кожному кроці прямого ходу виконуємо пошук ненульового головного елемента.

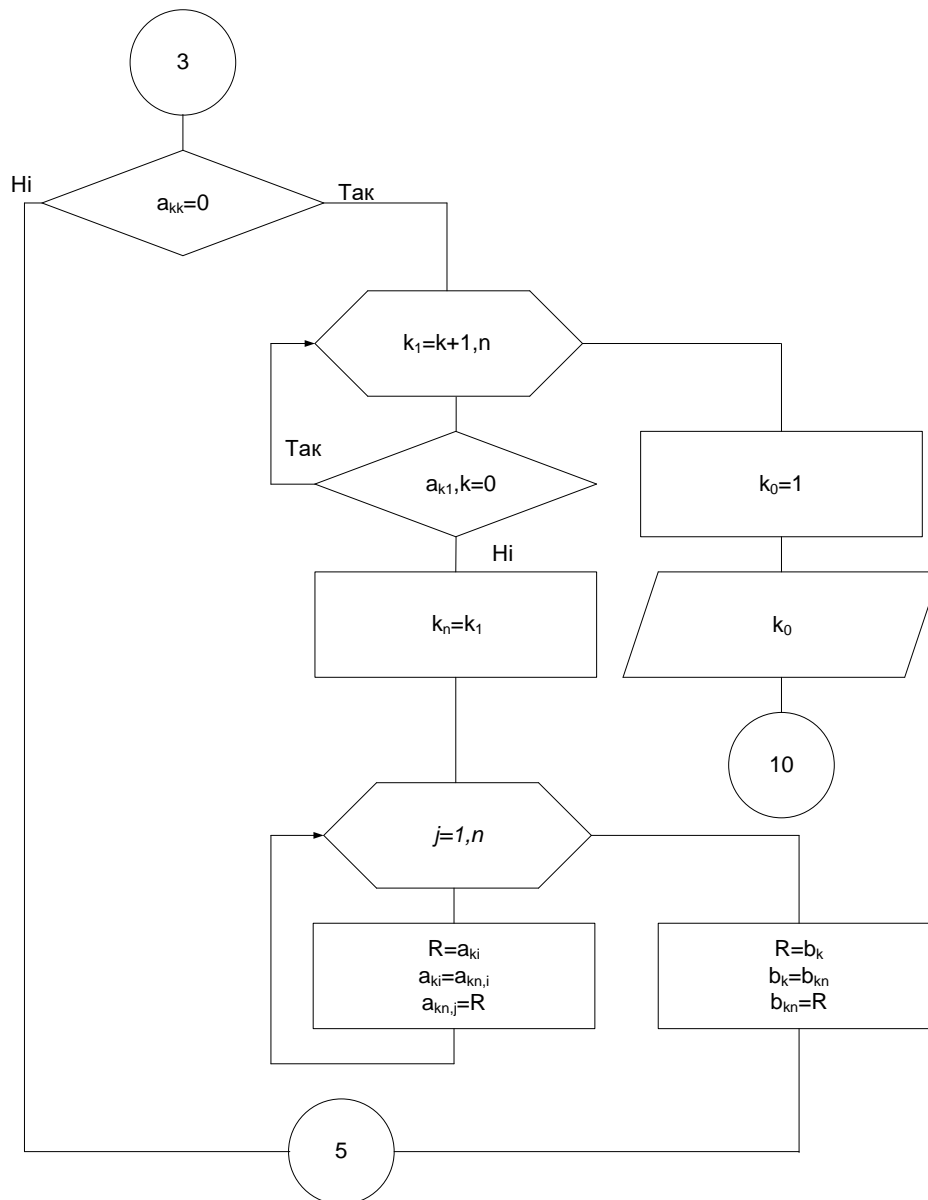


Рис. 5.3

Пошук ненульового головного елемента ведеться в такому порядку:

а) на кожному  $k$ -му кроці прямого ходу головний елемент кожного рядка порівнюється з нулем;

б) якщо в  $k$ -му рядку є нульовий головний елемент, то в  $k$ -му стовпці в циклі здійснюється пошук ненульового елемента;

в) якщо в якомусь рядку  $k_n$  такий ненульовий елемент знайдено, то рядки  $k_n$  і  $k$  поелементно у циклі по  $k_1=(k+1), n$  міняємо місцями. Для перестановки елементів використовується змінна  $R$ ;

г) якщо ненульовий головний елемент не знайдено, то коду помилки  $k_0$  привласнюємо значення 1, і розрахунки припиняються.

Блок 5 - крок прямого ходу. На кожному кроці прямого ходу проводимо виключення невідомих шляхом перетворення коефіцієнтів і вільних членів системи за отриманими раніше рекурентними формулами.

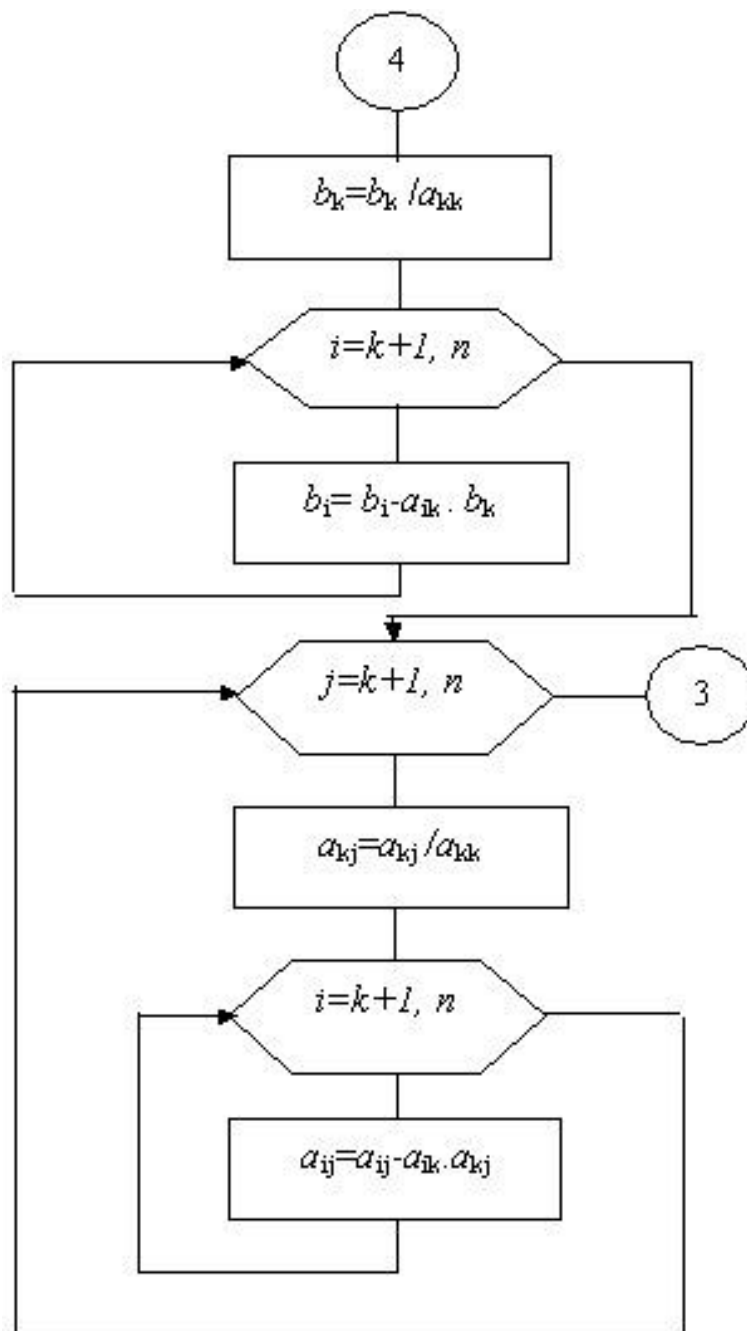


Рис. 5.4

Блок 6. У цьому блоці виведемо із циклу по k останній крок прямого ходу, тому що на цьому кроці не потрібні перетворення коефіцієнтів і вільних членів, а реалізується тільки одне обчислення

$$x_n = b_n / a_{n,n}.$$

Блок 7 - зворотний хід. У процесі зворотного ходу *методу Гаусса* з системи трикутного вигляду послідовно у зворотному порядку в циклі по  $i=(n-1), 1, -1$  знаходимо невідомі системи за рекурентною формулою

$$b_i = b_i - x_j \cdot a_{i,j}, \quad i=(n-1), 1, \quad j=(i+1), n.$$

При цьому в циклі по  $j=(i+1), n$  використано приймання послідовного віднімання  $x_j \cdot a_{i,j}$  з  $b_i$ , після чого вводиться переприсвоєння  $b_i = x_i$ .

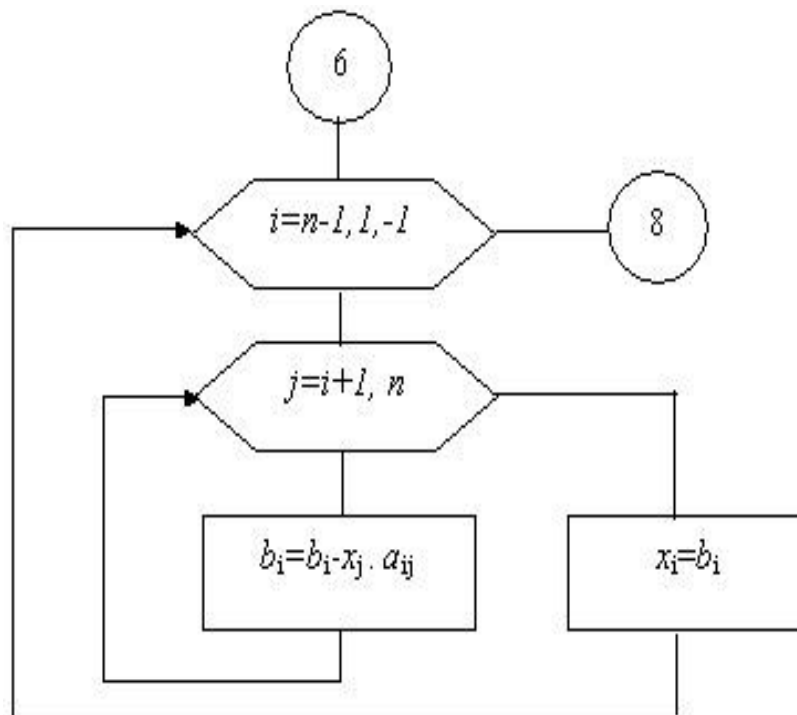


Рис. 5.5

Блок 9 - перевірка результату. У цьому блоці, підставляючи значення отриманих невідомих у вихідну систему й

використовуючи збережені значення коефіцієнтів системи  $c_{i,j}$  і вільних членів  $d_i$ , проводимо перевірку розв'язання задачі за формулою

$$F_i = -d_i - \sum_{j=1}^n C_{i,j} \cdot x_j.$$

Якщо корені системи знайдено, то  $F_i$  - це число, близьке до нуля.

Блок 9 в алгоритмі методу Гаусса рекомендується використовувати тільки в процесі налагодження методу.

Надалі при використанні методу Гаусса при розв'язанні різних прикладних задач, особливо в тих випадках, коли метод Гауса використовується всередині іншого методу, блок 9 можна вилучити, а в блоці 2 при введенні даних вихідні значення коефіцієнтів системи та її вільних членів можна не зберігати.

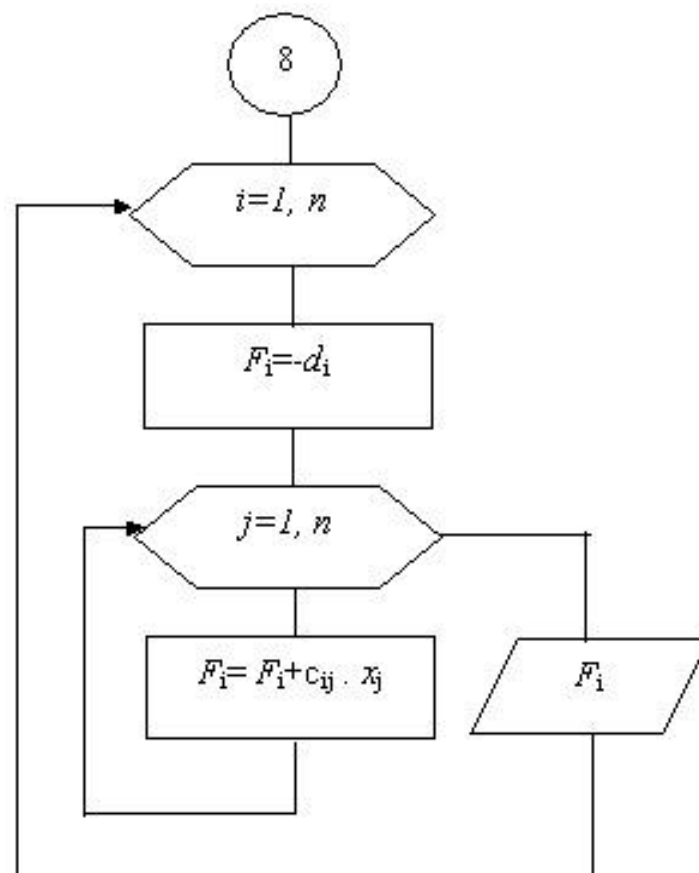


Рис. 5.6



## 6. МОДЕЛЮВАННЯ БАГАТОВИМІРНИХ НЕЛІНІЙНИХ СИСТЕМ

У задачах проектування та дослідження поведінки реальних об'єктів, процесів і систем математичні моделі повинні відображати реальні фізичні нелінійні процеси. При цьому ці процеси залежать, як правило, від багатьох змінних.

У результаті математичні моделі реальних об'єктів, процесів і систем описуються *системами нелінійних рівнянь*.

### Розв'язання систем нелінійних рівнянь

Дано систему нелінійних рівнянь

$$\begin{cases} f_1(x_1, x_2, x_3, \dots, x_n) = 0, \\ f_2(x_1, x_2, x_3, \dots, x_n) = 0, \\ \dots\dots\dots \\ f_n(x_1, x_2, x_3, \dots, x_n) = 0, \end{cases} \quad (6.1)$$

або

$$f_i(x_1, x_2, x_3, \dots, x_n) = 0, i = \overline{1 \dots n}.$$

Необхідно розв'язати цю систему, тобто знайти вектор  $\bar{X} = [x_1, x_2, x_3, \dots, x_n]$ , що задовольняє систему (6.1) з точністю  $\varepsilon$ .

Вектор  $\bar{X}$  визначає точку в  $n$ -мірному Евклідовому просторі, тобто  $\bar{X} \in$  цьому простору й задовольняє всі рівняння системи (6.1).

На відміну від систем лінійних рівнянь, для *систем нелінійних рівнянь* невідомі прямі методи розв'язання. При розв'язанні *систем нелінійних рівнянь* використовуються ітераційні методи. Ефективність усіх ітераційних методів залежить від вибору початкового наближення (початкової точки), тобто вектора  $\bar{X}^0 = [x_1^0, x_2^0, \dots, x_n^0]$ .

Область, у якій початкове наближення  $\bar{X}^0$  збігається до розв'язку, що шукають, називається областю збіжності  $G$ . Якщо



Метод простих ітерацій використовується для розв'язання таких систем лінійних рівнянь, у яких виконується умова збіжності ітераційного процесу пошуку, а саме

$$\left| \sum_{i=1}^n \frac{\delta \varphi_i}{\delta x_i} \right| < 1, j = \overline{1, n} \quad (6.4)$$

тобто сума абсолютних величин часткових похідних усіх перетворених рівнянь системи (6.2) по  $j$ -й змінній менше одиниці.

На рис. 6.1 наведено схему алгоритму розв'язання систем нелінійних рівнянь методом простих ітерацій.

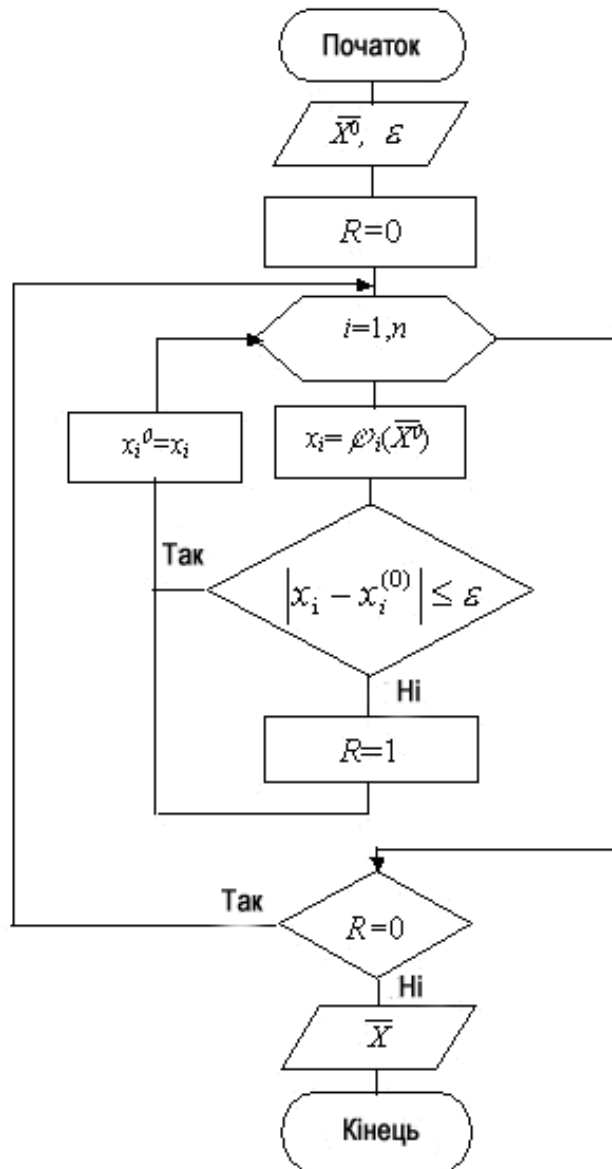


Рис. 6.1. Схема алгоритму методу простих ітерацій

Розглянемо приклад.

Дано систему нелінійних рівнянь:

$$\begin{cases} x_1^2 + x_2^2 = 1 \\ \ln x_1 - 2x_2 = -1 \end{cases}$$

Необхідно визначити область збіжності системи, вибрати початкову точку та знайти один з розв'язків системи.

1. Будуємо графіки рівнянь

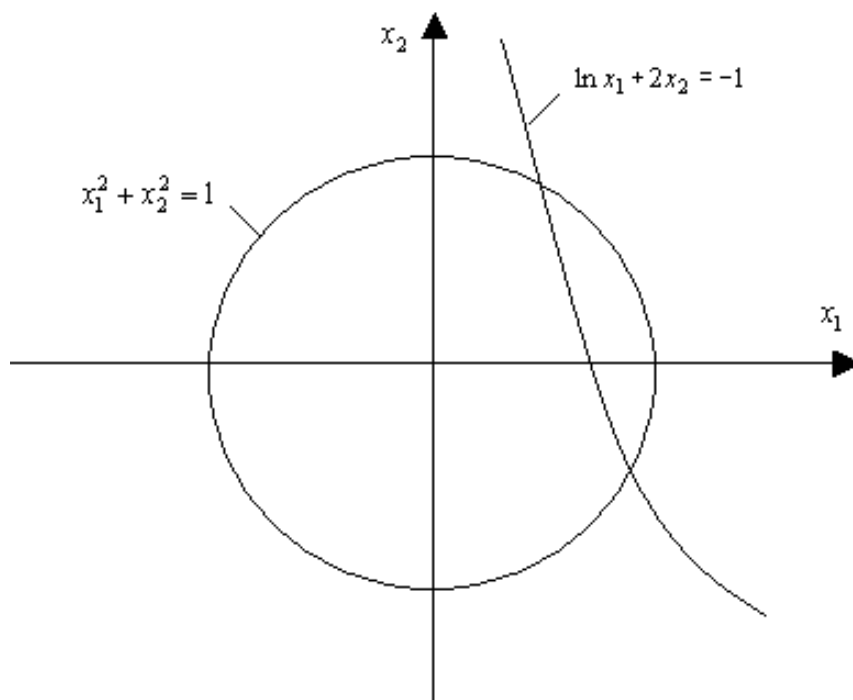


Рис. 6.2

2. Перетворимо систему для розв'язання *методом ітерацій*

$$\begin{cases} x_1 = \sqrt{1 - x_2^2} \rightarrow \varphi_1(x_1, x_2) \\ x_2 = -0,5 - 0,5 \ln x_1 \rightarrow \varphi_2(x_1, x_2) \end{cases}$$

Перевіряємо умову збіжності (6.4). Для заданої системи вона має вигляд

$$\left| \frac{\delta \varphi_1}{\delta x_1} \right| + \left| \frac{\delta \varphi_2}{\delta x_1} \right| < 1$$

$$\left| \frac{\delta \varphi_1}{\delta x_2} \right| + \left| \frac{\delta \varphi_2}{\delta x_2} \right| < 1$$

Знаходимо:

$$\frac{\delta \varphi_1}{\delta x_1} = 0; \quad \frac{\delta \varphi_1}{\delta x_2} = x_2 / \sqrt{1 - x_2^2},$$

$$\frac{\delta \varphi_2}{\delta x_1} = -1/(2x_1); \quad \frac{\delta \varphi_2}{\delta x_2} = 0.$$

У результаті умова (6.4) буде мати вигляд

$$\begin{aligned} |0| + |1/(2x_1)| < 1 \\ x_2 / \sqrt{1 - x_2^2} + |0| < 1 \end{aligned}$$

Визначаємо область збіжності G.

Границя області збіжності визначиться при розв'язанні системи;

$$\begin{cases} |1/(2x_1)| = 1 \\ x_2 / \sqrt{1 - x_2^2} = 1 \end{cases}$$

$$\text{Звідси } x_1 = 0,5; \quad x_2 = \pm \sqrt{0,5}.$$

У результаті область збіжності визначиться при  $|x_1| \geq 0,5$  та  $-0,707 \leq x_2 \leq 0,707$ .

На графіку рівнянь (рис 6.3) будемо область збіжності G.

Вибираємо початкову точку  $\bar{x}^0 = [0,8; -0,6]$ , що належить області збіжності G. Використовуючи обрану початкову точку, розв'язуємо задану систему нелінійних рівнянь.

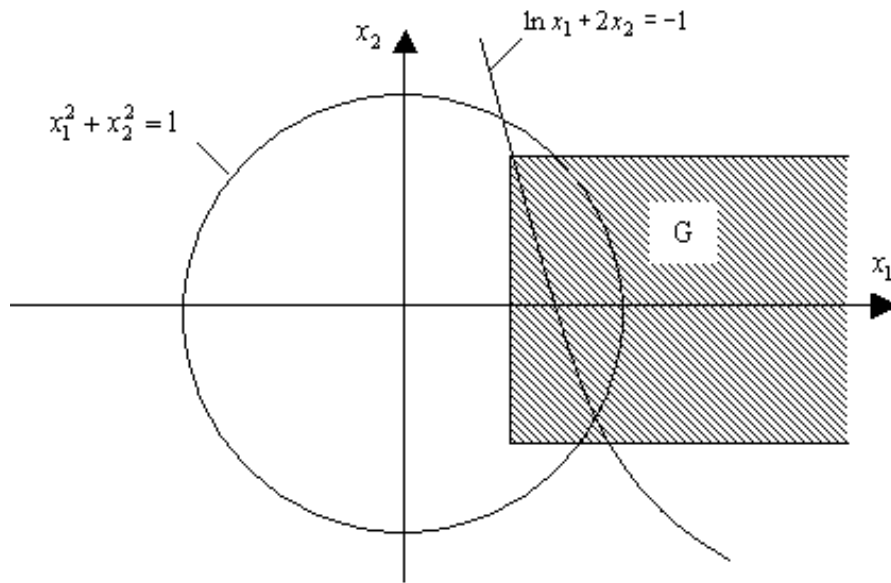


Рис. 6.3

## 6.2. Розв'язання систем нелінійних рівнянь методом Ньютона

Дано систему нелінійних рівнянь:

$$\begin{cases} f_1(x_1, x_2, x_3, \dots, x_n) = 0, \\ f_2(x_1, x_2, x_3, \dots, x_n) = 0, \\ \dots\dots\dots \\ f_n(x_1, x_2, x_3, \dots, x_n) = 0, \end{cases} \quad (6.5)$$

або

$$f_i(x_1, x_2, x_3, \dots, x_n) = 0, \quad i = \overline{1 \dots n}.$$

Необхідно розв'язати цю систему, тобто знайти вектор  $\overline{X} = [x_1, x_2, x_3, \dots, x_n]$ , що задовольняє систему (6.5) з точністю  $\varepsilon$ .

*Метод Ньютона* найпоширеніший метод розв'язання систем нелінійних рівнянь. Він забезпечує більш швидку збіжність порівняно з методом ітерацій.

В основі методу Ньютона лежить ідея лінеаризації всіх нелінійних рівнянь системи (6.5). Додамо всій системі (6.5) малі збільшення  $h_j$  і розкладемо кожне рівняння системи (6.5) у ряд Тейлора:

$$\left\{ \begin{array}{l} f_1(x_1 + h_1, x_2 + h_2, \dots, x_n + h_n) = f_1(x_1, x_2, \dots, x_n) + h_1 \frac{\delta f_1}{\delta x_1} + h_2 \frac{\delta f_1}{\delta x_2} + \dots + h_n \frac{\delta f_1}{\delta x_n} + R_1, \\ f_2(x_1 + h_1, x_2 + h_2, \dots, x_n + h_n) = f_2(x_1, x_2, \dots, x_n) + h_1 \frac{\delta f_2}{\delta x_1} + h_2 \frac{\delta f_2}{\delta x_2} + \dots + h_n \frac{\delta f_2}{\delta x_n} + R_2, \\ \dots\dots\dots \\ f_n(x_1 + h_1, x_2 + h_2, \dots, x_n + h_n) = f_n(x_1, x_2, \dots, x_n) + h_1 \frac{\delta f_n}{\delta x_1} + h_2 \frac{\delta f_n}{\delta x_2} + \dots + h_n \frac{\delta f_n}{\delta x_n} + R_n, \end{array} \right. \quad (6.6)$$

де  $h_j$  - збільшення по кожному  $x_j$ ;

$R_i$  - залишкові нелінійні члени другого та більш високих порядків кожного ряду Тейлора.

Якщо збільшення  $h_j$  такі, що змінні  $x_j$  набув значення, близькі до кореня, то будемо вважати, що ліві частини рівнянь системи (6.6) перетворюються в нулі. Тоді, відкинувши  $R_i$ , зведемо задачу розв'язання системи нелінійних рівнянь (7.5) до вирішення системи лінійних рівнянь, у якій невідомими є збільшення  $h_j$ ,  $j = \overline{1, n}$ ;

$$\left\{ \begin{array}{l} h_1 \frac{\delta f_1}{\delta x_1} + h_2 \frac{\delta f_1}{\delta x_2} + \dots + h_n \frac{\delta f_1}{\delta x_n} + R_1 = -f_1(x_1, x_2, \dots, x_n), \\ h_1 \frac{\delta f_2}{\delta x_1} + h_2 \frac{\delta f_2}{\delta x_2} + \dots + h_n \frac{\delta f_2}{\delta x_n} + R_2 = -f_2(x_1, x_2, \dots, x_n), \\ \dots\dots\dots \\ h_1 \frac{\delta f_n}{\delta x_1} + h_2 \frac{\delta f_n}{\delta x_2} + \dots + h_n \frac{\delta f_n}{\delta x_n} + R_n = -f_n(x_1, x_2, \dots, x_n), \end{array} \right. \quad (6.7)$$

Система (6.7) - система лінійних рівнянь із невідомими  $h_j$ ,  $j = \overline{1, n}$ . Запишемо систему (6.7) у матричній формі:

$$A \cdot \overline{H} = \overline{B},$$

де

$$A = \begin{bmatrix} \frac{\delta f_1}{\delta x_1} & \frac{\delta f_1}{\delta x_2} & \dots & \frac{\delta f_1}{\delta x_n} \\ \frac{\delta f_2}{\delta x_1} & \frac{\delta f_2}{\delta x_2} & \dots & \frac{\delta f_2}{\delta x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\delta f_n}{\delta x_1} & \frac{\delta f_n}{\delta x_2} & \dots & \frac{\delta f_n}{\delta x_n} \end{bmatrix} - \text{матриця коефіцієнтів системи.}$$

Матриця  $A$ , що складена з частинних похідних  $a_{ij} = \frac{\delta f_i}{\delta x_j}; i = \overline{1, n}; j = \overline{1, n}$  називається *матрицею Якобі* або *Якобіаном*.

Метод Ньютона складається з двох етапів:

1. На першому етапі реалізації методу Ньютона необхідно побудувати систему (6.7).

2. На другому етапі, починаючи з початкової точки  $\overline{X^0}$ , необхідно розв'язувати систему (6.7) на кожному кроці ітераційного процесу пошуку методом Гауса. Знайдені значення збільшень  $h_j$  використовуються як виправлення до розв'язку, отриманого на попередньому кроці пошуку, тобто

$$\begin{aligned} x_1 &= x_1 + h_{1j}, \\ x_2 &= x_2 + h_{2j}, \\ &\dots \dots \dots \\ x_n &= x_n + h_{nj}, \end{aligned} \tag{6.8}$$

або

$$x_j = x_j + h_j; j = \overline{1, n}.$$

Ітераційний процес припиняється, як тільки виконається умова

$$\begin{aligned} |h_j| &\leq \varepsilon; \\ j &= \overline{1, n} \end{aligned} \tag{6.9}$$



по всіх збільшеннях одночасно.

### Визначення матриці Якобі

У методі Ньютона на кожному кроці ітераційного процесу пошуку необхідно формувати *матрицю Якобі*, при цьому кожний елемент матриці можна визначити:

- 1) аналітично як часткову похідну  $\frac{\delta f_i}{\delta f_j}$  ;
- 2) методом чисельного диференціювання як відношення збільшення функції до збільшення аргументу, тобто  $\frac{\delta f_i}{\delta f_j} \approx \frac{\Delta f_1}{\Delta f_2}$ .

У результаті часткова похідна  $f_i(\bar{X})$  по першій координаті  $x_1$  визначиться як

$$\frac{\delta f_i}{\delta x_1} \approx \frac{f_i(x_1 + \Delta x_1, x_2, x_3 \dots x_n) - f_i(x_1, x_2, \dots, x_n)}{\Delta x_1},$$

а часткова похідна  $f_i(\bar{X})$  по координаті  $x_j$  визначиться як

$$\frac{\delta f_i}{\delta x_j} \approx \frac{f_i(x_1, x_2, \dots, x_j + \Delta x_j \dots x_n) - f_i(x_1, x_2, \dots, x_n)}{\Delta x_j},$$

де  $\Delta x_j \approx \varepsilon$ .

Метод Ньютона має переваги порівняно з іншими методами. Але для методу Ньютона також існує проблема збіжності: зі збільшенням числа невідомих область збіжності зменшується, а у випадку великих систем збіжність забезпечується, якщо початкова точка близька до шуканого розв'язку.

На рис. 6.4 наведено укрупнену схему алгоритму (блок-схему) методу Ньютона. На рис. 6.5 і 6.6 наведено схеми алгоритмів методу Ньютона з різними способами визначення *матриці Якобі*.



Рис. 6.4. Блок-схема алгоритму методу Ньютона

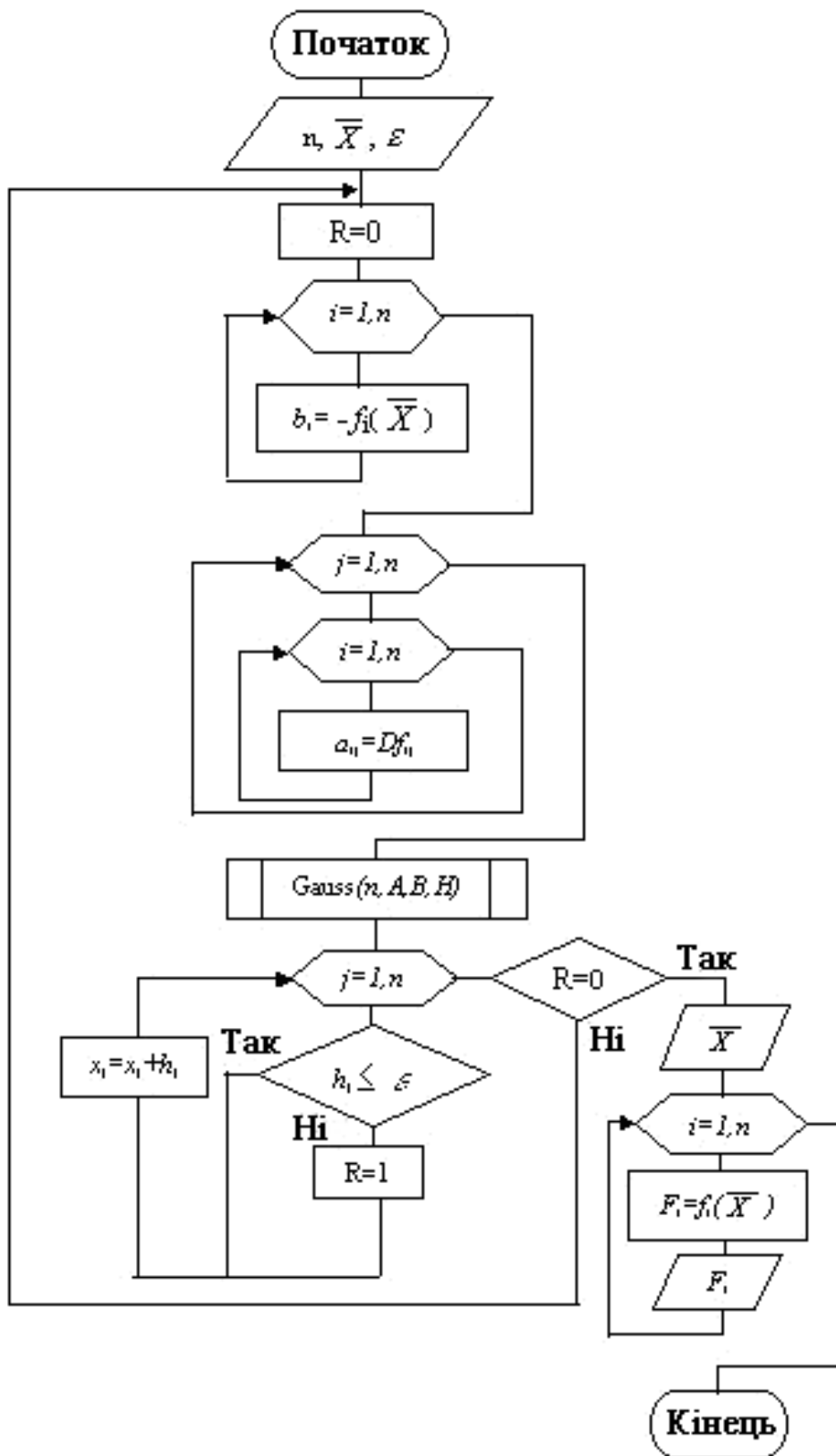


Рис. 6.5. Схема алгоритму методу Ньютона (аналітичне визначення матриці Якобі)

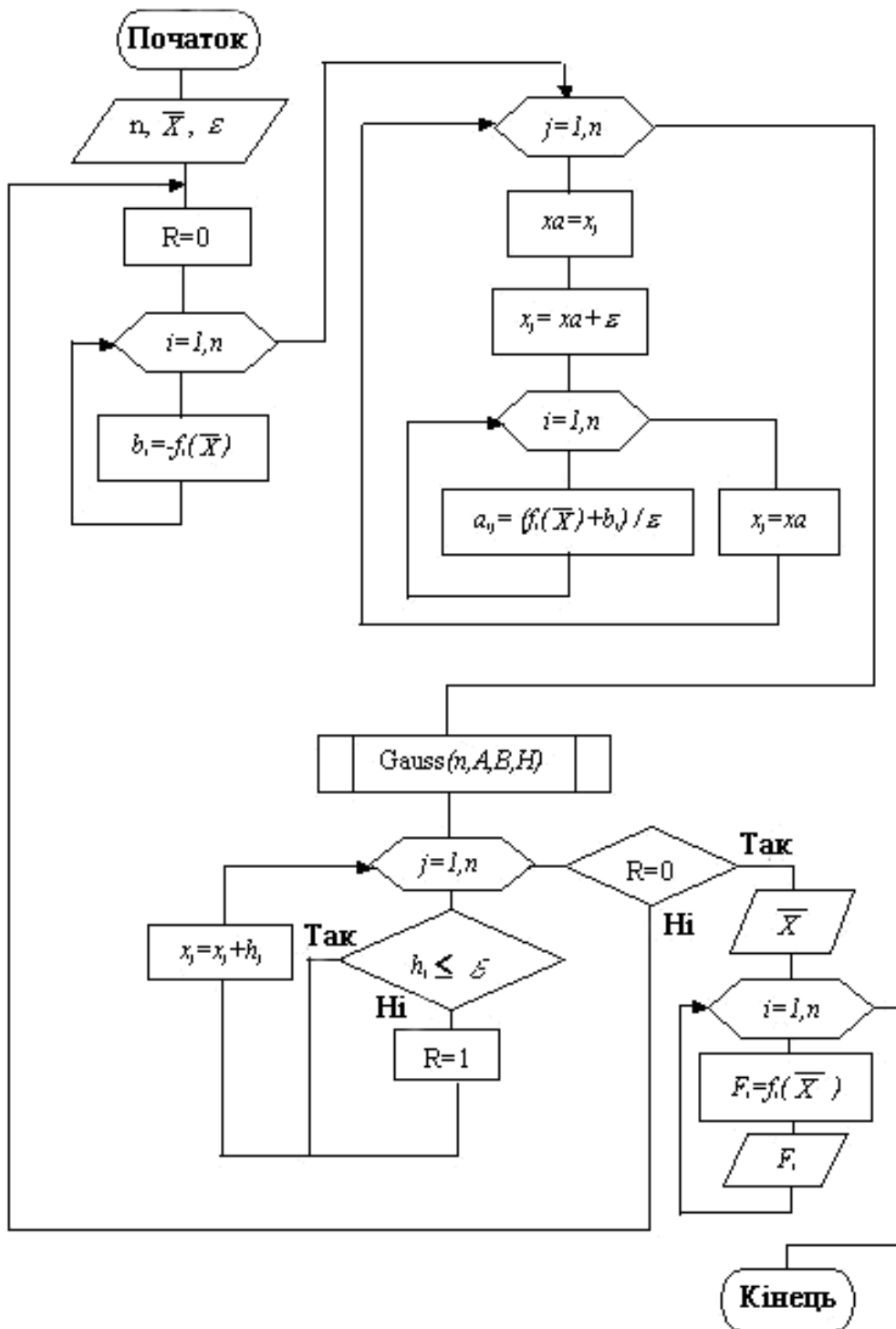


Рис. 6.6. Схема алгоритму методу Ньютона (визначення матриці Якобі за допомогою чисельного диференціювання)

## 7. КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ПРИ ОБРОБЦІ ДОСЛІДНИХ ДАНИХ

Будь-якому фахівцю у своїй практичній діяльності доводиться вивчати залежності між різними параметрами досліджуваних об'єктів, процесів і систем.

Наприклад: залежність числа обертів двигуна від навантаження, тобто  $n=f(M_{кр})$ ; залежність сили різання при обробці деталі на металорізальному верстаті від глибини різання, тобто  $P=f(t)$ , і т.д.

Із усіх способів завдання залежностей найбільш зручним є аналітичний спосіб у вигляді функції  $n=f(M_{кр})$ ,  $P=f(t)$ ,  $y=f(t)$ .

Однак на практиці фахівець найчастіше одержує значення експериментальних залежностей між досліджуваними параметрами. У цьому випадку ставиться натурний експеримент, змінюються значення параметрів на вході системи, вимірюються значення параметрів на виході системи. Результати вимірів заносяться в таблицю.

Таким чином, у результаті проведення натурального експерименту одержуємо так звану табличну функцію.

Далі з цією табличною функцією необхідно вести науково-дослідні розрахунки. Наприклад, необхідно проінтегрувати або продиференціювати табличну функцію і т. д.

Розглянемо два типи задач з обробки дослідних даних:

- 1) задачі інтерполяції;
- 2) задачі апроксимації.

## 7.1. Інтерполяція функцій

Дано табличну функцію, тобто дано таблицю, у якій для деяких дискретних значень аргументу  $x_i$ , розташованих у порядку зростання, задано відповідні значення функції  $y_i$ :

<b>i</b>	<b>x</b>	<b>y</b>
0	$x_0$	$y_0$
1	$x_1$	$y_1$
2	$x_2$	$y_2$
...	...	...
<b>i</b>	$x_i$	$y_i$
...	...	...
<b>n</b>	$x_n$	$y_n$

$$y_i = f(x), i = \overline{0, n}. \quad (7.1)$$

Точки з координатами  $(x_i, y_i)$  називаються вузловими точками або вузлами.

Кількість вузлів у табличній функції:  $N=n+1$ .

На графіку таблична функція подається у вигляді сукупності вузлових точок (рис. 7.1).

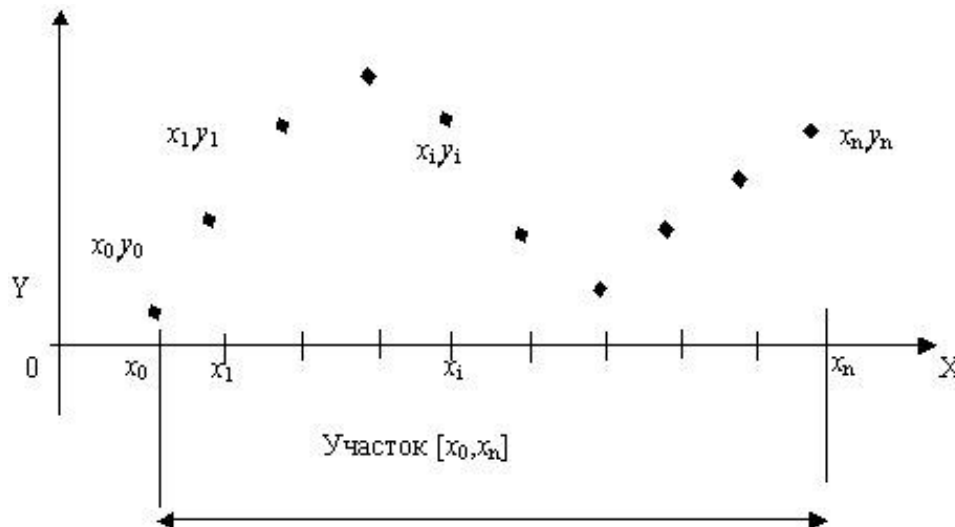


Рис. 7.1

Довжина інтервалу  $[x_0, x_n]$  дорівнює  $(x_n - x_0)$ .

У розрахунковій практиці інженера часто виникають задачі знайти значення функції для аргументів, які відсутні в таблиці. Такі задачі називаються задачами інтерполяції або екстраполювання.

Задача *інтерполяції функції* (або задача інтерполяції) полягає в тому, щоб знайти значення  $y_k$  табличної функції в будь-якій проміжній точці  $x_k$ , розташованій усередині інтервалу  $[x_0, x_n]$ , тобто

$$x_1 < x_k < x_{i+1}, x_k \in [x_0, x_n].$$

Задача екстраполювання функції (або задача *екстраполяції*) полягає в тому, щоб знайти значення  $y_l$  табличної функції в точці  $x_l$ , яка не входить в інтервал  $[x_0, x_n]$ , тобто

$$x_i < x_0; x_i > x_n.$$

Такі задачі часто називають задачами прогнозу.

Обидві ці задачі розв'язуються за допомогою знаходження аналітичного виразу деякої допоміжної функції  $F(x)$ , яка наближала б задану табличну функцію, тобто у вузлових точках виконувалася умова

$$F(x_i) = y_i; i = \overline{0, n}.$$

Для визначеності задачі шукану функцію  $F(x)$  будемо шукати із класу алгебраїчних многочленів:

$$P_n(x) = a_0 x^n + a_1 x^{n-1} + a_2 x^{n-2} + \dots + a_{n-1} x^1 + a_n x^0. \quad (7.2)$$

Цей многочлен повинен проходити через усі вузлові точки, тобто

$$P_n(x_i) = y_i. \quad (7.3)$$

Тому ступінь многочлена  $n$  залежить від кількості вузлових точок  $N$  і дорівнює кількості вузлових точок мінус один, тобто  $n=N-1$ .

Многочлен вигляду формули (7.2), який проходить через усі вузлові точки табличної функції, називається інтерполяційним многочленом.

Інтерполяція за допомогою алгебраїчних многочленів називається параболічною інтерполяцією.

Таким чином, для розв'язання задачі інтерполяції насамперед необхідно для функції  $F(x_i) = y_i; i = \overline{0, n}$ , заданої у табличному вигляді, побудувати інтерполяційний многочлен степеня  $n$ , який проходить через усі вузлові точки таблиці:

$$P_n(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_{n-1}x^1 + a_nx^0$$

де  $n$  - ступінь многочлена, що дорівнює кількості вузлових точок  $N$  мінус один, тобто  $n=N-1$ .

У результаті в будь-якій іншій проміжній точці  $x_k$ , розташованій усередині відрізка  $[x_0, x_n]$ , виконується наближена рівність  $P_n(x_k) = f(x_k) = y_k$  (рис. 7.2).

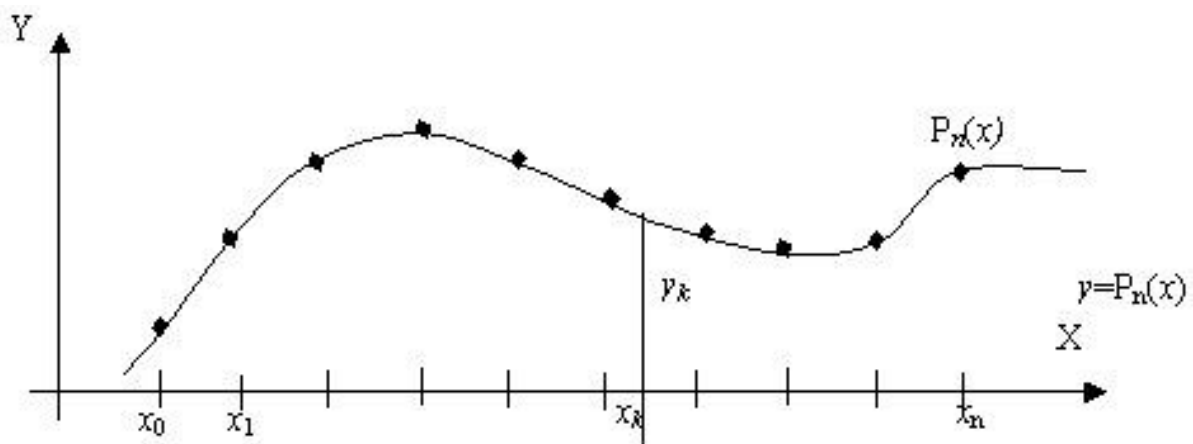


Рис. 7.2.



## 7.2. Побудова інтерполяційного многочлена в явному вигляді

Для побудови інтерполяційного многочлена вигляду формули (7.2) необхідно визначити його коефіцієнти  $a_0, a_1, \dots, a_n$ , тобто  $a_i$   $i=0,1,2,\dots,n$ . Кількість невідомих коефіцієнтів дорівнює  $n+1=N$ , де  $n$  - ступінь многочлена формули (7.2),  $N$  - кількість вузлових точок табличної функції (7.1).

Для знаходження коефіцієнтів, використовуємо властивість (7.3) інтерполяційного многочлена формули (7.2). На підставі цієї властивості інтерполяційний многочлен повинен пройти через кожен вузлову точку  $(x_i, y_i)$  таблиці (7.1), тобто

$$a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_{n-1}x^1 + a_nx^0 = y_i; i = \overline{0, n}. \quad (7.4)$$

Підставляючи у вираз (7.4) кожен вузлову точку таблиці (7.1) одержуємо систему лінійних рівнянь:

$$\begin{cases} a_0x_0^n + a_1x_0^{n-1} + \dots + a_{n-1}x_0 + a_n = y_0, \\ a_0x_1^n + a_1x_1^{n-1} + \dots + a_{n-1}x_1 + a_n = y_1, \\ \dots\dots\dots \\ a_0x_n^n + a_1x_n^{n-1} + \dots + a_{n-1}x_n + a_n = y_n. \end{cases} \quad (7.5)$$

Невідомими системи (7.5) є  $a_0, a_1, a_2, \dots, a_n$ , тобто коефіцієнти многочлена формули (7.2). Коефіцієнти при невідомих системи (7.5)

$$x_i^n, x_i^{n-1}, \dots, x_i^0, i = 0, 1, \dots, n, i = 0, 1, \dots, n$$

легко можуть бути визначені на підставі даних таблиці (7.1).

### 7.3. Інтерполяція за Лагранжем

Інтерполяційний многочлен може бути побудований за допомогою спеціальних інтерполяційних формул Лагранжа, Ньютона, Стерлінга, Бесселя й ін.

Інтерполяційний многочлен за формулою Лагранжа має вигляд

$$\begin{aligned}
 L_n(x) = & \frac{(x-x_1)(x-x_2)(x-x_3)\dots(x-x_n)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)\dots(x_0-x_n)} \cdot y_0 + \\
 & + \frac{(x-x_0)(x-x_2)(x-x_3)\dots(x-x_n)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)\dots(x_1-x_n)} \cdot y_1 + \\
 & + \frac{(x-x_0)(x-x_1)(x-x_3)\dots(x-x_n)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)\dots(x_2-x_n)} \cdot y_2 + \dots \\
 & + \frac{(x-x_0)(x-x_1)(x-x_2)\dots(x-x_{n-1})}{(x_n-x_0)(x_n-x_1)(x_n-x_2)\dots(x_n-x_{n-1})} \cdot y_n.
 \end{aligned} \tag{7.6}$$

Легко довести, що многочлен Лагранжа є інтерполяційним многочленом, що проходить через усі вузлові точки, тобто у вузлах інтерполяції  $x_i$  виконується умова  $L_n(x_i) = y_i$ , і ми можемо використовувати його як допоміжну функцію для розв'язання задач інтерполяції, тобто

$$L_n(x_k) \approx y_k.$$

Чим більше вузлів інтерполяції на відріжку  $[x_0, x_n]$ , тим точніше інтерполяційний многочлен наближає задану табличну функцію (7.1), тобто тим точніше рівність

$$f(x_k) \approx L_n(x_k).$$

Однак зі збільшенням кількості вузлів інтерполяції зростає степінь інтерполяційного многочлена  $n$  і в результаті значно зростає обсяг обчислювальної роботи. Тому при великій кількості вузлів необхідно застосовувати ЕОМ. У цьому випадку зручно знаходити значення функції в проміжних точках, не одержуючи

многочлен у явному вигляді.

При розв'язанні задачі екстраполяції функції за допомогою інтерполяційного многочлена обчислення значення функції за межами відрізка  $[x_0, x_n]$  звичайно проводяться не далі, ніж на один крок  $h$ , який дорівнює найменшій величині  $|x_{i+1} - x_i|$ , тому що за межами відрізка  $[x_0, x_n]$  похибки, як правило, збільшуються.

#### 7.4. Програмування формули Лагранжа

Згорнемо формулу Лагранжа (7.6). У результаті одержимо

$$Ln(x) = \sum_{j=0}^n B_j \cdot y_j,$$

де  $B_j = \prod_{i=0, i \neq j}^n \frac{x - x_i}{x_j - x_i},$

але при цьому обов'язковим є виконання умови  $i \neq j$ .

При побудові алгоритму використовують конструкцію з двох вкладених циклів.

Зовнішнім циклом накопичуємо суму  $L = \sum_{j=0}^n B_j \cdot y_j$ .

Внутрішнім циклом накопичуємо добуток  $B_j = \prod_{i=0, i \neq j}^n \frac{x - x_i}{x_j - x_i}, i \neq j$ .

Алгоритм (рис. 7.3) не передбачає одержання інтерполяційного многочлена в явному вигляді, а відразу розв'язує задачі інтерполяції функції в заданій точці,  $x=D$ .

Позначення в алгоритмі:

$n$  - степінь інтерполяційного многочлена Лагранжа (7.6), дорівнює кількості вузлових точок  $N$  мінус один, тобто  $n=N-1$ ;

$D$  - значення аргументу в точці, для якої розв'язується задача інтерполяції табличної функції (7.1);

L - значення многочлена формули (7.6).

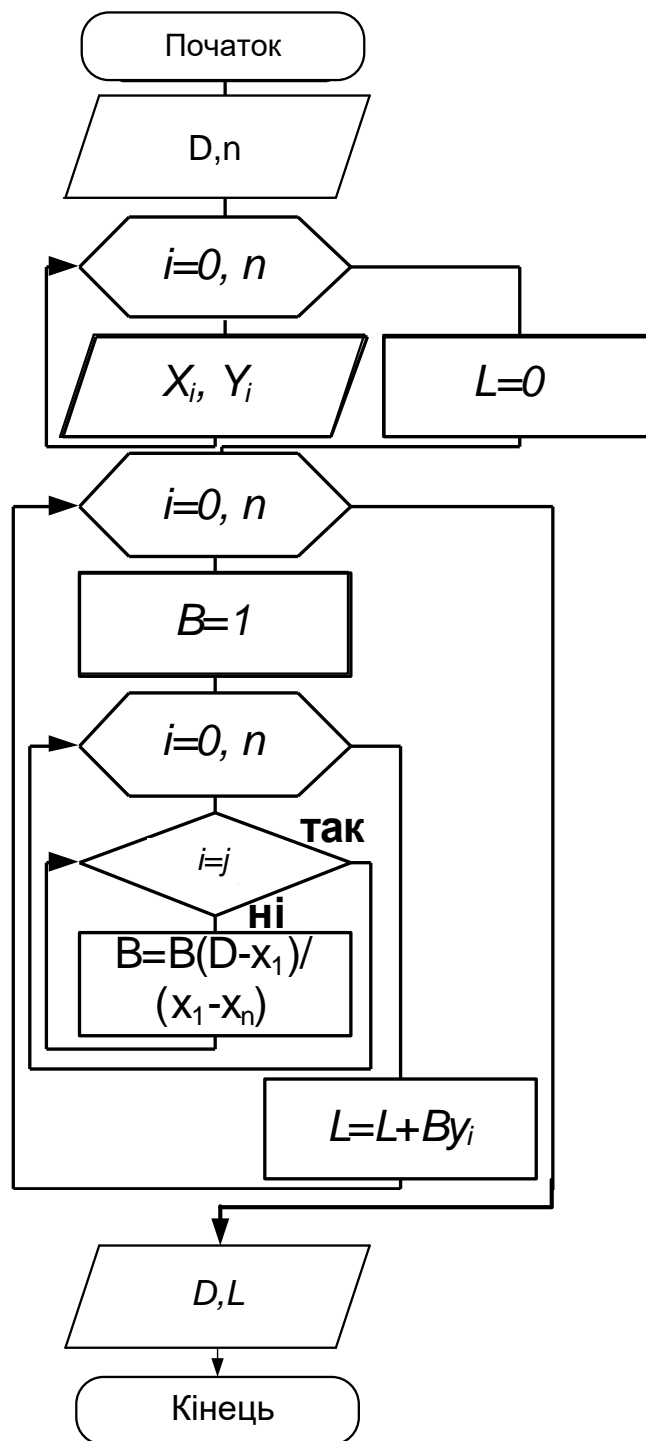


Рис. 7.3. Схема алгоритму інтерполяції за Лагранжем

## 7.5. Інтерполяція за Ньютоном

Дано табличну функцію:

<b>I</b>	<b>x<sub>i</sub></b>	<b>y<sub>i</sub></b>
0	x <sub>0</sub>	y <sub>0</sub>
1	x <sub>1</sub>	y <sub>1</sub>
2	x <sub>2</sub>	y <sub>2</sub>
...	...	...
N	x <sub>n</sub>	y <sub>n</sub>

або

$$y_i = f(x_i), i = \overline{0, n}.$$

Точки з координатами (x<sub>i</sub>, y<sub>i</sub>) називаються вузловими точками або вузлами.

Кількість вузлів у табличній функції N=n+1.

Необхідно знайти значення цієї функції в проміжній точці, наприклад x=D, причому  $D \in [x_0, x_n]$ .

Для розв'язання задачі будемо інтерполяційний многочлен.

Інтерполяційний многочлен за формулою Ньютона має вигляд

$$\begin{aligned}
 L_n(x) = & f(x_0) + (x-x_0) \cdot f(x_0; x_1) + \\
 & + (x-x_0) \cdot (x-x_1) \cdot f(x_0; x_1; x_2) + \\
 & + (x-x_0) \cdot (x-x_1) \cdot (x-x_2) \cdot f(x_0; x_1; x_2; x_3) + \dots + \\
 & + (x-x_0) \cdot (x-x_1) \cdot \dots \cdot (x-x_{n-1}) \cdot f(x_0; x_1; \dots; x_n),
 \end{aligned}
 \tag{7.7}$$

де n - ступінь многочлена,

$f(x_0), f(x_0; x_1), f(x_0; x_1; x_2), f(x_0; x_1; \dots; x_n)$  - розділені різниці 0-го, 1-го, 2-го, ..., n-го порядку відповідно.

## Розділені різниці

Значення  $f(x_0), f(x_1), \dots, f(x_n)$ , тобто значення табличної функції у вузлах, називаються розділеними різницями нульового порядку ( $k=0$ ).

Відношення  $f(x_0; x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$  називається розділеною різницею першого порядку ( $k=1$ ) на інтервалі  $[x_0, x_1]$  і дорівнює різниці розділених різниць нульового порядку на кінцях інтервалу  $[x_0, x_1]$ , розділеній на довжину цього інтервалу.

Для довільного інтервалу  $[x_i, x_{i+1}]$  розділена різниця першого порядку ( $k=1$ ) дорівнює  $f(x_i; x_{i+1}) = \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i}$ .

Відношення  $f(x_0; x_1; x_2) = \frac{f(x_1; x_2) - f(x_0; x_1)}{x_2 - x_0}$  називається розділеною різницею другого порядку ( $k=2$ ) на інтервалі  $[x_0, x_2]$  і дорівнює різниці розділених різниць першого порядку, розділеній на довжину ділянки  $[x_0, x_2]$ .

Для довільного інтервалу  $[x_i, x_{i+2}]$  розділена різниця другого порядку ( $k=2$ ) дорівнює

$$f(x_i; x_{i+1}; x_{i+2}) = \frac{f(x_{i+1}; x_{i+2}) - f(x_i; x_{i+1})}{x_{i+2} - x_i}. \quad (7.8)$$

Таким чином, розділена різниця  $k$ -го порядку на інтервалі  $[x_i, x_{i+k}]$  може бути визначена через *розділені різниці*  $(k-1)$ -го порядку за рекурентною формулою.

Максимальне значення  $k$  дорівнює  $n$ . Тоді  $i=0$  і розділена різниця  $n$ -го порядку на інтервалі  $[x_0, x_n]$  дорівнює

$$f(x_0; x_1; \dots; x_n) = \frac{f(x_1; x_2; \dots; x_n) - f(x_0; x_1; \dots; x_{n-1})}{x_n - x_0},$$

тобто дорівнює різниці розділених різниць  $(n-1)$ -го порядку, розділеній на довжину інтервалу  $[x_0, x_n]$ .

Розділені різниці  $f(x_0; x_1), f(x_0; x_1; x_2), \dots, f(x_0; x_1; \dots; x_n)$  є цілком визначеними числами, тому вираз (7.7) дійсно є алгебраїчним многочленом  $n$ -го ступеня. При цьому в многочлені виразу (7.7) усі розділені різниці визначені для інтервалів  $[x_0, x_{0+k}]$ ,  $k=1, \dots, n$ .

**Лема:** алгебраїчний многочлен вираз (7.7), побудований за формулами Ньютона, дійсно є інтерполяційним многочленом, тобто значення многочлена у вузлових точках дорівнює значенню табличної функції

$$L_n(x_i) = f(x_i) = y_i; i = 0, 1, \dots, n.$$

Це легко довести підрахунком значень відповідного многочлена у вузлових точках.

Зазначимо, що розв'язання задачі *інтерполяції за Ньютоном* має деякі переваги порівняно з розв'язання задачі *інтерполяції за Лагранжем*. Кожний доданок інтерполяційного многочлена Лагранжа залежить від усіх значень табличної функції  $y_i$ ,  $i=0, 1, \dots, n$ . Тому при зміні кількості вузлових точок  $N$  і ступеня многочлена  $n$  ( $n=N-1$ ) інтерполяційний многочлен Лагранжа потрібно будувати заново. У многочлені Ньютона при зміні кількості вузлових точок  $N$  і ступеня многочлена  $n$  потрібно тільки додати або відкинути відповідне число стандартних доданків у формулі Ньютона (7.7). Це зручно на практиці й прискорює процес обчислень.

## 7.6. Програмування формули Ньютона

Для побудови многочлена Ньютона за формулою (7.7) організуємо циклічний обчислювальний процес по  $k = \overline{1, n}$ . При цьому на кожному кроці пошуку знаходимо *розділені різниці*  $k$ -го порядку. Будемо поміщати *розділені різниці* на кожному кроці в масив  $Y$ .

Тоді рекурентна формула (7.8) буде мати вигляд

$$y_i = \frac{y_{i+1} - y_i}{x_{i+k} - x_i}$$

$$k = \overline{1, n}$$

$$i = \overline{0, n - k}$$
(7.9)

У формулі Ньютона (7.7) використовуються *розділені різниці*  $k$ -го порядку, підраховані тільки для інтервалів  $[x_0, x_{0+k}]$ , тобто *розділені різниці*  $k$ -го порядку для  $i=0$ . Позначимо ці *розділені різниці*  $k$ -го порядку як  $y_0$ . А розділені різниці, підраховані для  $i > 0$ , використовуються для розрахунків розділених різниць більш високих порядків.

Використовуючи вираз (7.9), згорнемо формулу (7.7). У результаті одержимо

$$L_n(x) = y_0 + \sum_{k=1}^n P \cdot y_0^*$$

де  $y_0$  – значення табличної функції (7.1) для  $x=x_0$ ;

$y_0^*$  – розділена різниця  $k$ -го порядку для інтервалу  $[x_0, x_{0+k}]$ .

$$P = (x-x_0)(x-x_1) \dots (x-x_{k-1}) = \prod_{j=0}^{k-1} (x-x_j).$$

Для обчислення  $P$  усередині циклу по  $k$  зручно використовувати рекурентну формулу  $P = P^*(x - x_{k-1})$ .

Схема алгоритму *інтерполяції за Ньютоном* наведена на рис. 7.4.



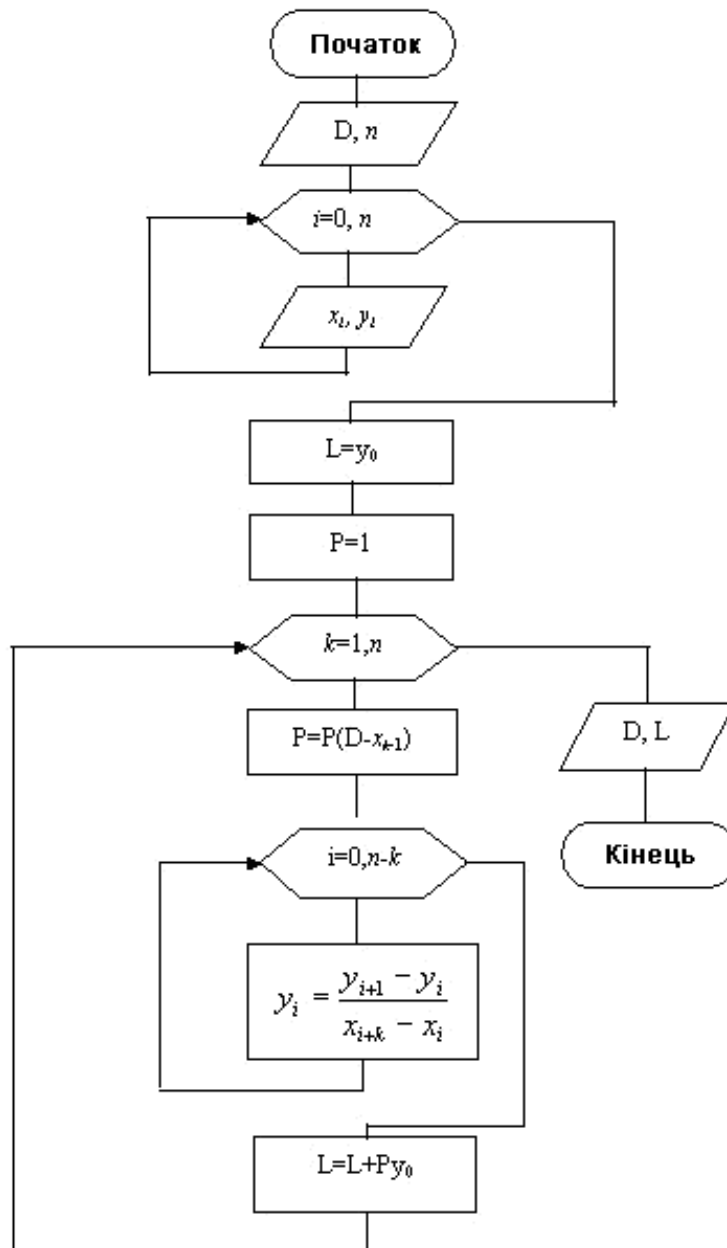


Рис. 7.4. Схема алгоритму інтерполяції за Ньютоном

### 7.7. Приклад інтерполяції за Ньютоном

Дана таблична функція:

<b>i</b>	<b>x<sub>i</sub></b>	<b>y<sub>i</sub></b>
0	2	0,693147
1	3	1,098613
2	4	1,986295

Обчислити розділені різниці 1-го, 2-го, 3-го порядків (n=3) і занести їх у діагональну таблицю.

Розділені різниці першого порядку:

$$f(x_0; x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{1,098613 - 0,693147}{3 - 2} = 0,405466.$$

$$f(x_1; x_2) = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{1,386295 - 1,098613}{4 - 3} = 0,287682.$$

$$f(x_2; x_3) = \frac{f(x_3) - f(x_2)}{x_3 - x_2} = \frac{1,609438 - 1,386295}{5 - 4} = 0,223143.$$

Розділені різниці другого порядку:

$$f(x_0; x_1; x_2) = \frac{f(x_1; x_2) - f(x_0; x_1)}{x_2 - x_0} = \frac{0,287682 - 0,405466}{4 - 2} = -0,058892.$$

$$f(x_1; x_2; x_3) = \frac{f(x_2; x_3) - f(x_1; x_2)}{x_3 - x_1} = \frac{0,223143 - 0,287682}{5 - 3} = -0,0322695.$$

Розділена різниця третього порядку:

$$f(x_0; x_1; x_2; x_3) = \frac{f(x_1; x_2; x_3) - f(x_0; x_1; x_2)}{x_3 - x_0} = \frac{-0,0322695 - (-0,058892)}{5 - 2} = 0,00887416$$

Таблиця 7.1

Діагональна таблиця розділених різниць

I	x <sub>i</sub>	Розділена різниця			
		0-го порядку	1-го порядку	2-го порядку	3-го порядку
0	2	0,693147			
			0,405466		
1	3	1,098613		-0,058892	
			0,287682		0,00887416
2	4	1,386295		-0,0322695	
			0,223143		
3	5	1,60943			

Інтерполяційний многочлен Ньютона для заданої табличної функції має вигляд

$$L_3 = f(x_0) + (x - x_0) * f(x_0; x_1) + (x - x_0)(x - x_1) * f(x_0; x_1; x_2) + (x - x_0)(x - x_1)(x - x_2) * f(x_0; x_1; x_2; x_3) = 0,693147 + (x - 2) * 0,405466 + (x - 2)(x - 3) * (-0,058892) + (x - 2)(x - 3)(x - 4) * 0,0887416.$$

Далі отриманий інтерполяційний многочлен Ньютона можна привести до нормального вигляду

$$L_3(x) = a_0 x^3 + a_1 x^2 + a_2 x + a_3$$

і використовувати його для розв'язання задач інтерполяції або прогнозу.

## 7.8. Сплайн-інтерполяція

Сплайни стали широко використовуватися в обчислювальній математиці порівняно недавно. У машинобудівному кресленні вони застосовуються вже давно, тому що сплайни - це лекала або гнучкі лінійки, деформація яких дозволяє провести криву через задані точки  $(x_i, y_i)$ .

Використовуючи теорію вигину бруса при малих деформаціях, можна показати, що сплайн - це група кубічних многочленів, у місцях сполучення яких перша й друга похідні безперервні. Такі функції називаються кубічними сплайнами. Для їхньої побудови необхідно задати коефіцієнти, які єдиним способом визначають многочлен у проміжку між даними точками.

Наприклад, для деяких функцій (рис. 7.5) необхідно задати всі кубічні функції  $q_1(x), q_2(x), \dots, q_n(x)$ .

У найбільш загальному випадку ці многочлени мають вигляд

$$q_i(x) = k_{1i} + k_{2i}x + k_{3i}x^2 + k_{4i}x^3; i = \overline{1, n},$$

де  $k_{ij}$  - коефіцієнти, обумовлені описаними раніше умовами, кількість яких дорівнює  $4n$ . Для визначення коефіцієнтів  $k_{ij}$  необхідно побудувати й розв'язати систему порядку  $4n$ .

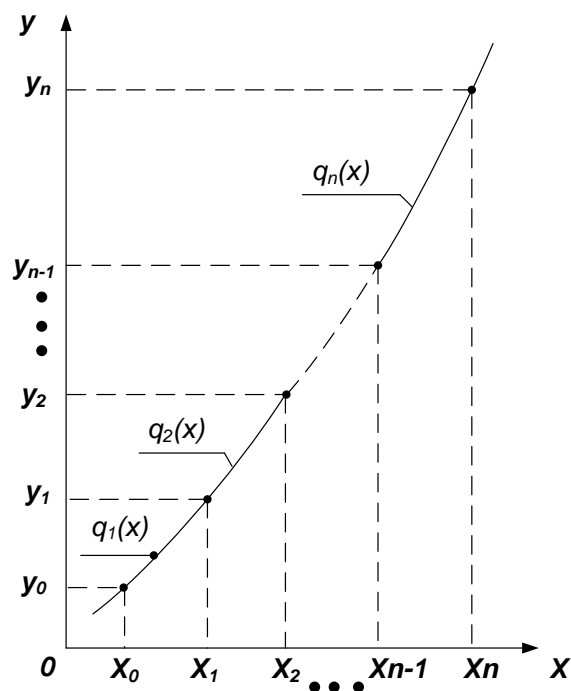


Рис. 7.5

Перші  $2n$  умов вимагають, щоб сплайни стикалися в заданих точках:

$$\begin{aligned} q_i(x_i) &= y_i, \quad i = \overline{1, n}; \\ q_{i+1}(x_i) &= y_i, \quad i = \overline{0, n-1}. \end{aligned}$$

Наступні  $(2n-2)$  умов вимагають, щоб у місцях зіткнення сплайнів були однакові перші й другі похідні:

$$\begin{aligned} q'_{i+1}(x_i) &= q'_i(x_i), \quad i = \overline{1, n-1}; \\ q''_{i+1}(x_i) &= q''_i(x_i), \quad i = \overline{0, n-1}. \end{aligned}$$

Система алгебраїчних рівнянь має розв'язок, якщо кількість рівнянь відповідає кількості невідомих. Для цього необхідно ввести ще два рівняння. Звичайно використовуються такі умови:

$$q''_1(x_0) = 0, \dots, q''_n(x_n) = 0.$$

При побудові алгоритму методу перші й другі похідні зручно апроксимувати розділеними різницями відповідних порядків.

Отриманий у такий спосіб сплайн називається природним кубічним сплайном. Знайшовши коефіцієнти сплайну, використовують цю кусочно-гладку поліноміальну функцію для зображення даних при інтерполяції.

### 7.9. Апроксимація дослідних даних

У результаті проведення натурального експерименту отримана таблична функція (табл. 7.2).

Таблиця 7.2

i	X	Y
0	$x_0$	$y_0$
1	$x_1$	$y_1$
2	$x_2$	$y_2$
3	$x_3$	$y_3$
...	...	...
n	$x_n$	$y_n$

Тут N - кількість вузлових точок у таблиці,  $n=N-1$ .

Задача апроксимації полягає у відшуванні аналітичної залежності  $y=f(x)$  отриманої табличної функції.

Існує 2 способи *апроксимації дослідних даних*.

*Перший спосіб.* Цей спосіб вимагає, щоб апроксимуюча крива  $F(x)$ , аналітичний вигляд якої необхідно знайти, проходила через усі вузлові точки таблиці. Цю задачу можна розв'язати за допомогою *побудови інтерполяційного многочлена степеня n*:

$$P_n(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_{n-1}x^1 + a_nx^0. \quad (7.10)$$

Однак цей спосіб *апроксимації дослідних даних* має недоліки:

1. Точність апроксимації гарантується в невеликому інтервалі  $[x_0, x_n]$  при кількості вузлових точок не більше 7-8.

2. Значення табличної функції у вузлових точках повинні бути задані з великою точністю.

Відомо, що як би точно не проводився експеримент, результати експерименту містять похибки. Справа в тому, що насправді досліджувана величина залежить не тільки від одного аргументу  $X$ , але й від інших випадкових факторів, які від дослідження до дослідження коливаються за своїми власними випадковими законами. Цим самим обумовлюється випадкове коливання досліджуваної функції.

У результаті апроксимувати дослідні дані за допомогою інтерполяційного многочлена, який проходив би через усі вузлові точки таблиці, не завжди вдається. Більш того, прагнучи пройти через усі вузлові точки таблиці й збільшуючи порядок многочлена, ми тим самим починаємо відтворювати не тільки закономірні зміни функції, але і її випадкові перешкоди.

*Другий спосіб.* На практиці знайшов застосування інший спосіб *апроксимації дослідних даних* - згладжування дослідних даних. Сутність цього методу полягає в тому, що табличні дані апроксимуються кривою  $F(x)$ , яка не обов'язково повинна пройти через усі вузлові точки, а повинна ніби згладити всі випадкові перешкоди табличної функції.

### **7.9.1. Згладжування дослідних даних методом найменших квадратів**

У цьому методі при згладжуванні дослідних даних апроксимуючу криву  $F(x)$  прагнуть провести так, щоб її відхилення від табличних даних по всіх вузлових точках були мінімальними (рис. 7.6).

$$\varepsilon_i = |F(x_i) - y_i| \rightarrow \min. \quad (7.11)$$

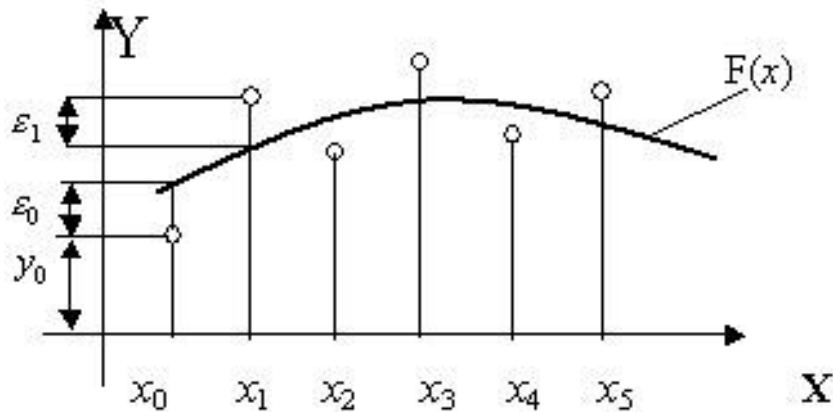


Рис. 7.6

Позбудемося знака відхилення. Тоді умова (7.11) буде мати вигляд

$$\varepsilon^2 = |F(x_i) - y_i|^2 \rightarrow \min . \quad (7.12)$$

Суть методу найменших квадратів полягає в такому: для табличних даних, отриманих у результаті експерименту, відшукати аналітичну залежність  $F(x)$ , сума квадратів відхилень якої від табличних даних по всіх вузлових точках була б мінімальною, тобто

$$\sum_{i=0}^n \varepsilon_i^2 = \sum_{i=0}^n (F(x_i) - y_i)^2 \rightarrow \min \quad (7.13)$$

Для визначеності задачі функцію  $F(x)$  будемо вибирати з класу алгебраїчних многочленів степеня  $m$ :

$$P_m(x) = a_0x^m + a_1x^{m-1} + a_2x^{m-2} + \dots + a_{m-1}x^1 + a_m . \quad (7.14)$$

Назвемо многочлен (7.14) апроксимуючим многочленом. Апроксимуючий многочлен не проходить через усі вузлові точки таблиці. Тому його степінь  $m$  не залежить від числа вузлових точок. При цьому завжди  $m < n$ . Степінь  $m$  може змінюватися в межах  $1 \leq m \leq N - 2$ .

Якщо  $m=1$ , то ми апроксимуємо табличну функцію прямою лінією. Такі задачі називаються лінійною регресією.

Якщо  $m=2$ , то ми апроксимуємо табличну функцію квадратичною параболою. Такі задачі називаються квадратичною апроксимацією.

Якщо  $m=3$ , то ми апроксимуємо табличну функцію кубічною параболою. Такі задачі називаються кубічною апроксимацією.

Уточнимо метод найменших квадратів: для табличної функції, отриманої в результаті експерименту, побудувати апроксимуючий многочлен степеня  $m$ , для якого сума квадратів відхилень по всіх вузлових точках мінімальна, тобто

$$S = \sum_0^n (P_m(x_i) - y_i)^2 \rightarrow \min . \quad (7.15)$$

Змінимо вигляд многочлена  $P_m$ . Поставимо на останнє місце складові елементи, що містять  $x^m$  на передостаннє – ті, що містять  $x^{m-1}$  і т. д. У результаті одержимо

$$P_m(x) = a_0x^0 + a_1x^1 + a_2x^2 + \dots + a_mx^m \quad (7.16)$$

або

$$P_m(x) = \sum_{j=0}^m a_j x^j .$$

При цьому змінимо індекси коефіцієнтів многочлена. Тоді умова (7.13) буде мати вигляд:

$$S = \sum_{i=0}^n (a_0x_i^0 + a_1x_i^1 + a_2x_i^2 + \dots + a_mx_i^m - y_i)^2 \rightarrow \min , \quad (7.17)$$

де  $x_i$  та  $y_i$  – координати вузлових точок таблиці;

$a_j, j = \overline{0, m}$  – невідомі коефіцієнти многочлена (7.11).

Необхідною умовою існування мінімуму функції  $S$  є рівність нулю її часткових похідних по кожній  $a_j$ .



$$\left\{ \begin{array}{l} \frac{\delta S}{\delta a_0} \rightarrow 2 \sum_{i=0}^n ((a_0 x_i^0 + a_1 x_i^1 + \dots + a_m x_i^m - y_i) x_i^0) = 0, \\ \frac{\delta S}{\delta a_1} \rightarrow 2 \sum_{i=0}^n ((a_0 x_i^0 + a_1 x_i^1 + \dots + a_m x_i^m - y_i) x_i^1) = 0, \\ \frac{\delta S}{\delta a_2} \rightarrow 2 \sum_{i=0}^n ((a_0 x_i^0 + a_1 x_i^1 + \dots + a_m x_i^m - y_i) x_i^2) = 0, \\ \dots\dots\dots \\ \frac{\delta S}{\delta a_m} \rightarrow 2 \sum_{i=0}^n ((a_0 x_i^0 + a_1 x_i^1 + \dots + a_m x_i^m - y_i) x_i^m) = 0. \end{array} \right.$$

У результаті одержали систему лінійних рівнянь. Розкриваючи дужки й переносячи вільні члени в правій частині рівнянь, одержимо в нормальній формі систему лінійних рівнянь:

$$\left\{ \begin{array}{l} c_0 a_0 + c_1 a_1 + c_2 a_2 + \dots + c_m a_m = d_0, \\ c_1 a_0 + c_2 a_1 + c_3 a_2 + \dots + c_{m+1} a_m = d_1, \\ c_2 a_0 + c_3 a_1 + c_4 a_2 + \dots + c_{m+2} a_m = d_0, \\ \dots\dots\dots \\ c_m a_0 + c_{m+1} a_1 + c_{m+2} a_2 + \dots + c_{2m} a_m = d_m, \end{array} \right. \quad (7.18)$$

де  $a_j$  - невідомі системи лінійних рівнянь (7.18);

$c_k = \sum_{i=0}^n x_i^k, k = \overline{0, 2m}$  - коефіцієнти системи лінійних рівнянь (7.18);

$d_j = \sum_{i=0}^n y_i x_i^j, j = \overline{0, m}$  - вільні члени системи лінійних рівнянь (7.18).

Порядок системи дорівнює  $m+1$ .

### 7.9.2. Програмування методу найменших квадратів

Змінимо індексацію в системі (7.18). У результаті одержимо:

$$\begin{cases} c_{11}a_1 + c_{12}a_2 + c_{13}a_3 + \dots + c_{1(m+1)}a_{m+1} = d_1, \\ c_{21}a_1 + c_{22}a_2 + c_{23}a_3 + \dots + c_{2(m+1)}a_{m+1} = d_2, \\ c_{31}a_1 + c_{32}a_2 + c_{33}a_3 + \dots + c_{3(m+1)}a_{m+1} = d_3, \\ \dots\dots\dots \\ c_{(m+1),1}a_1 + c_{(m+1),2}a_2 + \dots + c_{(m+1),(m+1)}a_{m+1} = d_{m+1}, \end{cases} \quad (7.19)$$

де  $a_j, j = \overline{1, (m+1)}$  - невідомі системи лінійних рівнянь (7.19);

$c_{k,j} = \sum_{i=1}^N x_i^{k+j-2}, k = \overline{1, (m+1)}, j = \overline{1, (m+1)}$  - коефіцієнти системи лінійних рівнянь (7.19);

$d_k = \sum_{i=1}^N y_i x_i^{j-1}, j = \overline{1, (m+1)}$  - вільні члени системи лінійних рівнянь (7.19);

$(x_i, y_i)$  - координати вузлових точок табличної функції,  $i = \overline{1, n}$ ;

$N$  - кількість вузлових точок;

$m$  - ступінь апроксимуючого многочлена вигляду

$$P_n(x) = a_1x^0 + a_2x^1 + a_3x^2 + \dots + a_{m+1}x^m. \quad 0) \quad (7.2)$$

### 7.9.3. Алгоритм розв'язання задачі

1. Будуємо систему лінійних рівнянь (7.19). Визначаємо коефіцієнти  $c_{k,j}$  і вільні члени  $d_k$ . Оскільки система (7.19) симетрична відносно головної діагоналі, то достатньо визначити тільки наддіагональні елементи системи.

2. Розв'язуємо систему (7.19) методом Гаусса. Знаходимо коефіцієнти  $a_j$  многочлена (7.20).

3. Будуємо апроксимуючий многочлен (7.20) і визначаємо його значення в кожній вузловій точці  $P_i = P_m(x_i)$ .

4. Знаходимо відхилення кожної вузлової точки  $\varepsilon_i = P_i - y_i$ .

5. Знаходимо суму квадратів відхилень по всіх вузлових точках  $S = \sum_{i=1}^N \varepsilon_i^2$ .

6. Знаходимо залишкову дисперсію  $D = \frac{S}{N - (m + 1)}$ .

Для побудови апроксимуючого многочлена (7.16) і обчислення його значення в кожній вузловій точці використовуємо раціональну форму многочлена:

$$P_m(x) = a_1 + x \cdot (a_2 + x \cdot (a_3 + \dots + x \cdot (a_m + x \cdot a_{(m+1)} \dots))) \quad (7.21)$$

Тоді для обчислення значення многочлена (7.21) зручно користуватися *схемою Горнера*. Рекурентна формула за схемою Горнера має вигляд

$$P = a_{m+1}$$
$$P = a_j + x_i P; j = \overline{m+1, 1}; i = \overline{1, n}$$

Укрупнена схема алгоритму наведена на рис. 7.7.

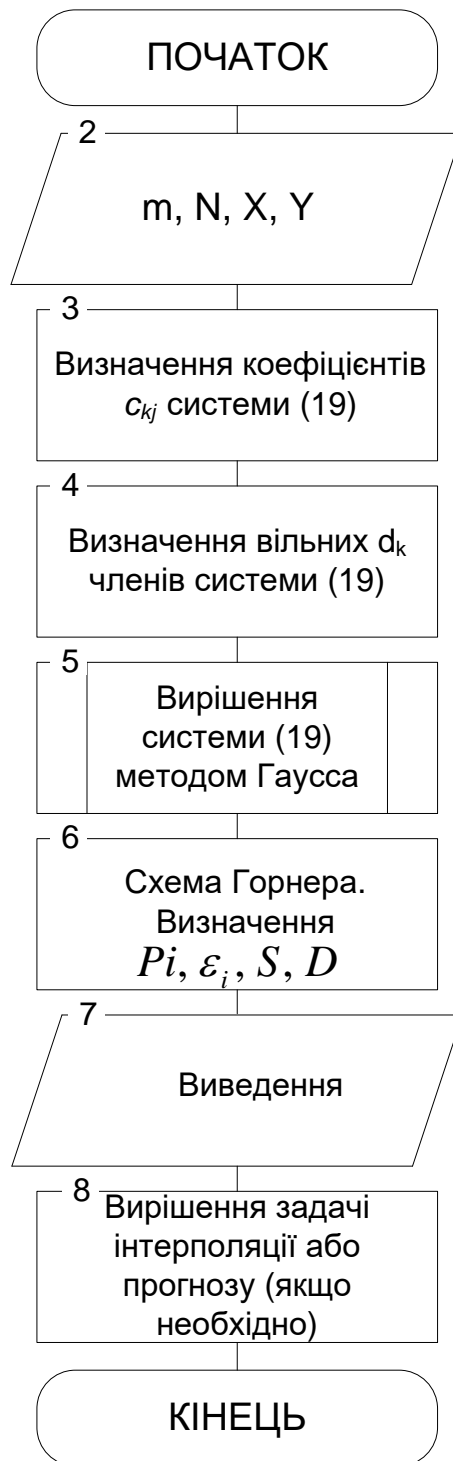


Рис. 7.7. Укрупнена схема алгоритму апроксимації методом найменших квадратів

Позначення в блоці 2:  $m$  – ступінь апроксимуючого многочлена,  $N$  – кількість вузлових точок табл. 7.2,  $X$ ,  $Y$  – масиви значень  $x$  і  $y$  з табл. 7.2.

## 8. ЧИСЕЛЬНІ МЕТОДИ ІНТЕГРУВАННЯ

Знаходження визначеного інтегралу в процесі моделювання об'єктів процесів або систем може застосовуватися в таких задачах:

1. Визначення шляху при змінній швидкості:

$$S = \int_{t_0}^t V(t) dt$$

2. Знаходження швидкості при змінному прискоренні:

$$V = \int_{t_0}^t a(t) dt$$

3. Визначення моментів інерції тіл:

$$Y_x = \int x^2 dm$$

4. Знаходження роботи змінної сили:

$$A = \int_{t_0}^t F(t) dt$$

5. При розв'язанні диференціальних рівнянь.

### 8.1. Сутність чисельних методів інтегрування

Дано функцію  $y=f(x)$ . Знайти інтеграл цієї функції на інтервалі  $[a,b]$ , тобто знайти  $\int_a^b f(x) dx$ . Якщо:

- підінтегральна функція  $f(x)$  задана в аналітичному вигляді;
- функція  $f(x)$  безперервна на відрізку  $[a,b]$ ;
- відома її первісна, тобто  $F'(x) = f(x), x \in [a,b]$ .

то інтеграл може бути обчислений за формулою Ньютона-Лейбница як збільшення первісної на інтервалі  $[a,b]$ , тобто

$$\int_a^b f(x)dx = F(b) - F(a).$$

Але на практиці формула Ньютона-Лейбниці для обчислення інтеграла використовується рідко. Чисельні методи інтегрування застосовуються в таких випадках:

- підінтегральна функція  $f(x)$  задана таблично на інтервалі  $[a,b]$ ;
- підінтегральна функція  $f(x)$  задана аналітично, але її первісна не виражається через елементарні функції;
- підінтегральна функція  $f(x)$  задана аналітично, має первісну, але її визначення занадто складне.

У чисельних методах інтегрування не використовується знаходження первісної. Основу алгоритмів чисельних методів інтегрування становить геометричний зміст визначеного інтеграла. Інтеграл чисельно дорівнює площині  $S$  криволінійної трапеції, розташованої під підінтегральною кривою  $f(x)$  на інтервалі  $[a,b]$  (рис. 8.1).

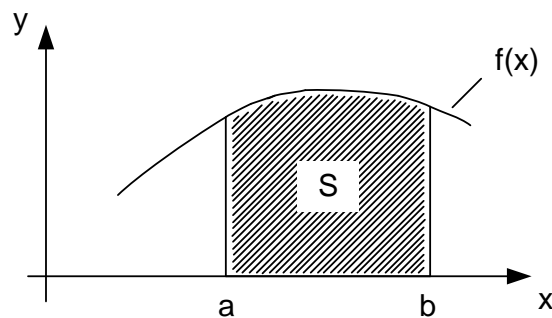


Рис. 8.1. Геометричний зміст визначеного інтеграла

Суть усіх чисельних методів інтегрування полягає в наближеному обчисленні зазначеної площі. Тому всі чисельні методи є наближеними.

При обчисленні інтеграла підінтегральна функція  $f(x)$  апроксимується інтерполяційним многочленом. На практиці, щоб не мати справи з многочленами високих ступенів, увесь відрізок

$[a,b]$  ділять на частини й інтерполяційні многочлени будують для кожної частини розподілу.

Порядок обчислення інтеграла чисельними методами такий (рис. 8.2):

1. Відрізок  $[a,b]$  ділимо на  $n$  рівних частин із кроком  $h=(b-a)/n$ .
2. У кожній частині розподілу підінтегральну функцію  $f(x)$  апроксимуємо інтерполяційним многочленом. Ступінь многочлена  $n = 0,1,2,\dots$
3. Для кожної частини розподілу визначаємо площу часткової криволінійної трапеції.
4. Підсумуємо ці площини. Наближене значення інтеграла  $I$  дорівнює сумі площин часткових трапецій

$$I = \sum_{i=0}^{n-1} S_i .$$

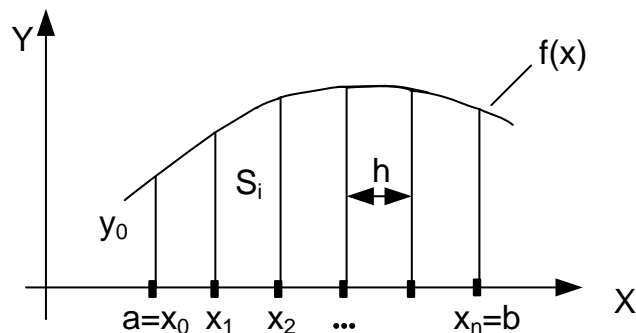


Рис. 8.2. Обчислення визначеного інтеграла

Знаходження наближеного значення інтеграла називається квадратурою, а формули для наближеного обчислення інтеграла - квадратурними формулами або квадратурними сумами.

Різниця  $R$  між точним значенням інтеграла й наближеним значенням називається залишковим членом або похибкою квадратурної формули, тобто

$$R = \int_a^b f(x)dx - \sum_{i=0}^{n-1} S_i .$$

Якщо в кожній із частин розподілу інтервалу  $[a,b]$  підінтегральна функція апроксимується многочленом нульового ступеня, тобто прямою, паралельною осі ОХ, то квадратурна формула називається формулою прямокутників, а метод - *методом прямокутників*.

Якщо в кожній із частин розподілу інтервалу  $[a,b]$  підінтегральна функція апроксимується многочленом першого ступеня, тобто прямою, що з'єднує дві сусідні вузлові точки, то квадратурна формула називається формулою трапецій, а метод - *методом трапецій*.

Якщо в кожній із частин розподілу інтервалу  $[a,b]$  підінтегральна функція апроксимується многочленом другого ступеня, то квадратурна формула називається формулою Симпсона, а метод - *методом Симпсона*.

## 8.2. Метод прямокутників

Словесний алгоритм *методу прямокутників*:

1. Відрізок  $[a,b]$  ділимо на  $n$  рівних частин із кроком

$$h=(b-a)/n.$$

2. Визначаємо значення  $y_i$  підінтегральної функції  $f(x)$  у кожній частині розподілу, тобто

$$y_i = f(x_i), i = \overline{0, n}.$$

3. У кожній частині розподілу підінтегральну функцію  $f(x)$  апроксимуємо інтерполяційним многочленом ступеня  $n = 0$ , тобто прямою, паралельною осі ОХ. У результаті вся підінтегральна функція на інтервалі  $[a,b]$  апроксимується ламаною лінією.

4. Для кожної частини розподілу визначаємо площу  $S_i$  часткового прямокутника.

5. Підсумуємо ці площини. Наближене значення інтеграла  $I$  дорівнює сумі площин часткових прямокутників.



Якщо висота кожного часткового прямокутника дорівнює значенню підінтегральної функції в лівих кінцях кожного кроку, то метод називається методом лівих прямокутників (рис. 8.3). Тоді квадратурна формула має вигляд

$$I = \sum_{i=0}^{n-1} S_i = \sum_{i=0}^{n-1} h \cdot y_i = h \sum_{i=0}^{n-1} y_i$$

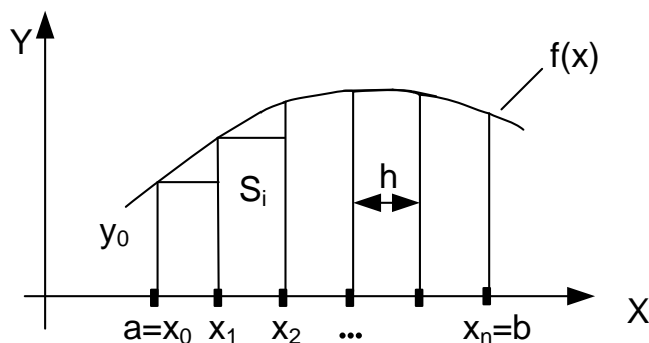


Рис. 8.3. Метод лівих прямокутників

Якщо висота кожного часткового прямокутника дорівнює значенню підінтегральної функції в правих кінцях кожного кроку, то метод називається методом правих прямокутників (рис. 8.4). Тоді квадратурна формула має вигляд

$$I = \sum_{i=0}^n S_i = \sum_{i=0}^n h \cdot y_i = h \cdot \sum_{i=0}^n y_i$$

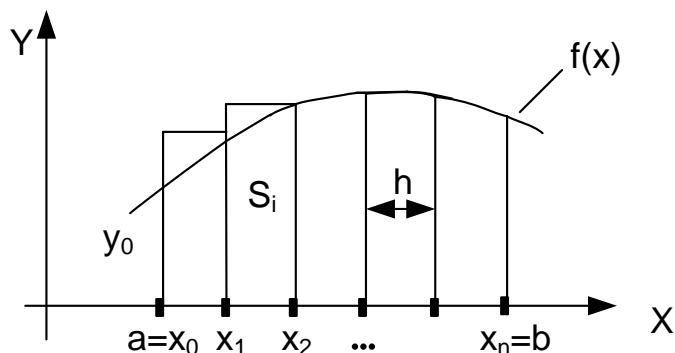


Рис. 8.4. Метод правих прямокутників

Точність кожного методу прямокутників має порядок  $h$ .

Алгоритм обчислення інтеграла побудуємо у вигляді ітераційного процесу пошуку з автоматичним вибором кроку. На кожному кроці будемо зменшувати крок у два рази, тобто збільшувати число кроків  $n$  у два рази. Вихід із процесу пошуку організуємо за точністю обчислення інтеграла. Початкове число кроків  $n=2$ . Схема алгоритму методів прямокутників (з автоматичним вибором кроку) наведена на рис.8.5.

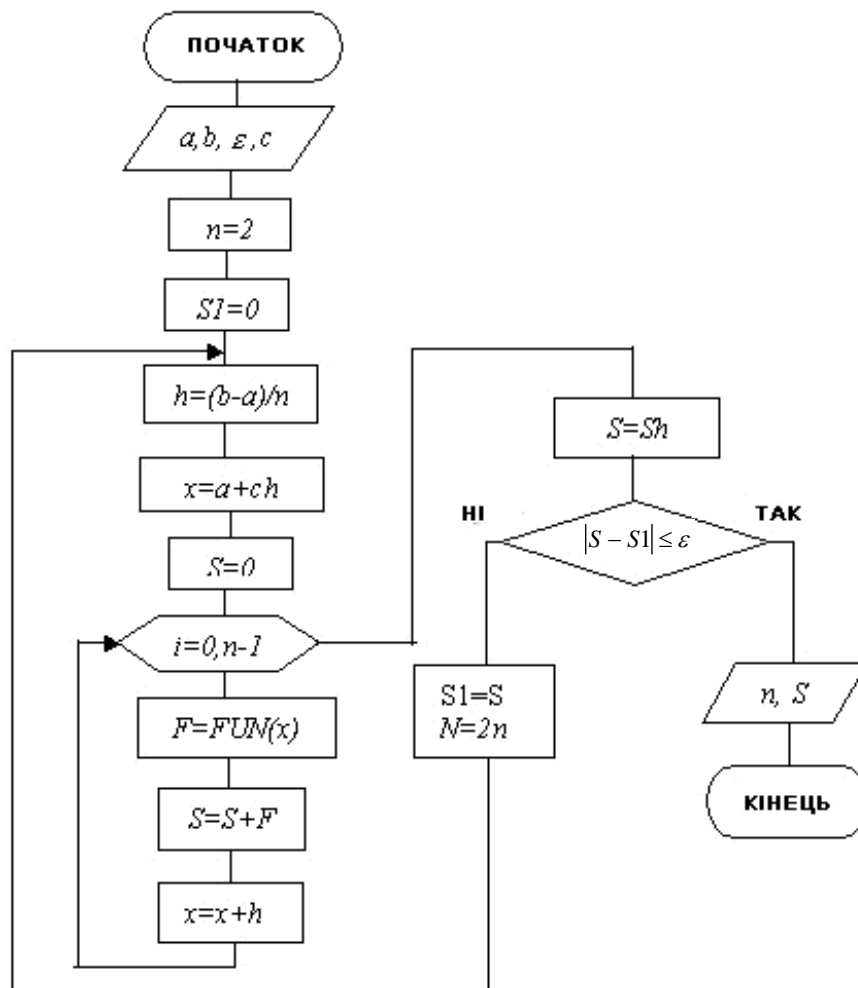


Рис. 8.5. Схема алгоритму методу прямокутників (з автоматичним вибором кроку)

Умовні позначки:  $a, b$  - кінці інтервалу,  $\varepsilon$  - задана точність,  $c=0$  - метод лівих прямокутників,  $c=1$  - метод правих прямокутників,  $S_1$  - значення інтеграла на попередньому кроці,  $S$  - значення інтеграла на поточному кроці.

### 8.3. Метод трапецій

Словесний алгоритм *методу трапецій*:

1. Інтервал  $[a,b]$  ділимо на  $n$  рівних частин із кроком  $h=(b-a)/n$ .

2. Обчислюємо значення підінтегральної функції в кожній вузловій точці

$$y_i = f(x_i), i = \overline{0, n}.$$

3. На кожному кроці підінтегральну функцію  $f(x)$  апроксимуємо прямою, що з'єднує дві сусідні вузлові точки. У результаті вся підінтегральна функція на  $[a,b]$  замінюється ламаною лінією, що проходить через усі вузлові точки.

4. Обчислюємо площину кожної часткової трапеції.

5. Наближене значення інтеграла дорівнює сумі площин часткових трапецій, тобто

$$I = \sum_{i=1}^{n-1} S_i.$$

Знайдемо площини  $S_i$  часткових трапецій:

$$S_0 = \frac{1}{2}h(y_0 + y_1),$$

$$S_1 = \frac{1}{2}h(y_1 + y_2),$$

$$S_2 = \frac{1}{2}h(y_2 + y_3),$$

....

$$S_{n-1} = \frac{1}{2}h(y_{n-1} + y_n).$$

Наближене значення інтеграла дорівнює

$$I = \sum_{i=1}^{n-1} S_i = h/2 \sum_{i=1}^n (y_i + y_{i+1}) = h/2 (y_0 + y_n + 2 \sum_{i=1}^{n-1} y_i).$$

Точність *методу трапецій* має порядок  $h^2$ .

Схема алгоритму *методу трапецій* наведена на рис. 8.6.

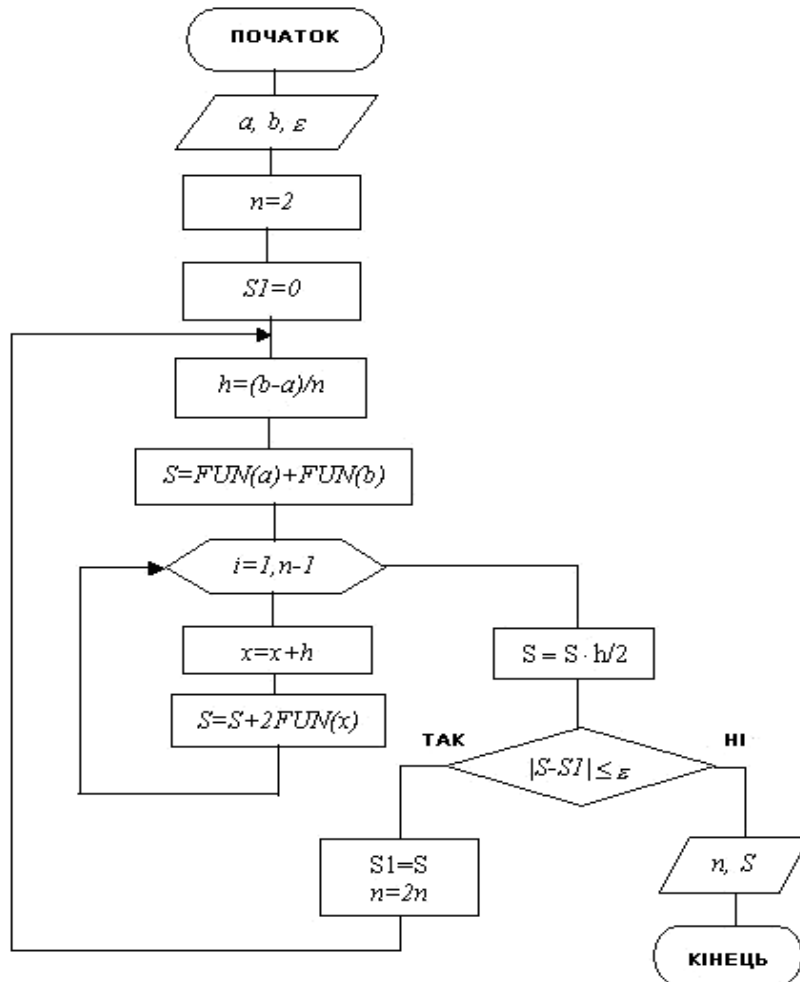


Рис. 8.6. Схема алгоритму методу трапецій (з автоматичним вибором кроку)

## 8.4. Метод Сімпсона

У *методі Сімпсона* в кожній частині розподілу підінтегральна функція апроксимується квадратичною параболою  $a_0x^2+a_1x+a_2$ . У результаті вся крива підінтегральної функції на інтервалі  $[a,b]$  замінюється кусочно-безперервною лінією, що

складається з відрізків квадратичних парабол. Наближене значення інтеграла  $I$  дорівнює сумі площин під квадратичними параболою.

Оскільки для побудови квадратичної параболі необхідно мати три точки, то кожна частина розподілу в *методі Симпсона* включає два кроки, тобто  $L_k=2h$ .

У результаті кількість частин розподілу  $N=2n$ . Тому  $n$  у *методі Симпсона* завжди парне число.

Визначимо площу  $S_1$  на інтервалі  $[x_0, x_2]$  (рис.8.2).

Виходячи з геометричного змісту визначеного інтеграла площа  $S_1$  рівна визначеному інтегралу від квадратичної параболі на інтервалі  $[x_0, x_2]$ :

$$S_1 = \int_{x_0}^{x_2} (a_0 x^2 + a_1 x + a_2) dx = \left. \frac{1}{3} a_0 x^3 + \frac{1}{2} a_1 x^2 + a_2 x \right|_{x_0}^{x_2} = \frac{x_2 - x_0}{6} (2a_0(x_0 + x_2) + 3a_1(x_0 + x_2) + 6a_2).$$

Невідомі коефіцієнти квадратичної параболі  $a_0, a_1, a_2$  визначаємо з умови проходження параболою через три вузлові точки з координатами  $(x_0, y_0), (x_1, y_1), (x_2, y_2)$ .

На підставі цієї умови будемо систему лінійних рівнянь:

$$\begin{cases} a_0 x_0^2 + a_1 x_0 + a_2 = y_0, \\ a_0 x_1^2 + a_1 x_1 + a_2 = y_1, \\ a_0 x_2^2 + a_1 x_2 + a_2 = y_2. \end{cases}$$

Розв'язуючи цю систему, знайдемо коефіцієнти параболі.

У результаті маємо  $S_1 = \frac{h}{3}(y_0 + 4y_1 + y_2)$ .

Для  $[x_2, x_4]$   $S_2 = \frac{h}{3}(y_2 + 4y_3 + y_4)$ .

.....

Для  $[x_{i-1}, x_{i+1}]$   $S_k = \frac{h}{3}(y_{i-1} + 4y_i + y_{i+1})$ ,

де  $k = \frac{i+1}{2}$ .

Підсумовуючи усі площини  $S_i$  під квадратичними параболою, одержимо квадратурну формулу *методу Сімпсона*:

$$S = \sum_{k=1}^{N2} S_k = \frac{h}{3} \sum_{k=1}^{N2} (y_{i-1} + y_i + y_{i+1}),$$

де  $N2$  - кількість частин розподілу.

Точність *методу Сімпсона* має порядок  $h^3 \cdot h^4$ .

Схема алгоритму *методу Сімпсона* наведена на рис. 8.7.

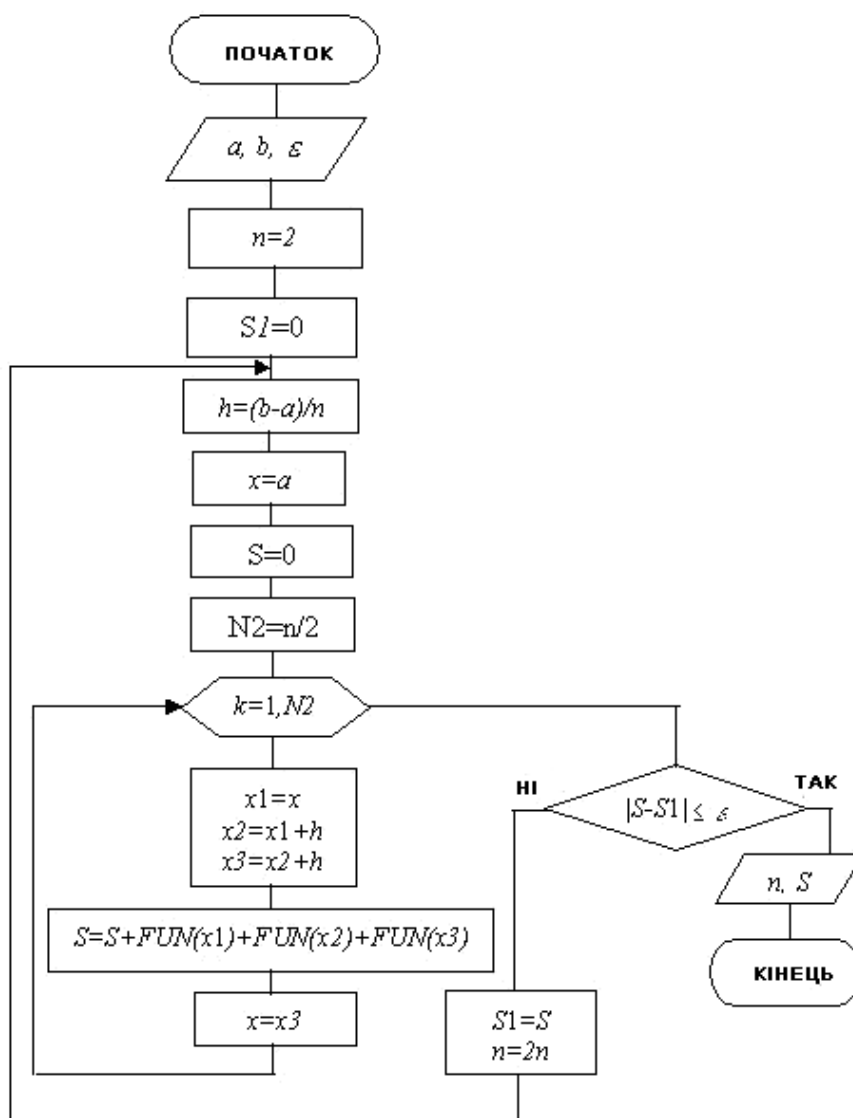


Рис. 8.7. Схема алгоритму Сімпсона (з автоматичним вибором кроку)

## 9. КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ ДИНАМІЧНИХ СИСТЕМ

Динамічні *системи* - це системи, у яких вхідні змінні є функціями від часу або якихось інших параметрів. Описуються ці системи диференціальними й інтегральними рівняннями. Наприклад, більша частина законів механіки, електротехніки, теорії пружності, теорії управління і т. д. описуються за допомогою диференціальних рівнянь.

На практиці *динамічні системи* зустрічаються дуже часто. Моделювання систем, пов'язаних з рухом тіл, розрахунками потоків енергії, розрахунками потоків матеріальних ресурсів, розрахунками обігу коштів і т. д., в остаточному підсумку зводиться до побудови й розв'язання диференціальних рівнянь.

Прямолінійний рух тіла під дією змінної сили  $F(t, S, S')$ , де  $S=S(t)$ , описується диференціальним рівнянням другого порядку у формі рівняння Ньютона:

$$m \cdot S'' = F(t, S, S'),$$

де  $m$  - маса тіла;  
 $S$  - переміщення тіла;  
 $S'$  - лінійна швидкість;  
 $S''$  - лінійне прискорення.

При цьому початкові умови, що задаються, мають чіткий фізичний зміст:

$$\begin{aligned} S \Big|_{t=0} &= S_0, \\ S' \Big|_{t=0} &= S'_0. \end{aligned}$$

Це - початкове положення тіла і його початкова швидкість.

Круговий рух тіла під дією крутного моменту  $M_{kp}(t, \varphi, \varphi')$ , де  $\varphi = \varphi(x)$ , описується аналогічно:

$$I_p \cdot \varphi'' = M(t, \varphi, \varphi'),$$

де  $I_p$  - полярний момент інерції тіла;  
 $\varphi$  - кут повороту;  
 $\varphi'$  - кутова швидкість;  
 $\varphi''$  - кутове прискорення.

При побудові математичних моделей систем, машин, механізмів з урахуванням коливань, що виникають у них, також необхідно побудувати й розв'язати диференціальне рівняння, тому що всі види коливань (вільні гармонійні, змушені) також описуються диференціальними рівняннями.

На практиці лише невелике число диференціальних рівнянь допускає інтегрування у квадратурах, ще рідше вдається одержати розв'язок в елементарних функціях. Саме тому велике поширення при дослідженні математичних моделей за допомогою ЕОМ одержали чисельні методи розв'язання диференціальних рівнянь.

### **Чисельні методи розв'язання диференціальних рівнянь першого порядку**

Загальний вигляд диференціального рівняння

$$F(x, y, y') = 0. \quad (9.1)$$

Нормальна форма диференційного рівняння

$$y' = f(x, y), \quad (9.2)$$

де  $y = y(x)$  - невідома функція, що підлягає визначенню,

$f(x, y)$  - права частина диференційного рівняння в нормальній формі, дорівнює першій похідній функції  $y(x)$ . До функції  $f(x, y)$ , крім аргументу  $x$ , входить і сама невідома функція  $y(x)$ .



Приклад.

$x \cdot y' - (x^2 - 1) \cdot y = 0$  - загальний вигляд диференціального рівняння першого порядку,

$y' = \frac{x^2 - 1}{x} \cdot y$  - нормальна форма цього самого рівняння.

Якщо невідома функція  $y(x)$  залежить від одного аргументу  $x$ , то диференціальне рівняння вигляду

$$y' = f(x, y)$$

називається звичайним диференціальним рівнянням.

Якщо функція залежить від декількох аргументів, то таке диференціальне рівняння називається диференціальним рівнянням у частинних похідних.

Загальним розв'язком звичайного диференціального рівняння

$$y' = f(x, y),$$

є сімейство функцій  $y = y(x, c)$  (рис. 9.1).

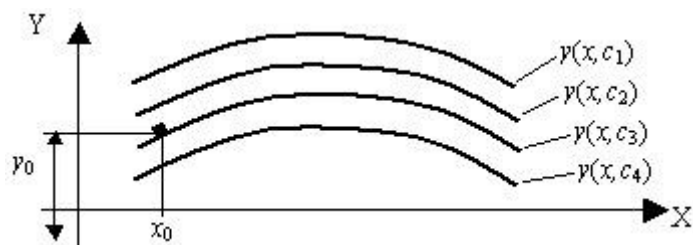


Рис. 9.1

При розв'язанні прикладних задач шукають частинний розв'язок диференціальних рівнянь. Виділення частинного розв'язку з сімейства загальних розв'язків здійснюється за допомогою завдання початкових умов:

$$y \Big|_{x=x_0} = y_0, \quad (9.3)$$

тобто початкової точки з координатами  $(x_0, y_0)$ .

Знаходження частинного розв'язку диференціального рівняння (9.2), що задовольняє початкову умову (9.3), називається задачею Коші.

У чисельних методах задача Коші ставиться в такий спосіб: знайти табличну функцію  $y' = f(x, y), i = \overline{1, n}$ , яка задовольняє заданому диференціальному рівнянню (9.2) і початкову умову (9.3) на відрізку  $[a, b]$  із кроком  $h$ , тобто побудувати таблицю

<b>i</b>	<b>x</b>	<b>Y</b>
0	$x_0$	$y_0$
1	$x_1$	$y_1$
2	$x_2$	$y_2$
3	$x_3$	$y_3$
...	...	...
<b>n</b>	$x_n$	$y_n$

Тут  $h$  – крок інтегрування диференціального рівняння,  $a=x_0$  – початок відрізка інтегрування рівняння,  $b=x_n$  – кінець відрізка,  $n=(b-a)/h$  – число кроків інтегрування рівняння.

На графіку (рис. 9.2) рішення задачі Коші чисельними методами зображено у вигляді сукупності вузлових точок з координатами  $(x_i, y_i), i = \overline{1, n}$ .

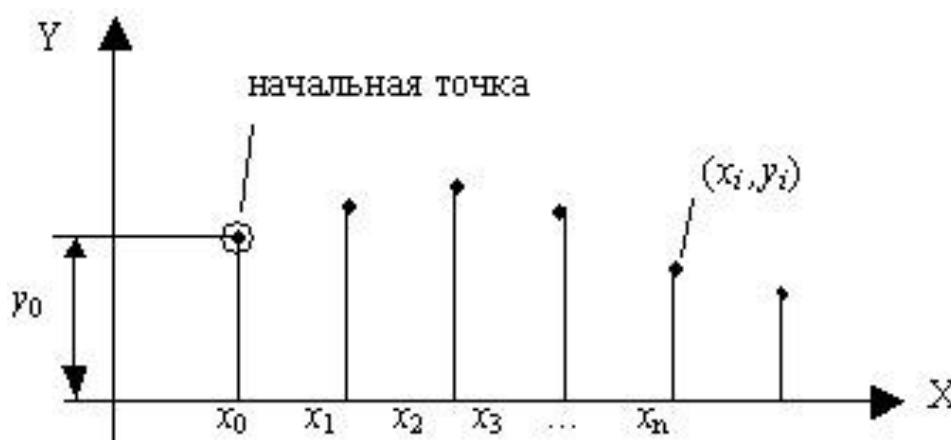


Рис. 9.2

## 9.1. Методи Рунге - Кутта

Найбільш ефективними та поширеними методами розв'язання задачі Коші є методи Рунге – Кутта. Вони засновані на апроксимації шуканої функції  $y(x)$  у межах кожного кроку многочленом, який отриманий за допомогою розкладання функції  $y(x)$  в околі кроку  $h$  кожної  $i$ -ї точки в ряд Тейлора:

$$\begin{aligned} y(x_i + h) = & y(x_i) + h \cdot y'(x_i) + \frac{h^2}{2!} y''(x_i) + \\ & + \frac{h^3}{3!} y'''(x_i) + \frac{h^4}{4!} y^{(4)}(x_i) + \frac{h^5}{5!} y^{(5)}(x_i) + \dots \end{aligned} \quad (9.4)$$

Зрізаючи ряд Тейлора в різних точках і відкидаючи праві члени ряду, Рунге і Кутт одержували різні методи для визначення значень функції  $y(x)$  у кожній вузловій точці. Точність кожного методу визначається відкинутими членами ряду.

## 9.2. Метод Рунге - Кутта 1-го порядку (метод Ейлера)

Відкинемо у виразі (9.4) члени ряду, що містять  $h^2, h^3, h^4 \dots$ .

Тоді  $y(x_i + h) = y(x_i) + h \cdot y'(x_i)$ .

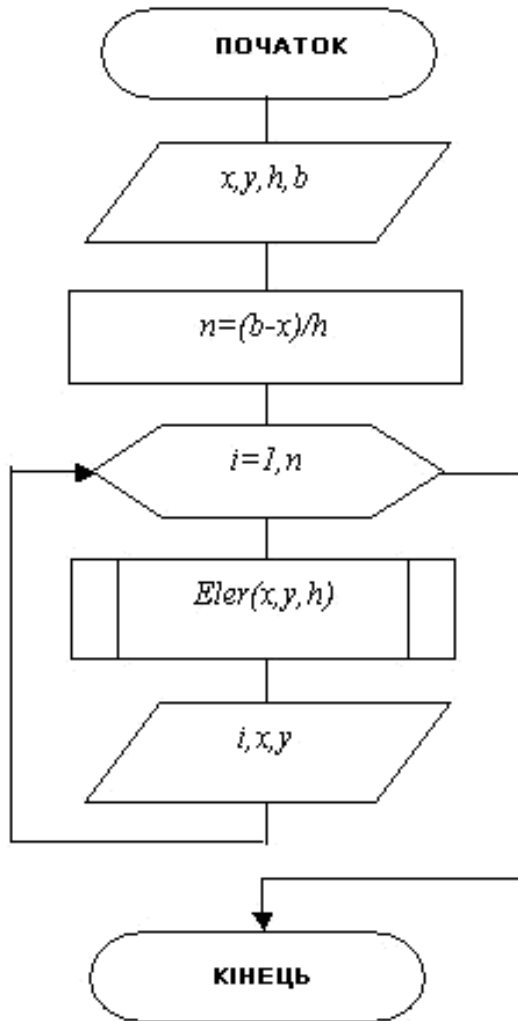
Оскільки  $y'(x_i) = f(x_i, y_i)$ , одержимо формулу Ейлера:

$$y_{i+1} = y_i + h \cdot f(x_i, y_i). \quad (9.5)$$

Оскільки точність методів Рунге-Кутта визначається відкинутими членами ряду (9.4), то точність методу Ейлера на кожному кроці становить  $\approx h^2$ .

Алгоритм *методу Ейлера* можна побудувати у вигляді двох програмних модулів: основної програми й підпрограми ELER, що реалізує метод.

Основна програма



Підпрограма ELER

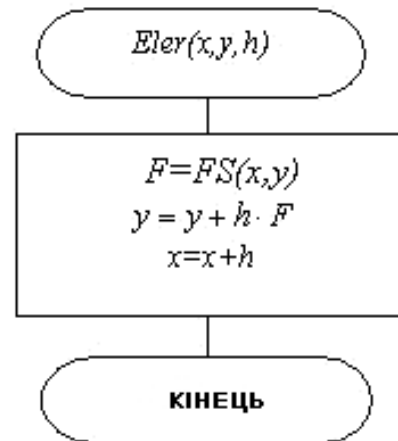


Рис. 9.3. Схема алгоритму методу Ейлера

Тут  $(x, y)$  - при введенні початкова точка, далі - поточні значення табличної функції,  $h$ -крок інтегрування диференціального рівняння,  $b$ -кінець інтервалу інтегрування.

Розглянемо геометричний зміст *методу Ейлера*.

Формула Ейлера має вигляд

$$y_{i+1} = y_i + h \cdot f(x_i, y_i),$$

де  $f(x_i, y_i) = y'(x_i) = \operatorname{tg} \alpha_i$ .

Тоді формула Ейлера має вигляд

$$y_{i+1} = y_i + h \cdot \operatorname{tg} \alpha_i,$$

де  $\operatorname{tg} \alpha_i$  - тангенс кута нахилу дотичної до шуканої функції  $y(x)$  у початковій точці кожного кроку.

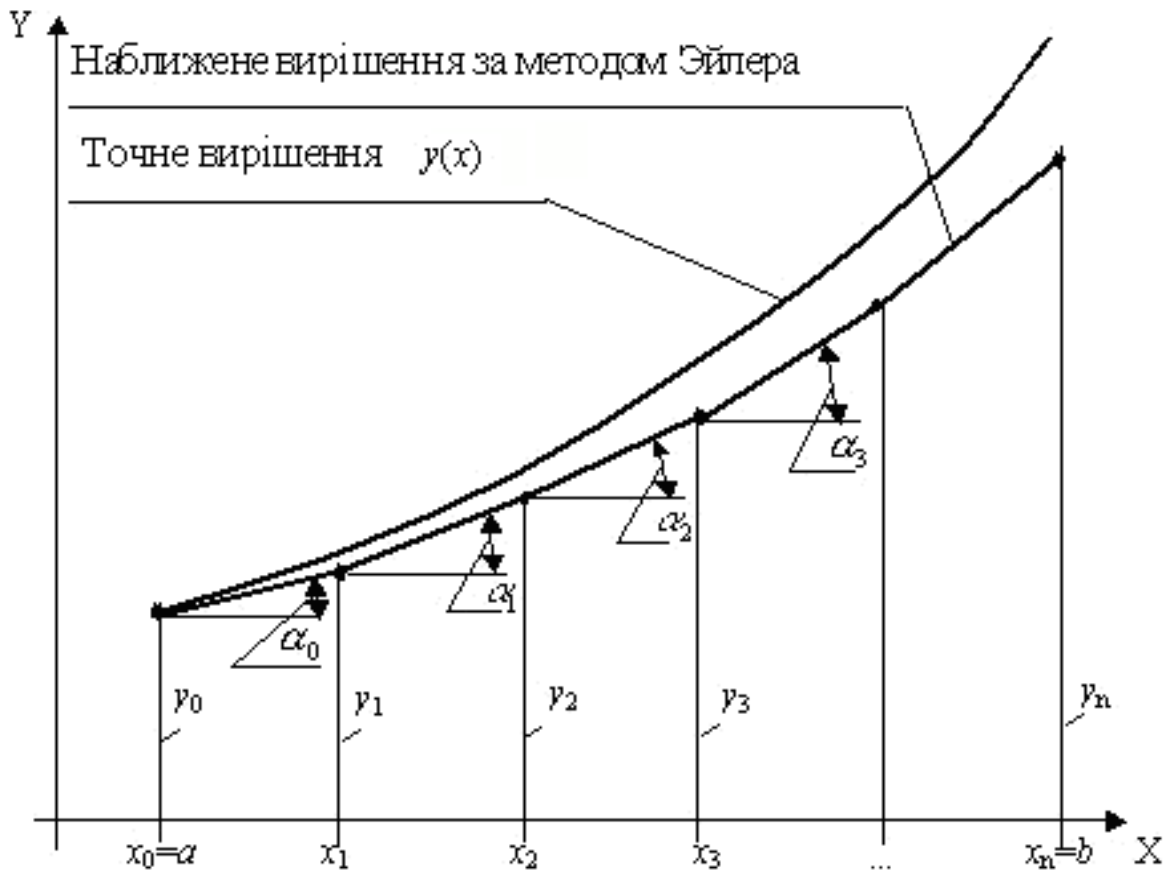


Рис. 9.4. Геометричний зміст методу Ейлера

У результаті в *методі Ейлера* на графіку (рис. 9.4) уся шукана функція  $y(x)$  на інтервалі  $[a,b]$  апроксимується ламаною лінією, кожний відрізок якої на кроці  $h$  лінійно апроксимує шукану функцію. Тому метод Ейлера одержав ще назву методу ламаних.

У *методі Ейлера* нахил дотичної в межах кожного кроку вважається постійним і дорівнює значенню похідної в початковій точці кроку  $x_i$ . У дійсності похідна (тангенс кута нахилу дотичної до кривої  $y(x)$ ) у межах кожного кроку змінюється. Тому в точці  $x_{i+h}$  нахил дотичної не повинен дорівнювати нахилу в точці  $x_i$ . Отже, на кожному кроці вноситься похибка.

Перший відрізок ламаної дійсно торкається шуканої інтегральної кривої  $y(x)$  у точці  $(x_0, y_0)$ . На наступних же кроках дотичні проводяться з точок  $(x_i, y_i)$ , підрахованих з похибкою. У результаті з кожним кроком помилки накопичуються.

Основний недолік *методу Ейлера* - систематичне накопичування помилок. Тому *метод Ейлера* рекомендується застосовувати для розв'язання диференціальних рівнянь при малих значеннях кроку інтегрування  $h$ .

## 10. ДЕЯКІ ВІДОМОСТІ З ЕЛЕМЕНТАРНОЇ ТЕОРІЇ ПОХИБОК

### 10.1. Джерела й класифікація похибок результату чисельного розв'язання задач

Одержуваний на ЕОМ розв'язок  $y^*$  майже завжди (за винятком деяких досить спеціальних випадків) містить похибку, тобто є наближеним. Неможливість одержання точного розв'язку впливає вже з обмеженої розрядності обчислювальної машини.

Наявність похибки розв'язання обумовлена досить глибокими причинами. Перелічимо їх.

1. Математична модель є лише наближеним описом реального процесу. Характеристики процесу, обчислені в рамках прийнятої моделі, свідомо відрізняються від дійсних характеристик, причому їх похибка залежить від ступеня адекватності моделі реальному процесу.

2. Вихідні дані, як правило, містять похибки, оскільки вони або отримуються в результаті експериментів (вимірювання), або є результатом розв'язку деяких допоміжних задач.

3. Застосовувані для розв'язання задачі методи в більшості випадків є наближеними. Знайти розв'язок задачі, що виникає на практиці, у вигляді кінцевої формули можливо тільки в окремих, дуже спрощених ситуаціях.

4. При введенні вихідних даних в ЕОМ, виконанні арифметичних операцій і виведенні результатів на друк проводяться округлення.

Нехай  $y$  - точне значення величини, обчислення якої є метою поставленої задачі. Похибка, що відповідає першим двом із зазначених причин,  $\delta_n y$  називається *неусувною похибкою*.

Така назва пояснюється тим, що прийняття математичної моделі й завдання вихідних даних вносять у розв'язок помилку, яка не може бути усунута далі. Єдиний спосіб зменшити цю похибку - перейти до більш точної математичної моделі й задати більш точні вихідні дані.

Похибка  $\delta_m u$ , джерелом якої є метод розв'язання задачі, називається *похибкою методу*, а похибка  $\delta_e u$ , що виникає через округлення при введенні, виведенні та обчисленнях, - *обчислювальною похибкою*. Таким чином, повна похибка результату розв'язання задачі на ЕОМ  $\delta u = u - u^*$  складається з трьох складових: похибки, що неможна усунути, похибки методу і обчислювальної похибки, тобто  $\delta u = \delta_n u + \delta_m u + \delta_e u$ .

Будемо далі виходити з припущення, що математична модель фіксована й вхідні дані задаються ззовні, так що вплинути на значення величини  $\delta_n u$  в процесі розв'язання задачі дійсно неможна. Однак це зовсім не означає, що попередні оцінки величини похибки, що неможна усунути, не потрібні. Достовірна інформація про порядок величини  $\delta_n u$  дозволяє усвідомлено вибрати метод розв'язання задачі та розумно задати його точність. Бажано, щоб величина похибки методу була в 2-10 раз менше похибки, що неможна усунути. Більше значення  $\delta_m u$  відчутно знижує точність результату, менше - звичайно вимагає збільшення витрат, практично вже не впливаючи на значення повної похибки. Іноді характер використання результату такий, що цілком припустимо, щоб величина  $\delta_m u$  була порівнянна з  $\delta_n u$  або навіть трохи перевищувала її.

Уміння аналізувати похибки при розв'язанні прикладної задачі і дотримувати між ними розумний компроміс дозволяє суттєво заощаджувати використовувані ресурси і є ознакою високої кваліфікації.

## **10.2. Наближені числа. Абсолютна й відносна похибки**

### ***10.2.1. Абсолютна й відносна похибки***

Нехай  $a$  - точне (загалом кажучи, невідоме) значення деякої величини,  $a^*$  - відоме наближене значення тієї самої величини (наближене число). Помилкою (або похибкою) наближеного числа  $a^*$  називають різницю  $a - a^*$  між точним і наближеним значеннями.



Найпростішим кількісним засобом оцінювання помилки є абсолютна похибка

$$\Delta(a^*) = |a - a^*| \quad (10.1)$$

Однак за величиною абсолютної похибки далеко не завжди можна зробити правильний висновок щодо якості наближення.

Наприклад,  $\Delta(a^*) = 0.01$ . Як визначити, це велика похибка або мала? Усе залежить від прийнятих одиниць вимірювання й масштабу величини  $a$ . Якщо  $a \approx 0,05$ , то точність наближення невелика, якщо  $a = 5 \cdot 10^6$ , то точність обчислень може бути визнана дуже високою.

Таким чином, природньо співвіднести похибку величини та її значення, для чого вводиться поняття відносної похибки (при  $a \neq 0$ )

$$\delta(a^*) = \frac{|a - a^*|}{|a|} = \frac{\Delta(a^*)}{|a|}. \quad (10.2)$$

Оскільки значення  $a$  невідоме, то безпосереднє обчислення величин  $\Delta(a^*)$  і  $\delta(a^*)$  за формулами (10.1), (10.2) неможливо. Більш реальна задача, що часто піддається розв'язанню, полягає в одержанні оцінок похибки у вигляді

$$|a - a^*| \leq \bar{\Delta}(a^*), \quad (10.3)$$

$$\frac{|a - a^*|}{|a|} \leq \bar{\delta}(a^*), \quad (10.4)$$

де  $\bar{\Delta}(a^*)$  й  $\bar{\delta}(a^*)$  - відомі величини, які ми будемо називати верхніми границями (або просто *границями*) абсолютної та відносної похибок.

**Зауваження.** У літературі з методів обчислень широко використовується термін "точність". Прийнято говорити про точність вхідних даних і розв'язку, про підвищення й зниження точності обчислень і т. д. Ми також будемо використовувати цю термінологію, за якою ховається досить простий зміст. Точність у якісних міркуваннях звичайно виступає як протилежність похибки, хоча для кількісного її вимірювання використовуються ті самі характеристики (наприклад, абсолютна й відносна похибки).

Точне значення величини - це значення, що не містить похибки. Підвищення точності сприймається як зменшення похибки, а зниження точності - як збільшення похибки. Фраза "потрібно знайти розв'язок із заданою точністю  $\varepsilon$ " означає, що ставиться задача про знаходження наближеного розв'язку, для якого значення похибки якого не перевищує заданого значення  $\varepsilon$ .

Таким чином, варто було б говорити про абсолютну точність і відносну точність, але частіше цього не роблять, вважаючи, що з контексту зрозуміло, як вимірюється величина похибки.

### ***10.2.2. Правила запису наближених чисел***

Нехай наближене число  $a^*$  задано у вигляді кінцевого десяткового дроби:

$$a^* = a_n a_{n-1} \dots a_0 \cdot \beta_1 \beta_2 \dots \beta_m.$$

Значущими цифрами числа  $a^*$  називають усі цифри в його запису, починаючи з першої ненульової ліворуч.

Приклад.

Число  $a_1^* = 0.0284$  і  $a_2^* = 0,028400$ .

Значущі цифри підкреслені. Перше число має 3, а друге 5 значущих цифр.

Значущу цифру  $a^*$  називають правильною, якщо абсолютна похибка числа не перевищує одиниці розряду, відповідного цій цифрі.

Приклад.

Якщо  $\Delta(a^*) = 2 \cdot 10^{-4}$ , то число  $a^* = 0.028400$  має 2 правильні значущі цифри, які підкреслені.

Зв'язок точності числа з кількістю його правильних значущих цифр і зв'язок між втратою точності й втратою правильних цифр можна виразити такими твердженнями:

1. Якщо число  $a^*$  містить  $N$  правильних значущих цифр, то справедлива нерівність  $\delta(a^*) \leq (10^{N-1} - 1)^{-1} \approx 10^{-N+1}$ .

2. Для того щоб число  $a^*$  містило  $N$  правильних значущих цифр, досить, щоб була виконана нерівність  $\delta(a^*) \leq (10^{N-1} + 1)^{-1} \approx 10^{-N}$ .

3. Якщо число  $a^*$  має рівно  $N$  правильних значущих цифр, то  $10^{N-1} \leq \delta(a^*) \leq 10^{-N+1}$ , і в такий спосіб  $\delta(a^*) \sim 10^{-N}$ .

Зазначимо, що границі абсолютної й відносної похибки прийнято записувати з однією або двома значущими цифрами. Більша точність у записі цих величин, як правило, не має смислу, тому що звичайно вони є досить грубими оцінками дійсних значень похибок, і, крім того, для практичного використання часто буває досить знати тільки їх порядок.

Той факт, що число  $a^*$  є наближеним значенням числа з верхньою границею абсолютної похибки  $\bar{\Delta}(a^*)$ , прийнято записувати у вигляді  $a = a^* \pm \bar{\delta}(a^*)$ .

Приклад.

Оцінимо точність часто використовуваного в найпростіших розрахунках наближення  $\pi = 3.14$  до числа  $\pi$ . Відомо, що  $\pi = 3.14159$ , тому  $\pi - \pi^* = 0.00159 \dots$ . Отже, можна прийняти  $\bar{\Delta}(\pi^*) = 0.0016$  й  $\bar{\delta}(\pi^*) = 0.0016$ . Отже,  $\pi = 3.14(1 \pm 0.051\%)$ .

**Зауваження.** Якщо число  $a^*$  наводиться в якості результату без зазначення величини похибки, то прийнято вважати, що всі його значущі цифри є правильними. Користувач-початківець часто занадто довіряє виведеним з ЕОМ цифрам, припускаючи, що обчислювальна машина дотримується тієї самої угоди. Однак це не зовсім так: число може бути виведене з такою кількістю

значущих цифр, скільки зажадає програміст завданням відповідного формату. Як правило, серед цих цифр тільки невелике число перших виявляться правильними, а, можливо, правильних цифр немає зовсім. Аналізувати результати обчислень і визначати ступінь їх вірогідності зовсім непросто. Одна із цілей вивчення обчислювальних методів і полягає в досягненні розуміння того, що можна й чого неможна очікувати від результатів, що отримані на ЕОМ.

### **10.2.3. Округлення**

Часто виникає необхідність в округленні числа  $a$ , тобто в заміні його іншим числом  $a^*$  з меншим числом значущих цифр. Найбільш простий з них - зрізання - полягає у відкиданні всіх цифр, розташованих праворуч від  $n$ -ї значущої цифри. Більш кращим є округлення по доповненню. У найпростішому варіанті це правило округлення полягає в такому. Якщо перша ліворуч із цифр, що відкидаються, менше 5, то цифри, що зберігаються, залишаються без зміни. Якщо ж вона більше або дорівнює 5, то в молодший розряд, що зберігається, додається одиниця.

Абсолютна величина похибки округлення при округленні по доповненню не перевищує половини одиниці розряду, що відповідає останній цифрі, яка залишається, а при округленні зрізанням - одиниці того самого розряду.

Приклад.

При округленні числа  $a=2.14821$  до трьох значущих цифр усіканням одержуємо число  $a^*=2.14$ , а по доповненню -  $a^*=2.15$ .

Границі абсолютної й відносної похибок прийняте завжди округляти в бік збільшення.

#### 10.2.4. Похибки арифметичних операцій над наближеними числами

Нехай  $a^*$  і  $b^*$  наближені значення чисел  $a$  і  $b$ .  
Справедливі наступні твердження:

1. Абсолютна похибка алгебраїчної суми (суми або різниці) не перевершує суми абсолютних похибок, що складаються:

$$\Delta(a^* \pm b^*) \leq \Delta(a^*) + \Delta(b^*). \quad (10.5)$$

Маємо

$$\Delta(a^* \pm b^*) = |(a \pm b) - (a^* \pm b^*)| = |(a - a^*) \pm (b - b^*)| \leq \Delta(a^*) + \Delta(b^*).$$

Наслідок. У силу нерівності (2.8) природно покласти

$$\bar{\Delta}(a^* \pm b^*) = \bar{\Delta}(a^*) + \bar{\Delta}(b^*). \quad (10.6)$$

2. Нехай  $a$  і  $b$  ненульові числа одного знака. Тоді слушні нерівності

$$\delta(a^* + b^*) \leq \delta_{\max}, \quad \delta(a^* - b^*) \leq \nu \delta_{\max}, \quad (10.7)$$

де  $\delta_{\max} = \max\{\bar{\delta}(a^*), \delta(b^*)\}$ ,  $\nu = |a + b|/|a - b|$ .

Приклад.

Нехай розв'язується інженерна задача, у якій остаточний варіант у обчислюється за формулою  $y = 1 - x$  за допомогою попередньо обумовленого уточнення  $x$ . Припустимо, що знайдене наближене значення  $x^* = 0.0999996$  містить 6 правильних значущих цифр, тоді  $y^* = 1 - 0.0999996 = 0.0000004$ . У процесі обчислень загублено 5 верхніх цифр. Це катастрофічна втрата точності.

Підкреслимо, що тут винуватцем "катастрофи" є не операція віднімання, а запропонований метод розв'язання задачі, де остаточний результат одержують за допомогою віднімання двох близьких чисел. Виконання цієї операції лише робить очевидним те, що дійсно корисною є інформація про значення у вже виявилася загубленою до віднімання. Якщо немає іншого варіанта розрахунків, то для одержання прийняттого результату варто було б попередньо обчислити  $x$  із суттєво більшим числом правильних знаків, враховуючи, що 5 старших значущих цифр при вирахованні будуть загублені.

Отже, одержуємо такий важливий висновок. При побудові чисельного методу розв'язання задач слід уникати віднімання близьких чисел одного знака. Якщо таке віднімання неминуче, то слід обчислювати аргументи з підвищеною точністю, враховуючи її втрату приблизно в  $\nu = |a+b|/|a-b|$  раз.

3. Для відносних похибок добутку й частки наближених чисел правильними є оцінки

$$\delta(a^*b^*) \leq \delta(a^*) + \delta(b^*) + \delta(a^*)\delta(b^*), \quad (10.8)$$

$$\delta(a^*/b^*) \leq \frac{\delta(a^*) + \delta(b^*)}{1 - \delta(b^*)}, \quad (10.9)$$

в останній з яких вважається, що  $\delta(b^*) < 1$ .

Наслідок. Якщо  $\bar{\delta}(a^*) < 1$  й  $\bar{\delta}(b^*) \leq 1$ , то для оцінки границь відносних похибок можна використовувати такі наближені нерівності:

$$\bar{\delta}(a^*b^*) \approx \bar{\delta}(a^*) + \bar{\delta}(b^*), \quad \bar{\delta}(a^*/b^*) \approx \bar{\delta}(a^*) + \bar{\delta}(b^*). \quad (10.10)$$

Саме нерівності (10.10) найчастіше використовують для практичної оцінки похибки.

Таким чином можливо зробити такий висновок: виконання арифметичних операцій над наближеними числами, як правило,

супроводжується втратою точності. Єдина операція, при якій втрата точності не відбувається, - це додавання числа одного знака. Найбільша втрата точності може відбутися при відніманні чисел одного знака.

### 10.2.5. Похибка функції

Нехай  $f(x) = f(x_1, x_2, \dots, x_m)$  - функція  $m$  змінних, що диференціюється в області  $G$ , обчислення якої проводиться при приблизно заданих значеннях аргументів.  $x_1^*, x_2^*, \dots, x_m^*$  Така ситуація виникає, наприклад, щораз, коли на ЕОМ проводяться розрахунки за формулою. Важливо знати, якою є величина неусувної помилки, викликаній тим, що замість значення  $y = f(x)$  в дійсності обчислюється значення  $y^* = f(x^*)$ , де  $x^* = (x_1^*, x_2^*, \dots, x_m^*)$ .

Введемо позначення: нехай  $[x, x^*]$  - відрізок, що з'єднує точки  $x$  й  $x^*$ , і  $f'_{x_j} = \partial f / \partial x_j$ .

Твердження 1. Для абсолютної похибки значення  $y^* = f(x^*)$  справедливою є така оцінка:

$$\Delta(y^*) \leq \sum_{j=1}^m \max_{[x, x^*]} |f'_{x_j}| \Delta(x_j^*). \quad (10.11)$$

Оцінка (10.11) випливає з формули кінцевих приростів Лагранжа:

$$f(x) - f(x^*) = \sum_{j=1}^m f'_{x_j}(\tilde{x})(x_j - x_j^*), \quad \tilde{x} \in [x, x^*].$$

Наслідок. Якщо  $x^* \approx x$ , то в силу виразу (10.11) можна покласти, що

$$\bar{\Delta}(y^*) \approx \sum_{j=1}^m |f'_{x_j}(x^*)| \bar{\Delta}(x_j^*), \quad (10.12)$$

$$\bar{\Delta}(y^*) \approx \sum_{j=1}^m |f'_{x_j}(x)| \bar{\Delta}(x_j^*). \quad (10.13)$$

Рівність (10.12) зручна для практичних оцінок.

З формул (10.12), (10.13) випливають наближені рівності для оцінки відносних похибок:

$$\bar{\delta}(y^*) \approx \sum_{j=1}^m v_j^* \bar{\delta}(x_j^*), \quad \delta(y^*) \approx \sum_{j=1}^m v_j^* \delta(x_j^*), \quad (10.14)$$

Тут

$$v_j^* = \frac{|x_j^*| |f'_{x_j}(x^*)|}{|f(x^*)|}, \quad v_j = \frac{|x_j| |f'_{x_j}(x^*)|}{|f(x^*)|}. \quad (10.15)$$

Приклад.

Розглянемо квадратне рівняння  $x^2 + bx + c = 0$ , де  $b \approx 10^3$ ,  $c \approx 1$ .

Оцінимо вплив похибок задавання коефіцієнтів на точність обчислення коренів

$$x_1 = f(b, c) = (-b - \sqrt{d})/2;$$

$$x_2 = g(b, c) = (-b + \sqrt{d})/2,$$

де  $d = b^2 - 4c$ .

Зазначимо, що  $x_1 \cdot x_2 = c$ .

При заданих значеннях

$$\sqrt{d} \approx \sqrt{10^6 - 4} \approx 10^3,$$

$$x_1 \approx (-10^3 - 10^3)/2 \approx -10^3,$$

$$x_2 \approx c/x_1 \approx 1/(-10^3) \approx -10^{-3}$$



знайдемо  $f'_b, f'_c, g'_b, g'_c$

$$f'_b = (-1 - b/\sqrt{d})/2 \approx -1 - 10^3(10^3)/2 = \underline{-1},$$

$$f'_c = 1/\sqrt{d} \approx 10^{-3},$$

відповідно

$$g'_b = -x_2/\sqrt{2} \approx 10^{-6},$$

$$g'_c = -1/\sqrt{d} \approx -10^{-3}.$$

Користуючись формулами для оцінки відносної похибки маємо:

- для  $x_1$

$$y^*_{1.1} = |b^*| |f'_b| / |x_1^*| \approx 10^{-6},$$

$$y^*_{1.2} = |c^*| |f'_c| / |x_1^*| \approx 10^{-6};$$

- аналогічно для  $x_2$

$$y^*_{2.1} = |b^*| |g'_b| / |x_2^*| \approx 1,$$

$$y^*_{2.2} = |c^*| |g'_c| / |x_2^*| \approx 1,$$

$$\bar{\delta}(x_1^*) \approx \bar{\delta}(b^*) + 10^{-6} \bar{\delta}(c^*),$$

$$\bar{\delta}(x_2^*) \approx \bar{\delta}(b^*) + \bar{\delta}(c^*).$$

Отже, точність першого кореня практично визначається тільки точністю задачі коефіцієнта  $b$ , у той час як коефіцієнт  $c$  може бути заданий дуже грубо. Для другого кореня вплив похибок у задаванні коефіцієнтів  $b$  і  $c$  практично однаковий.

### 10.2.6. Особливості машинної арифметики

Для проведення розрахунків на ЕОМ з необхідною точністю користувач обов'язково повинен знати особливості машинної арифметики, оскільки в основі причин появи обчислювальної похибки лежить сам спосіб зображення чисел на ЕОМ.

#### 1. Системи числення

Прийнятий спосіб запису чисел полягає у поданні їх упорядкованим набором цифр:

$$x = \pm \alpha_n \dots \alpha_1 \alpha_0 \cdot \beta_1 \beta_2 \dots \beta_m.$$

Кожній позиції (розряду), яку займає цифра до десяткової точки, відповідає певний степінь числа 10. Звідси

$$\bar{x} = \pm \alpha_n \dots \alpha_1 \alpha_0 \cdot \beta_1 \beta_2 \dots \beta_m = \pm (\alpha_n \cdot 10^n + \dots + \alpha^1 \cdot 10^1 + \alpha_0 \cdot 10^0 + \beta_1 \cdot 10^{-1} + \beta_2 \cdot 10^{-2} + \dots + \beta_m \cdot 10^{-m})$$

Приклад.

$$x = 26.5 = 2 \cdot 10^1 + 6 \cdot 10^0 + 5 \cdot 10^{-1}.$$

Для зображення чисел в обчислювальних машинах також використовують позиційні системи числення, однак основами систем служать, як правило, степені числа 2. Це викликано способом зберігання чисел у пам'яті ЕОМ, кожне з яких можна розглядати як набір однотипних елементів, здатних перебувати тільки в одному з двох можливих стійких станів - "включений" або "виключений". Ці стани інтерпретуються відповідно як 0 або 1 - значення двійкового числа. Найпоширеніші системи числення з основою 2 (базисна двійкова система числення) - 8 і 16.

Ігноруючи деякі мало суттєві деталі, будемо вважати, що всі обчислювальні машини працюють у двійковій системі числення. У ній дійсне число  $x$  записується у вигляді виразу (2.24), однак

$\alpha_i$  і  $\beta_j$  - уже двійкові цифри (0 і 1). У цьому випадку повний запис виглядає так:

$$x = \pm(\alpha_n \cdot 2^n + \dots + \alpha_1 \cdot 2^1 + \alpha_0 \cdot 2^0 + \beta_1 \cdot 2^{-1} + \beta_2 \cdot 2^{-2} + \dots + \beta_m \cdot 2^{-m} + \dots).$$

Для зберігання числа в пам'яті ЕОМ виділяється поле стандартної довжини (машинне слово), у якому число записується у вигляді послідовності двійкових цифр. За формою зображення, способом зберігання й реалізації арифметичних операцій суттєво вирізняються два типи використовуваних на обчислювальних машинах чисел: цілі числа й дійсні числа.

## 2. Зображення цілих чисел

Ціле число  $n$  подають у вигляді

$$n = \pm(\alpha_L 2^L + \dots + \alpha_1 2^1 + \alpha_0 2^0), \quad (10.16)$$

де  $L$  - деяке стандартне для ЕОМ ціле число;

$\alpha_i$  - двійкові цифри.

Усього для зберігання числа  $n$  відводять  $L+2$  розрядів (один з них для зберігання знака).

З виразу (10.16) видно, що максимальне за модулем ціле число, надане в ЕОМ, є  $n_{\max} = 2L + \dots + 2^1 + 2^0 = 2^{L+1} - 1$ . Звичайно це значення не дуже велике.

Операції додавання, віднімання й множення над цілими числами реалізовані так, що якщо результат не перевищує за модулем ціле число  $n_{\max}$ , то він є точним. Однак слід зазначити, що якщо модуль результату перевищує  $n_{\max}$ , то на більшості обчислювальних машин ця ситуація не доводиться до відома користувача, відбувається присвоєння результату деякого значення (меншого  $n_{\max}$  за модулем) і обчислення тривають далі.

### 3. Зображення дійсних чисел

У більшості сучасних ЕОМ для дійсних чисел прийнята форма з рухомою точкою, коли кожне число подано у вигляді

$$x = \pm(\gamma_1 \cdot 2^{-1} + \gamma_2 \cdot 2^{-2} + \dots + \gamma_t \cdot 2^{-t})2^p. \quad (10.17)$$

Тут  $\gamma_1, \gamma_2, \dots, \gamma_t$  - двійкові цифри. Число  $x$  нормалізується так, щоб  $\gamma_1 = 1$ , і тому в пам'яті ЕОМ зберігаються тільки значущі цифри. Число  $\mu = \pm(\gamma_1 \cdot 2^{-1} + \gamma_2 \cdot 2^{-2} + \dots + \gamma_t \cdot 2^{-t})$  називається мантиєю числа  $x$ . Кількість  $t$  цифр, які виділяються для запису мантиї, називається розрядністю мантиї, залежить від конструктивних особливостей конкретної обчислювальної машини, але завжди є кінцевим числом. У формулі (10.17)  $p$  - ціле число, яке називається двійковим порядком. Порядок також записують як двійкове ціле число  $p = \pm(\delta_1 \delta_{L-1} \dots \delta_0)_2$ , для зберігання якого в машинному слові виділяється  $L+2$  двійкових розрядів. На рис. 10.1 схематично зображена структура машинного слова для зберігання дійсного числа.

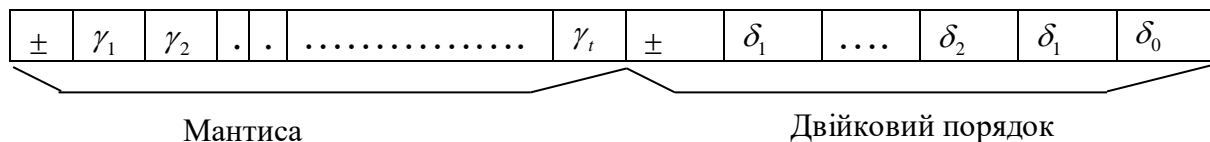


Рис. 10.1

Оскільки нуль - ненормалізоване число (його не можна представити у вигляді виразу (10.17) при  $\gamma_1 \neq 0$ ), для його зберігання передбачають особливий спосіб запису.

Приклад.

Представимо число 26,5 у двійковій системі із рухомою точкою.

$$x = (26.5)_{10} = (11010.1)_2,$$

перемістимо двійкову точку на 5 позицій уліво, одержимо

$$x = (0.110101)_2 \cdot 2^5.$$

Враховуючи інформацію про зображення чисел в ЕОМ, можна зробити такі висновки:

1. На ЕОМ можливе зображення не всіх чисел, а лише кінцевого набору раціональних чисел спеціального виду. Ці числа створюють подану множину обчислювальної машини. Для всіх інших чисел  $x$  можливе лише їх наближене зображення з помилкою, яку прийнято назвати помилкою зображення (або помилкою округлення).

Звичайно наближене зображення числа  $x$  в ЕОМ позначають як  $x^* = fl(x)$ . Якщо округлення роблять по доповненню, то границя відносної похибки зображення дорівнює одиниці першого відкинутого розряду мантиси, тобто  $\bar{\delta}(x^*) = \varepsilon_m = 2^{-l}$  (порядок числа не впливає на відносну похибку зображення).

Якщо ж округлення роблять усіканням, то  $\bar{\delta}(x^*) = \varepsilon_m = 2^{l-t}$ . Величина  $\varepsilon_m$  відіграє в обчисленнях на ЕОМ фундаментальну роль; її називають відносною точністю ЕОМ, а також машинною точністю (або машинним епсілоном).

Корисно зазначити, що серед чисел, що надані на ЕОМ, немає не тільки жодного ірраціонального (у тому числі й таких важливих постійних, як  $\pi, e, \sqrt{2}$ ), але й навіть числа 0.1, яке широко використовується в обчисленнях. Справа в тому, що двійковий запис числа 0.1 - нескінченний періодичний дроб:  $0.1 = (0.0001100110011\dots)_2$ . Тому це число завжди подається на ЕОМ приблизно, з похибкою, викликаною необхідністю округлення.

2. Діапазон зміни чисел в ЕОМ обмежений. Насправді, якщо  $\gamma_1 = 1$ , тоді для мантиси  $\mu$  правильними є оцінки  $0.5 \leq |\mu| < 1$ . У той же час для зображення в ЕОМ порядку  $p$  використовується кінцеве число  $(l+1)$  двійкових цифр і тому  $|p| \leq p_{\max} = 2^{l+1} - 1$ . Таким чином, для всіх чисел  $x$ , що можуть бути зображені на

ЕОМ (за винятком нуля), маємо  $0 < X_0 \leq |x| < X_\infty$ , де  $X_0 = 2^{-(p_{\max} + 1)}$ ,  $X_\infty = 2^{p_{\max}}$ . Зазначимо, що діапазон зображення чисел на ЕОМ цілком визначається розрядністю порядку.

3. Всі числа  $x$ , за модулем більші від  $X_\infty$ , не можуть бути зображені на ЕОМ і можуть розглядатися як машинна нескінченність. Спроба одержати таке число призводить до аварійного зупинення ЕОМ з переповнення. Всі числа  $x$ , за модулем менші від  $X_0$ , для обчислювальної машини нерозрізнені й подаються як нуль (машинний нуль). Одержання числа  $x$  такого, що  $|x| < X_0$ , називають зникненням порядку (або антипереповненням). Звичайно при зникненні порядку автоматично вважається  $fl(x) = 0$  й обчислення тривають.

4. На машинній числовій осі (рис. 10.2) числа розташовані нерівномірно. Щільність їх зростає з наближенням до нуля й падає з видаленням від нуля. Щоб переконається в цьому, зазначимо, що відстань від одного машинного числа  $x$  на ЕОМ до іншого дорівнює одиниці останнього розряду мантиси, помноженого на  $2^p$  і дорівнює  $2^{p-t}$ . Оскільки  $t$  зафіксоване, то відстань зменшується зі зменшенням порядку  $p$  і зростає зі збільшенням  $p$ .



Рис. 10.2

#### 4. Арифметичні операції над числами з рухомою точкою

Правила виконання арифметичних операцій у двійковій системі числення надзвичайно прості й легко реалізуються на ЕОМ.

Однак у силу обмеженої розрядності мантиси операції додавання, віднімання, множення і ділення над дійсними числами

в ЕОМ не можуть бути реалізовані точно. Справа в тому, що арифметичні операції над числами, мантиси яких містять  $t$  розрядів, призводять, як правило, до результатів, що містять більше  $t$  розрядів. Округлення результату до  $t$  розрядів і служить головним джерелом похибки. Ігноруючи несуттєві деталі, можна вважати, що результат машинної арифметичної операції збігається з результатом точного виконання тієї самої операції з похибкою, яка приблизно дорівнює похибці округлення.

Зазначимо, що машинні арифметичні операції мають інші властивості, ніж звичайні математичні операції. Наприклад, не виконується відоме правило арифметики "від зміни місць доданків сума не змінюється".

## 5. Подвійна точність

На багатьох обчислювальних машинах можлива реалізація арифметичних операцій над числами, розрядність мантис яких удвічі перевершує стандартну розрядність  $t$ . Це приводить до істотного підвищення машинної точності.

На ЕОМ, де арифметика подвійної точності реалізована апаратно, час виконання програм зростає не більш ніж удвічі (часто цей коефіцієнт ближче до одиниці). Якщо ж реалізація обчислень з подвійною мантисою здійснюється за допомогою програми, то час рахунку збільшується в кілька раз. Оскільки для зберігання числа з подвійною мантисою виділяється два машинні слова, у цьому випадку вдвічі зростає необхідна пам'ять.

Зазначимо, що подвійна точність не ліквідує помилки округлення, а тільки зменшує її значення, а також що для кожної ЕОМ машинний епсілон  $\varepsilon_m$  можна обчислити як мінімальне з чисел  $\varepsilon$ , які мають зображення на ЕОМ, для яких справедлива рівність  $1 + \varepsilon > 1$ .

У зв'язку з вищезазначеним необхідно обов'язково при розв'язанні задач математичного моделювання пам'ятати про коректність її постановки, яка містить у собі задавання множини припустимих вхідних даних  $X$  і множини можливих розв'язків  $Y$ .

## БІБЛІОГРАФІЧНИЙ СПИСОК

1. Советов, Б.Я. Моделирование систем [Текст] / Б.Я. Советов, С.А. Яковлев. - М.: Высш. шк., 1985. - 353 с.
2. Щуп, Т. Решение инженерных задач с использованием ЭВМ. Практическое руководство [Текст] / Т. Щуп. - М.: Мир, 1982. - 238 с.
3. Боглаев, Ю.П. Вычислительная математика и программирование [Текст] / Ю.П. Боглаев. - М.: Высш. шк., 1990. - 544 с.
4. Шенон, Р. Имитационное моделирование систем - искусство и наука [Текст] / Р. Шенон. - М., 1978. - 420 с.
5. Вентцель, Е.С. Исследование операций [Текст] / Е.С. Вентцель. - М., 1972. - 552 с.
6. Маликов, В.Т. Вычислительные методы и применение ЭВМ [Текст] / В.Т. Маликов, Р.Н. Кветный. - К.: Высш. шк., 1989. - 213 с.
7. Петров, А.В. Вычислительная техника в инженерных и экономических расчетах [Текст] / А.В. Петров. - М.: Высш. шк., 1984. - 320 с.
8. Turbo Pascal: алгоритми і програми: Чисельні, методи в фізиці та математиці [Текст]: навч. посібник / А.Б. Бартків, Я.І. Гринчкшин, А.М. Ломакович, Ю.С. Рамський. - К.: Вища школа., 1992. - 248 с.
9. Тарасик, В.П. Математическое моделирование технических систем [Текст]: учеб. для вузов / В.П. Тарасик. - Минск: ДизайнПРО, 2004. - 640 с.



## ОСНОВНІ ВІДОМОСТІ ЛІНІЙНОЇ АЛГЕБРИ

### Функції

**Визначення.** Величина  $y$  називається функцією змінної  $x$ , якщо кожному значенню  $x$  з даної числової множини відповідає за деяким законом цілком визначене значення  $y$ .

Змінна  $x$  називається аргументом або незалежною змінною, величина  $y$  - залежною змінною або функцією.

Областю визначення функції називається множина точок числової осі, у яких функція має цілком визначене значення.

Нехай задана функція  $y = f(x)$ .

**Визначення.** Функція називається безперервною в точці  $x = x_0$ , якщо нескінченно малому збільшенню аргументу в цій точці відповідає нескінченно мале збільшення функції.

**Визначення.** Якщо функція безперервна в будь-якій точці проміжку  $(a, b)$ , то вона називається безперервною в цьому проміжку.

Властивості безперервною на відрізку функції:

1. Безперервна на відрізку  $[a, b]$  функція має хоча б в одній точці цього відрізка своє найбільше значення й хоча б в одній точці своє найменше значення.

2. Якщо на кінцях відрізка  $[a, b]$  безперервна функція має протилежні за знаком значення, то хоча б в одній проміжній точці вона має значення нуль. Якщо таких точок не одна, то їх може бути тільки непарне число.

### Матриці

**1. Визначення.** Прямокутною матрицею називається сукупність чисел, взагалі кажучи, комплексних, розташованих у вигляді прямокутної таблиці, що містить  $n$  рядків і  $m$  стовпців.

Така матриця записується у вигляді

$$\begin{bmatrix} a_{11} & a_{12} & a_{1m} \\ a_{21} & a_{22} & a_{2m} \\ \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{nm} \end{bmatrix},$$

або скорочено у вигляді

$$A = (a_{ij}), i = 1, 2, \dots, n; j = 1, 2, \dots, m.$$

Дві матриці є такими, що дорівнюють одна одній, якщо дорівнюють один одному відповідні елементи.

Матриця, що складається з одного рядка, називається рядком; матриця, що складається з одного стовпця, - стовпцем; матриця  $A = (a)$ , що складається з одного числа, ототожнюється з цим числом. Якщо число  $n$  рядків матриці дорівнює числу її стовпців, то матриця називається квадратною. У цьому випадку число  $n$  називається порядком матриці.

Серед квадратних матриць важливу роль відіграють так звані діагональні матриці, тобто матриці, у яких відмінні від нуля лише ті елементи, що містяться вздовж діагоналі. Діагональні матриці позначаються  $[a_1, a_2, \dots, a_n]$ , так що

$$[a_1, a_2, \dots, a_n] = \begin{bmatrix} a_1 & 0 & \dots & 0 \\ 0 & a_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_n \end{bmatrix}.$$

Якщо всі числа  $a_i$  при цьому рівні між собою, матриця називається скалярною:

$$[a] = \begin{bmatrix} a & 0 & \dots & 0 \\ 0 & a & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a \end{bmatrix}.$$

У випадку якщо  $a = 1$  - одиничною:

$$E = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix} = (\delta_{ij}),$$

де  $\delta_{ij}$  так званий символ Кронекера, тобто  $\delta_{ij} = 0$  при  $i \neq j$ ,  $\delta_{ji} = 1$ .

Нарешті, матриця, всі елементи якої дорівнюють нулю, називається нульовою. Ми будемо її позначати символом 0.

Переставивши в матриці

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & a_{2m} \\ a_{1m} & a_{2m} & \dots & a_{nm} \end{bmatrix}$$

рядки зі стовпцями, ми одержимо так звану *транспоновану* матрицю:

$$A' = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{n1} \\ a_{12} & a_{22} & \dots & a_{n2} \\ \dots & \dots & \dots & a_{2m} \\ a_{1m} & a_{2m} & \dots & a_{nm} \end{bmatrix}$$

Квадратна матриця  $A$  дорівнює транспонованій  $A'$  в тому й тільки в тому випадку, коли вона *симетрична*, тобто якщо  $a_{ij} = a_{ji}$ .

Очевидно, що матриця, транспонована з рядком, є стовпцем, складеним з тих самих елементів. Цю обставину ми часто будемо використовувати для зручності запису стовпців.

*Визначником* квадратної матриці називається визначник, елементи якого дорівнюють елементам матриці. Визначник матриці  $A$  позначається через  $|A|$ .

Будь-який визначник, рядки й стовпці якого "укладаються" у рядки й стовпці матриці, називається *мінором* цієї матриці. Мінор порядку  $k$  матриці  $A$  є визначником  $k$ -го порядку, складений з елементів, що перебувають на перетинанні деяких  $k$  рядків і  $k$  стовпців матриці  $A$  в їхньому природньому розташуванні.

*Рангом* матриці  $A$  називається максимальний порядок відмінних від нуля мінорів матриці. Інакше, рангом матриці називається таке число  $r$ , при якому серед мінорів матриці існує мінор порядку  $r$  нерівний нулю, а всі мінори порядку  $r + 1$  й вище дорівнюють нулю або не можуть бути складені.

**2. Дії з матрицями.** *Добутком матриці  $A(a_{ij})$  і числа  $\alpha$*  називається матриця, елементи якої отримані з елементів матриці  $A$  множенням на число  $\alpha$ .

$$\alpha A = \begin{bmatrix} \alpha a_{11} & \alpha a_{12} & \dots & \alpha a_{1m} \\ \dots & \dots & \dots & \dots \\ \alpha a_{n1} & \alpha a_{n2} & \dots & \alpha a_{nm} \end{bmatrix}.$$

*Сумою двох прямокутних матриць, що  $A = (a_{ij})$  й  $B = (b_{ij})$*  мають однакове число як рядків, так і стовпців, називається матриця  $C$ , елементи якої дорівнюють сумах відповідних елементів матриць  $A$  і  $B$ , тобто

$$A + B = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & \dots & a_{1m} + b_{1m} \\ a_{21} + b_{21} & a_{22} + b_{22} & \dots & a_{2m} + b_{2m} \\ \dots & \dots & \dots & \dots \\ a_{n1} + b_{n1} & a_{n2} + b_{n2} & \dots & a_{nm} + b_{nm} \end{bmatrix}.$$

*Множення матриць  $A$  і  $B$*  визначається тільки в припущенні, що число стовпців матриці  $A$  дорівнює числу рядків матриці  $B$ . У цьому припущенні елементи добутку  $C$  визначаються в такий спосіб: елемент  $i$ -го рядка  $j$ -го стовпця матриці  $C$  дорівнює сумі добутків елементів  $i$ -й рядка матриці  $A$  на відповідні елементи  $j$ -го стовпця матриці  $B$ . Таким чином,

$$AB = \begin{bmatrix} a_{11} & a_{12} & a_{1m} \\ a_{21} & a_{22} & a_{2m} \\ \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{nm} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{1p} \\ b_{21} & b_{22} & b_{2p} \\ \dots & \dots & \dots \\ b_{m1} & b_{m2} & b_{mp} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{1p} \\ c_{21} & c_{22} & c_{2p} \\ \dots & \dots & \dots \\ c_{m1} & c_{m2} & c_{mp} \end{bmatrix},$$

де  $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{im}b_{mj}$  ( $i = 1, \dots, n; j = 1, \dots, p$ ).

Зазначимо, що добуток двох прямокутних матриць є знову прямокутною матрицею, число рядків якої дорівнює числу рядків першої матриці, а число стовпців дорівнює числу стовпців другої матриці. Так, наприклад, добуток квадратної матриці на матрицю, що складається з одного стовпця, є матрицею з одного стовпця.

### *Зворотна матриця*

Квадратна матриця  $A = (a_{ij})$  називається неособливою або невиродженою, якщо її визначник не дорівнює нулю; а якщо дорівнює нулю, то матриця називається особливою.

Введемо тепер поняття зворотної матриці. Матрицю  $B$  назвемо зворотною до квадратної матриці  $A$ , якщо

$$AB = E.$$

### *Поліном від матриці*

Визначимо тепер додатний степінь квадратної матриці:

$$A^n = \overbrace{A \dots A}^{n \text{ раз}}.$$

У силу асоціативного закону байдуже, як у цьому добутку розставити дужки, і тому ми їх опускаємо. З визначення зрозуміло, що

$$\left. \begin{aligned} A^n A^m &= A^{n+m} \\ (A^n)^m &= A^{nm} \end{aligned} \right\}.$$

Вираз вигляду

$$\alpha_0 A^n + \alpha_1 A^{n-1} + \dots + \alpha_n E,$$

де  $\alpha_0, \alpha_1, \dots, \alpha_n$  - комплексні числа, називається поліномом матриці або матричним поліномом. Поліном від матриці можна розглядати як результат підстановки матриці замість змінної  $t$  в алгебраїчний поліном.

$$f(t) = \alpha_0 t^n + \alpha_1 t^{n-1} + \dots + \alpha_n.$$

### *Елементарні перетворення матриць*

Часто з матрицями потрібно робити такі операції:

- множення елементів якогось рядка на число;
- додавання до елементів якого-небудь рядка чисел, пропорційних елементам якого-небудь попереднього рядка;
- додавання до елементів якого-небудь рядка чисел, пропорційних елементам якого-небудь наступного рядка.

Іноді такі перетворення доводиться робити зі стовпцями. Перетворення зазначеного виду будемо називати елементарними перетвореннями матриць.

## ТЕСТОВІ ПИТАННЯ

### Тема 1. Загальні питання математичного моделювання

#### 1. Чим викликана неусувна похибка?

а) тим, що математична модель досліджуваного об'єкта ніколи не враховує всіх без винятку явищ, що впливають на стан об'єкта, і тим, що вхідні параметри (числа або функції) вимірюються з якою-небудь помилкою;

б) тим, що будь-які арифметичні операції над числами виконуються за наявності обмеженої кількості використовуваних для запису чисел розрядів позиційної системи числення;

в) тим, що в результаті застосування чисельного методу можуть бути отримані не точні, а наближені значення шуканої функції, навіть якщо всі запропоновані методом обчислення пророблені абсолютно точно.

#### 2. Чим викликана похибка округлення при чисельному розв'язанні поставленої задачі?

а) тим, що математична модель досліджуваного об'єкта не може враховувати всі без винятку явища, що впливають на стан об'єкта;

б) тим, що будь-які арифметичні операції над числами виконуються за наявності обмеженої кількості використовуваних для запису чисел розрядів позиційної системи вираховування;

в) тим, що в результаті застосування чисельного методу можуть бути отримані не точні, а наближені значення шуканої функції, навіть якщо всі запропоновані методом обчислення пророблені абсолютно точно.

### 3. Чим викликана похибка методу при чисельному розв'язанні поставленої задачі?

а) тим, що в результаті застосування чисельного методу можуть бути отримані не точні, а наближені значення шуканої функції, навіть якщо всі запропоновані методом обчислення пророблені абсолютно точно;

б) тим, що математична модель досліджуваного об'єкта не може враховувати всі без винятку явища, що впливають на стан об'єкта;

в) тим, що будь-які арифметичні операції над числами виконуються за наявності обмеженої кількості використовуваних для запису чисел розрядів позиційної системи виражування.

### 4. Як визначити неусувну похибку?

а) найчастіше метод завдання математичної задачі буває наближеним. Це означає, що в результаті застосування методу можуть бути отримані не точні, а наближені значення шуканої функції, навіть якщо всі запропоновані методом обчислення виконані абсолютно точно. Чисельне розв'язання в цьому випадку відрізняється від розв'язання вихідної задачі, а помилка, внесена в розв'язання математичної задачі застосуванням наближеного методу, називається неусувною похибкою;

б) математична постановка будь-якої прикладної задачі містить неусувну похибку у зв'язку з двома обставинами. По-перше, математична модель реального об'єкта ніколи не враховує всіх без винятку явищ, що впливають на стан цього об'єкта, тобто завжди доводиться жертвувати деякими факторами, які для даного завдання можна вважати несуттєвими. По-друге, у рівняння задачі входять деякі параметри - числа або функції. Значення цих параметрів виходять з результату вимірів різних характеристик об'єкта, що моделюється, виробляються з помилкою;

в) нехай  $a^*$  – точне,  $a$  – наближене значення деякого числа. Неусувною похибкою наближення  $a$  називається така величина  $\delta_a$ , що  $a - a^* \leq \delta_a$ .



## **5. Завдання моделювання полягає:**

- а) у відтворенні явища подібного до оригіналу і його дослідженні;
- б) виборі мови програмування для реалізації алгоритму;
- в) налагодженню математичної моделі.

## **6. За характером досліджуваних процесів моделі не можуть бути:**

- а) статичними;
- б) динамічними;
- в) епістолярними;
- г) детермінованими;
- д) стохастичними;
- е) дискретними;
- ж) безперервними.

## **7. Під альтернативами розуміють:**

- а) можливі шляхи або засоби досягнення мети досліджуваної системи;
- б) кількісні оцінки ступеня досягнення мети;
- в) числові характеристики властивостей системи.

## **8. Під показником ефективності системи розуміють:**

- а) рекомендації з досягнення мети досліджуваної системи;
- б) витрати ресурсів, які оцінюються показником вартості системи;
- в) кількісні оцінки ступеня досягнення мети.

## **9. Методи наукового дослідження поділяються:**

- а) на аналіз і синтез;
- б) вибір і розрахунок;
- в) алгоритмізацію й програмування;
- г) розроблення й впровадження.

**10. Що не належать до основної вимоги, якій повинна відповідати модель?**

- а) показність;
- б) критичність;
- в) унімодалність;
- г) правильний облік випадковостей і невизначеностей;
- д) максимально можлива простота.

## **Тема 2. Комп'ютерне моделювання й обчислювальний експеримент**

### **1. Сформулюйте постановку задачі інтерполяції функції:**

- а) потрібно обчислити похідні від функцій, які задані у табличному вигляді;
- б) потрібно знайти значення функції  $f(x)$ ,  $x \neq x_i$  ( $i = 0, 1, \dots, n$ ), якщо відомі вузли інтерполяції  $x_i$  ( $i = 0, 1, \dots, n$ ) і значення функції  $f(x)$  у цих вузлах;
- в) потрібно визначити припустиму похибку аргументів за припустимою похибкою функції.

### **2. Що є мірою якості апроксимації функції $f(x)$ поліномом $P_m(x)$ у методі найменших квадратів?**

- а) максимум модуля різниці  $f(x_i)$  і  $P_m(x_i)$  ( $i = 1, 2, \dots, n$ );
- б) сума  $\sum_{i=1}^n [y_i - P_m(x_i)]^2$ ;
- в) сума  $\sum_{i=1}^n \sqrt{[y_i - P_m(x_i)]}$ .

### **3. Яку функцію називають апроксимуючою, якщо для кінцевої множини значень аргументу $x_0, x_1, \dots, x_n$ відомі табличні значення функцій $f(x_0), f(x_1), \dots, f(x_n)$ ?**

- а) функцію  $\varphi(x)$ , розрахунки якої за її аналітичною формулою або збігаються, або в певному розумінні наближаються до даних значень функцій при відповідному значенні аргументу;

б) функцію  $\varphi(x)$ , похідні від якої в заданих точках дорівнюють похідним функції  $f(x)$ ;

в) функцію  $\varphi(x)$ , значення якої відрізняються від даних значень функцій на постійну величину.

#### 4. Назвіть переваги й недоліки інтерполяційних формул Лагранжа:

а) перевага - метод дуже простий у розумінні й організації обчислювального процесу. Основний недолік методу - при збільшенні числа вузлів і відповідно ступеня інтерполяційного многочлена Лагранжа його потрібно будувати заново;

б) перевага - метод належить до ітераційних методів і має найбільшу точність інтерполяції. Основний недолік методу - повільна швидкість збіжності, що призводить до значних витрат машинного часу;

в) перевага - використання многочленів невисокого порядку й внаслідок цього мале накопичування похибок у процесі обчислень. Основний недолік методу - із числа методів інтерполяції він найбільш складний в організації обчислювального процесу.

#### 5. У чому сутність методу найменших квадратів?

а) метод полягає в наступному. Увесь відрізок інтерполяції розбивають на часткові відрізки й на кожному з часткових відрізків приблизно заміняють функцію  $f(x)$ , що інтерполюється, многочленом невисокого ступеня. Для того щоб не виникало розривів похідної в місцях зчленування, на кожному частковому відрізку ступінь полінома береться «із запасом», а виникаюча свобода у виборі коефіцієнтів поліномів використовується для сполучення похідних на границях ділянок;

б) метод полягає в тому, що будується поліном, сума квадратів відхилень якого від табличних значень функції, що інтерполюється,  $y_i = f(x_i)$  мінімальна, тобто за міру якості апроксимації функції  $f(x)$  поліномом  $P_m(x)$  у вузлах  $x_i$  вважають

$$\text{суму } \sum_{i=1}^n [y_i - P_m(x_i)]^2 ;$$

в) метод полягає в тому, що будується поліном вигляду

$$\sum_{i=1}^n y_i \prod_{\substack{j=1 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j},$$

що в точках  $x_i$ , які названі вузлами, має значення

функції  $f(x_i)$ , яка інтерполюється.

## 6. Що є методом найменших квадратів?

а) метод полягає в тому, що будується поліном, сума квадратів відхилень якого від табличних значень функції, що інтерполюється,  $y_i = f(x_i)$  мінімальна. Таким чином, за міру якості апроксимації функції  $f(x)$  поліномом  $P_m(x)$  у вузлах  $x_i$  вважають

$$\text{суму } \sum_{i=1}^n [y_i - P_m(x_i)]^2 ;$$

б) задана система лінійних рівнянь яким-небудь чином приводиться до еквівалентного вигляду. Виходячи з довільного початкового вектора будується ітераційний процес. При виконанні достатніх умов збіжності одержуємо послідовність векторів, що неорганічно наближаються до точного розв'язку;

в) згідно з даним методом обчислення інтеграла  $I$  за обраною квадратурною формулою роблять двічі: спочатку інтеграл  $Ih$  з якимсь кроком  $h$ , потім інтеграл  $Ih/2$  із кроком  $h/2$ , а потім порівнюють їх. Якщо виявиться, що  $Ih - Ih/2 < \varepsilon$ , де  $\varepsilon$  - припустима похибка, то вважаються  $I \approx Ih / 2$ . Якщо ж  $Ih - Ih/2 \geq \varepsilon$ , то розрахунки повторюють із кроком  $h/4$  і т. д.

## 7. Основні області застосування задачі інтерполяції функцій:

а) до інтерполяції функцій найчастіше прибігають, коли потрібно визначити припустиму похибку аргументів за припустимою похибкою функції. Інтерполяцію застосовують і у випадку, коли необхідно обчислити похибку функції декількох змінних при заданих похибках аргументів;

б) до задачі інтерполяції функцій прибігають, коли доводиться обчислювати похідні від функцій, які задані у вигляді таблиці, або коли безпосередньо диференціювати функцію важко.

Інтерполяцію застосовують і у випадку, коли необхідно обчислити похідні від функцій, що мають розрив 2-го роду;

в) до задачі інтерполяції функцій прибігають, коли доводиться обчислювати значення функції в проміжних точках, при цьому дана функція задана в табличному вигляді й аналітичний вираз функції невідомий. Інтерполяцію застосовують і у випадку, коли аналітичний вигляд функції відомий, але складний і вимагає великого обсягу обчислень для визначення окремих значень функції.

### **8. Що таке регресійна модель?**

- а) функція, яка проходить через задані точки;
- б) функція, яка проходить поблизу заданих точок;
- в) функція, яка усереднює експериментальні значення;
- г) функція, графік якої співпадає з експериментальними значеннями.

### **9. При методі найменших квадратів:**

- а) визначається функція, для якої квадрат суми відхилень її значень від заданих значень мінімальний;
- б) визначається функція, для якої квадрати відхилень її значень від заданих значень мінімальні;
- в) визначається квадратична функція, для якої сума відхилень її значень від заданих значень мінімальна;
- г) визначається функція, для якої сума квадратів відхилень її значень від заданих значень мінімальна.

### **10. Задача апроксимації поліномом 2-го порядку складається з таких етапів:**

- а) 1. За координатами точок масивів  $\{x_i\}$ ,  $\{y_i\}$ , обчислюються коефіцієнти системи лінійних алгебраїчних рівнянь (СЛАР) щодо невідомих  $A$ ,  $B$ ,  $C$ , що входять у функцію  $f(x) = Ax^2 + Bx + C$ ;

2. Розв'язується СЛАР, і в результаті визначаються значення  $A, B, C$ ;
  3. Обчислюються значення апроксимуючої кривої в досліджуваному діапазоні;
- б) 1. Розв'язується СЛАР, і в результаті визначаються значення  $A, B, C$ ;
2. За координатами точок масивів  $\{x_{ij}\}, \{y_{ij}\}$ , обчислюються коефіцієнти СЛАР щодо невідомих  $A, B, C$ , що входять у функцію  $f(x) = Ax^2 + Bx + C$ ;
  3. Обчислюються значення апроксимуючої кривої в досліджуваному діапазоні;
- в) 1. Обчислюються значення апроксимуючої кривої в досліджуваному діапазоні;
2. За значеннями невідомих  $A, B, C$  обчислюють координати точок масивів  $\{x_{ij}\}, \{y_{ij}\}$ ;
  3. Розв'язується СЛАР, і в результаті визначаються значення  $A, B, C$ .

## 11. Інтерполяційний поліном Лагранжа зображений:

- а) на рис. Д.2.1;
- б) рис. Д.2.2;
- в) рис. Д.2.3.

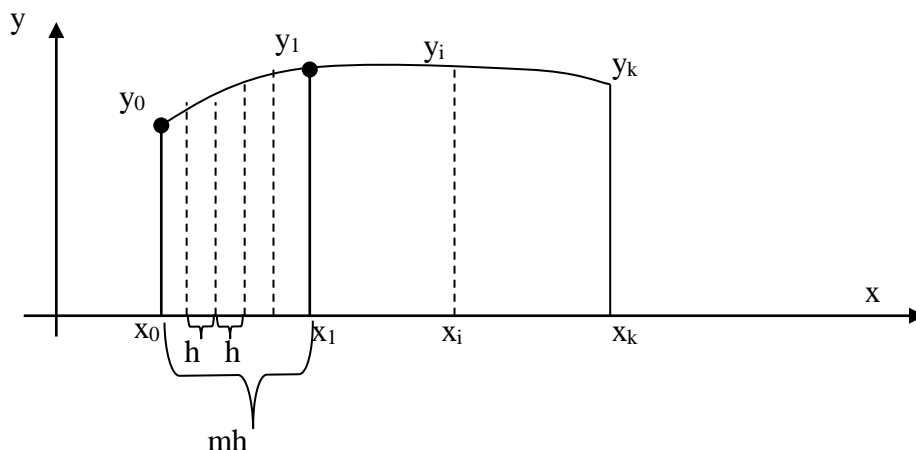


Рис. Д.2.1

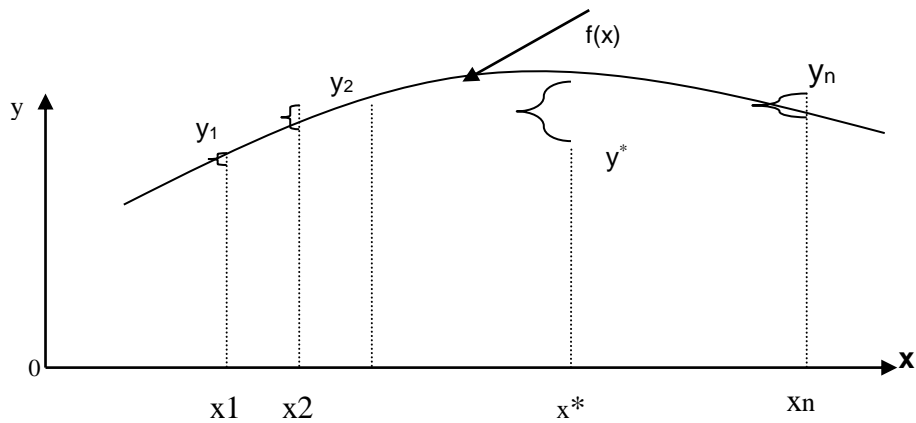


Рис. Д.2.2

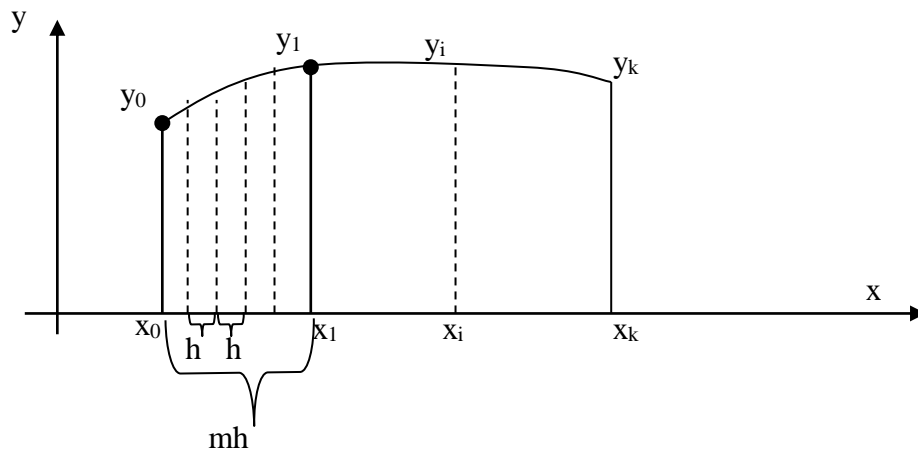


Рис. Д.2.3

**12. Якщо функція, що апроксимує, є лінійною (поліном 1-го порядку), то вона має вигляд:**

- а)  $f(x) = a + bx$ ;
- б)  $f(x) = ax^2 + bx + c$ ;
- в)  $f(x) = 1 / (a + bx)$ .

### Тема 3. Розв'язання нелінійних рівнянь

**1. Назвіть основні етапи процесу знаходження кореня нелінійного рівняння:**

а) на першому етапі ліва частина нелінійного рівняння  $f(x) = 0$  апроксимується на інтервалі  $[a, b]$  інтерполяційним багаточленом.

На другому етапі, використовуючи задане початкове наближення, будується ітераційний процес, який дозволяє уточнити значення кореня, що відшукується;

б) на першому етапі перевіряється виконання достатніх умов збіжності.

На другому етапі нелінійне рівняння заміняється на інтервалі  $[a, b]$  еквівалентним рівнянням. На третьому етапі будується ітераційний процес, що дозволяє визначити значення кореня нелінійного рівняння;

в) на першому етапі вивчається розташування коренів і проводиться їх роз'єднання, тобто знаходиться деякий інтервал  $[a, b]$  осі  $Ox$ , усередині якого перебуває один корінь, і немає інших розв'язків даного нелінійного рівняння.

На другому етапі, використовуючи задане початкове наближення, будується ітераційний процес, що дозволяє уточнити значення кореня даного нелінійного рівняння.

**2. При розв'язанні нелінійного рівняння  $f(x) = 0$  на інтервалі  $[a, b]$  методом простої ітерації нелінійне рівняння замінюється еквівалентним рівнянням:**

- а)  $x = \varphi(x)$ ;
- б)  $f(x) = f((a+b)/2+x)$ ;
- в)  $x = \varphi(x) + f(x)$ .

**3. Що є методом ділення відрізка навпіл?**

а) для знаходження кореня нелінійного рівняння  $f(x) = 0$  потрібно, щоб на кінцях інтервалу  $[a, b]$  функція  $f(x)$  мала ненульові значення протилежного знака. Ітераційна процедура



полягає в переході від такого інтервалу до нового інтервалу, що збігається з однією з половин попереднього та має таку саму властивість. Процес закінчується, коли довжина знов отриманого інтервалу стане менше заданої точності  $\varepsilon$ , і коренем рівняння приблизно вважається значення координати середини цього інтервалу;

б) згідно з даним методом загальна похибка обчислення інтеграла розглядається як сума похибки усікання  $\varepsilon_s$  і похибки округлення  $\varepsilon_p$ . Оскільки зі зменшенням кроку розрахунків  $h$  погрішність  $\varepsilon_s$  зменшується, а  $\varepsilon_p$  зростає, то існує оптимальний крок  $h$ , який визначено таким чином, щоб  $\varepsilon_s$  дорівнювало приблизно половині  $\varepsilon_p$ ;

в) будується система рівновіддалених точок  $x_i = x_0 + i \cdot h$  ( $i = 0, 1, 2, \dots$ ) при достатньо малому кроці  $h$ . Наближені значення  $y(x_i)$ , розв'язок диференціального рівняння  $y' = f(x, y)$  знаходяться послідовно за формулою  $y_{i+1} = y_i + h \cdot f(x_i, y_i)$ .

**4. Рівняння вигляду  $f(x)=0$  називається алгебраїчним, якщо  $f(x)$  може бути представлено у вигляді**

а)  $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ ;

б)  $f(x) = f(\log(x), \ln(x), \dots)$ ;

в)  $f(x) = f(\cos(x), \sin(x), \dots)$ .

**5. При відділенні коренів інтервал існування функції  $[a, b]$  розбивають на рівні відрізки довжиною  $\Delta x$  і обчислюють значення функції на кінцях цих відрізків ( $f(a)$  і  $f(b)$  відповідно).**

**Умовою наявності кореня буде:**

а)  $f(a) \cdot f(b) \leq 0$ ;

б)  $f(a) \cdot f(b) < 0$ ;

в)  $f(a) \cdot f(b) = 0$ ;

г)  $f(a) \cdot f(b) > 0$ .

**6. Опишіть метод ділення відрізка навпіл (знаходження коренів нелінійного рівняння):**

а) нелінійне рівняння  $f(x) = 0$  на інтервалі  $[a, b]$  замінюється еквівалентним рівнянням  $x = \varphi(x)$ . Ітерації створюються за правилом  $x_{k+1} = \varphi(x_k)$ , ( $k = 0, 1, \dots$ ), причому задається початкове наближення  $x_0$ . Якщо послідовність чисел  $x_k$  має границю при  $k \rightarrow \infty$ , то ця границя є коренем рівняння  $x = \varphi(x)$ .

б) для знаходження кореня нелінійного рівняння  $f(x) = 0$  потрібно, щоб на кінцях інтервалу  $[a, b]$  функція  $f(x)$  мала ненульові значення протилежного знака. Ітераційна процедура полягає в переході від такого інтервалу до нового інтервалу, що збігається з однією з половин попереднього та має таку саму властивість. Процес закінчується, коли довжина знов отриманого інтервалу стане менш заданої точності  $\varepsilon$ , і коренем рівняння приблизно вважається значення координати середини цього інтервалу.

в) для знаходження кореня нелінійного рівняння  $f(x) = 0$  потрібно, щоб функція  $f(x)$  мала на інтервалі  $[a, b]$  безперервні похідні 1-го й 2-го порядків, які зберігають на  $[a, b]$  постійний знак. Для початку обчислень необхідне задавання одного початкового наближення  $x_0$ . Наступні наближення знаходяться за формулою  $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$ , ( $k = 0, 1, \dots$ ).

**7. У методі половинного ділення передбачається, що функція на кінцях інтервалу  $[a, b]$  має значення різних знаків і  $t = \frac{a+b}{2}$ . На кожній ітерації аналізується знак:**

а)  $f(t-a) \cdot f(b-t)$ ;

б)  $f(a) \cdot f(t)$ ;

в)  $f(a) \cdot f(b)$ ;

г)  $f(b-a) \cdot f(t)$ .

**8. Для збіжності процесу знаходження коренів нелінійного рівняння  $x=f(x)$  методом простої ітерації потрібно, щоб на інтервалі, на якому уточнюється корінь, виконувалася нерівність:**

- а)  $|f'(x)| < 1$ ;
- б)  $f'(x) > 0$ ;
- в)  $|f'(x)| > 1$ ;
- г)  $0 < f'(x) < 1$ .

**9. Рівняння вигляду  $f(x)=0$  називається трансцендентним, якщо  $f(x)$  може бути подано у вигляді:**

- а)  $f(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ ;
- б)  $f(x) = f(\log(x), \ln(x), \cos(x), \sin(x), e^x, \dots)$ ;
- в)  $f(x) = f_1(x) + f_2(x)$ .

**10. Метод простих ітерацій для розв'язок нелінійного рівняння - це:**

а) нелінійне рівняння  $f(x) = 0$  на інтервалі  $[a, b]$  замінюється еквівалентним рівнянням  $x = \varphi(x)$ . Ітерації створюються за правилом  $x_{k+1} = \varphi(x_k)$  ( $k = 0, 1, \dots$ ), причому задається початкове наближення  $x_0$ . Якщо послідовність чисел  $x_k$  має границю при  $k \rightarrow \infty$ , то ця границя є коренем рівняння  $x = \varphi(x)$ ;

б) для знаходження кореня нелінійного рівняння  $f(x) = 0$  потрібно, щоб на кінцях інтервалу  $[a, b]$  функція  $f(x)$  мала ненульові значення протилежного знака. Ітераційна процедура полягає в переході від такого інтервалу до нового інтервалу, що збігається з однією з половин попереднього та має таку саму властивість. Процес закінчується, коли довжина знов отриманого інтервалу стане менше заданої точності  $\varepsilon$ , і коренем рівняння приблизно вважається значення координати середини цього інтервалу;

в) для знаходження кореня нелінійного рівняння  $f(x) = 0$  потрібно, щоб функція  $f(x)$  мала на інтервалі  $[a, b]$  безперервні похідні 1-го й 2-го порядків, які зберігають на  $[a, b]$  постійний знак. Для початку обчислень необхідне задавання одного

початкового наближення  $x_0$ . Наступні наближення знаходяться за формулою  $x_{k+1} = x_k - f(x_k) / f'(x_k)$ , ( $k = 0, 1, \dots$ ).

## Тема 4. Методи чисельного інтегрування

### 1. У чому полягає сутність методів чисельного інтегрування функцій?

а) заміна підінтегральної функції  $f(x)$  допоміжною, інтеграл від якої легко обчислюється в елементарних функціях;

б) при заданій кількості інтервалів розбиття слід розташувати їхні кінці так, щоб одержати найвищу точність інтегрування;

в) з підінтегральної функції  $f(x)$  виділяють деяку функцію  $g(x)$ , яка має ті самі особливості, що функція  $f(x)$ , елементарно інтегровану на даному проміжку, і таку, щоб різниця  $f(x) - g(x)$  мала потрібне число похідних.

### 2. Обчислення визначеного інтеграла за формулами прямокутників -це:

а) відрізок інтегрування  $[a, b]$  розбивається на  $n$  рівних інтервалів. У межах кожного інтервалу  $[x_i, x_{i+1}]$  підінтегральна функція  $f(x)$  замінюється інтерполяційним многочленом Лагранжа першого ступеня з вузлами  $x_i, x_{i+1}$ , що відповідає заміні кривої на січну. Інтеграл по  $[a, b]$  обчислюється як сума інтегралів по всіх часткових відрізках;

б) у квадратурних формулах  $\int_a^b f(x)dx = \sum_{k=1}^m A_k f(x_k) + R$

коефіцієнти  $A_k$  і абсциси  $x_k$  підбираються так, щоб формули були точні для многочленів найвищого можливого ступеня  $M$ . При  $m$  вузлах точно інтегруються всі багаточлени ступеня  $M \leq 2m - 1$ . Коефіцієнти  $A_k$  і абсциси  $x_k$  знаходять із системи  $2m-1$  нелінійних рівнянь;

в) відрізок інтегрування  $[a, b]$  розбивають на часткові відрізки  $[x_i, x_{i+1}]$  рівної довжини. На кожному відрізку  $[x_i, x_{i+1}]$

підінтегральна функція  $f(x)$  заміняється на постійну величину  $f(x_i)$  або  $f(x_{i+1})$  і інтеграл по  $[a, b]$  обчислюється як сума інтегралів по всіх часткових відрізках.

**3. При обчисленні інтеграла методом Сімпсона підінтегральна функція інтерполюється:**

- а) поліномом нульового порядку;
- б) поліномом першого порядку;
- в) поліномом другого порядку;
- г) поліномом  $n$ -го порядку.

**4. Обчислення визначеного інтеграла за формулою трапеції – це:**

а) відрізок інтегрування  $[a, b]$  розбивають на часткові відрізки  $[x_i, x_{i+1}]$  однакової довжини. На кожному відрізку  $[x_i, x_{i+1}]$  підінтегральна функція  $f(x)$  заміняється на постійну величину  $f(x_i)$  і інтеграл по  $[a, b]$  обчислюється як сума інтегралів по всіх часткових відрізках;

б) відрізок інтегрування  $[a, b]$  розбивається на  $n$  рівних інтервалів. У межах кожного інтервалу  $[x_{i-1}, x_{i+1}]$  підінтегральна функція  $f(x)$  заміняється інтерполяційним багаточленом Лагранжа першого ступеня з вузлами  $[x_{i-1}, x_{i+1}]$ , що відповідає заміні кривої на січну. Інтеграл по  $[a, b]$  обчислюється як сума інтегралів по всіх часткових відрізках;

в) відрізок інтегрування  $[a, b]$  розбивається на  $n$  рівних інтервалів. У межах кожного інтервалу  $[x_{i-1}, x_{i+1}]$  підінтегральна функція  $f(x)$  заміняється інтерполяційним багаточленом другого ступеня з вузлами  $x_{i-1}, x_i$  і  $x_{i+1}$ . Інтеграл по  $[a, b]$  обчислюється як сума інтегралів по всіх часткових відрізках.

**5. Числові методи інтегрування основані:**

а) на точному обчисленні площини фігури, обмеженої графіком функції і межами інтервалу змінення її аргументу;

б) приблизному обчисленні площини фігури, обмеженої графіком функції, межами інтервалу змінення її аргументу та віссю абсцис;

- в) обчисленні інтеграла аналітичним способом;
- г) обчисленні інтеграла за допомогою розкладання підінтегральної функції на стандартні функції, що можуть бути проінтегровані.

**6. Який вираз відповідає обчисленню інтеграла методом трапецій?**

- а)  $D=h*(y_1+y_2+\dots+y_n)$ ;
- б)  $D=h*(f(a)+y_2+\dots+y_{n-1}+f(b))$ ;
- в)  $D=h*(f(a)/2+y_2+\dots+y_{n-1}+f(b)/2)$ ;
- г)  $D=h/3*(f(a)+4y_2+2y_3+\dots+4y_{n-2}+2y_{n-1}+f(b))$ .

**7. При обчисленні інтеграла методом прямокутників підінтегральна функція інтерполюється:**

- а) поліномом нульового порядку;
- б) поліномом першого порядку;
- в) поліномом іншого порядку;
- г) поліномом  $n$ -го порядку.

**8. Обчислення визначеного інтеграла за формулою Сімпсона - це:**

а) відрізок інтегрування  $[a, b]$  розбивають на часткові відрізки  $[x_i, x_{i+1}]$  рівної довжини. На кожному відрізку  $[x_i, x_{i+1}]$  підінтегральна функція  $f(x)$  замінюється на постійну величину  $f(x_i)$  і інтеграл по  $[a, b]$  обчислюється як сума інтегралів по всіх часткових відрізках;

б) у квадратурних формулах 
$$i \int_a^b f(x) dx = \sum_{k=1}^m C_k f(t_k) + R$$

коефіцієнти  $C_k$  і абсциси  $tk$  підбираються так, щоб формули були точні для багаточленів найвищого можливого ступеня  $M$ . При  $m$  вузлах точно інтегруються всі багаточлени ступеня  $M \leq 2m - 1$ . Коефіцієнти  $C_k$  і абсциси  $tk$  знаходяться з системи  $2m-1$  нелінійних рівнянь;

в) відрізок інтегрування  $[a, b]$  розбивається на  $n$  рівних інтервалів. У межах кожного інтервалу  $[x_{i-1}, x_{i+1}]$  підінтегральна функція  $f(x)$  замінюється інтерполяційним багаточленом другого ступеня з вузлами  $x_{i-1}, x_i$  і  $x_{i+1}$ . Інтеграл по  $[a, b]$  обчислюється як сума інтегралів по всіх часткових відрізках.

**9. При обчисленні інтеграла методом трапецій підінтегральна функція інтерполюється:**

- а) поліномом нульового порядку;
- б) поліномом першого порядку;
- в) поліномом іншого порядку;
- г) поліномом  $n$ -го порядку.

**10. Який вираз відповідає обчисленню інтеграла методом прямокутників?**

- а)  $D=h*(y_1+y_2+\dots+y_n)$ ;
- б)  $D=h/2*(f(a)+y_2+\dots+y_{n-1}+f(b))$ ;
- в)  $D=h*(f(a)/2+y_2+\dots+y_{n-1}+f(b)/2)$ ;
- г)  $D=h/3*(y_1+y_2+\dots+y_n)$ .

**11. Наведіть іншу назву визначеного інтеграла:**

- а) інтеграл Рімана;
- б) інтеграл Ейлера;
- в) інтеграл Лобачевського;
- г) невластний інтеграл.

**12. Формула обчислення визначеного інтеграла методом Сімпсона має вигляд:**

- а)  $D=h*(f(a)+4y_2+2y_3+\dots+4y_{n-2}+2y_{n-1}+f(b))$ ;
- б)  $D=h/6*(f(a)+4y_2+2y_3+\dots+4y_{n-2}+2y_{n-1}+f(b))$ ;
- в)  $D=h/3*(f(a)+4y_2+2y_3+\dots+4y_{n-2}+2y_{n-1}+f(b))$ .

## **Тема 5. Метод динамічного програмування**

**1. Для розв'язання яких задач застосовується метод динамічного програмування?**

- а) дослідження динамічних систем;
- б) оптимізації параметрів динамічних систем;
- в) оптимізації управління рухом системи;
- г) оптимізації процесів, які складені з окремих кроків.

**2. Управління являє собою:**

- а) організацію процесу, яка забезпечує досягнення певної мети;
- б) механізм збору інформації й виконання ухваленого рішення;
- в) умову протікання процесів.

**3. Що не є етапом процесу управління:**

- а) збір і обробка інформації з метою оцінки ситуації, що склалася;
- б) ідеалізація й спрощення реального процесу;
- в) ухвалення рішення про найбільш доцільні дії;
- г) виконання ухваленого рішення.

**4. Оптимальним управлінням називається:**

- а) спосіб управління, який задовольняє всі поставлені обмеження і перетворює на мінімум (максимум) критерій якості управління;
- б) метод математичного програмування;
- в) цільова функція, що включає обмеження.



## 5. Метод динамічного програмування призначений:

- а) для пошуку оптимальних розв'язків задач оптимізації багатокрокових процесів прийняття рішень;
- б) формалізації мети управління, яка виражена через критерій якості управління;
- в) обмеження ресурсів або інших величин, які використовуються в управлінні.

6. Якщо процес, який оптимізується, складається з окремих кроків ( $m$  - число кроків), то загальний виграш або програш ( $W$ ) може бути представлений як ( $\omega_i$  - виграш або програш на  $i$ -му кроці):

а)  $W = \prod_{i=1}^m \omega_i$ ;

б)  $W = \sum_{i=1}^m \omega_i$ ;

в)  $W = \int_1^m \omega(x) dx$ .

7. Критерій  $W$  вигляду  $W = \sum_{i=1}^m \omega_i$ , називається:

- а) адитивним;
- б) комутативним;
- в) регресійним;
- г) адіозним.

8. У якості моделі задачі динамічного програмування використовується:

- а) віконт;
- б) ромб;
- в) граф;
- г) тромб;
- д) невизначений інтеграл.

**9. Процес розв'язання задачі методом динамічного програмування містить у собі дії в наступній послідовності:**

- а) I. Оптимізація кроків, при цьому визначаються умовно-оптимальні керування для кожного кроку.  
II. Оптимізація процесу в цілому, при цьому з оптимальних кроків складається оптимальний процес шляхом вибору оптимального управління з можливих умовно-оптимальних;
- б) I. Оптимізація кроків, при цьому визначається знак кожного кроку.  
II. Оптимізація процесу в цілому, при цьому визначається знак добутку окремих кроків;
- в) I. Оптимізація процесу в цілому, при цьому з оптимальних кроків складається оптимальний процес шляхом вибору оптимального управління з можливих умовно-оптимальних.  
II. Оптимізація кроків, при цьому визначаються умовно-оптимальні керування для кожного кроку.

**Тема 6. Методи оптимізації**

**1. Оптимізація багатокрокового процесу є:**

- а) керованою операцією;
- б) некерованою операцією;
- в) ідеалізованою операцією;
- г) стохастичною операцією.

**2. Оптимізацією називається:**

- а) процес знаходження найкращого за яким-небудь показником варіанта з множини варіантів;
- б) процес знаходження області застосування математичного методу;

в) процес вибору значення похибки при чисельному розв'язанні задачі.

**3. Математична модель оптимізації в загальному вигляді** ( $Q$  - цільова функція;  $x_1, x_2, \dots, x_n$  - змінні оптимізації) - це:

а)  $Q(x_1, x_2, \dots, x_n) \rightarrow \min (\max)$   
за умовою виконання обмежень;  
 $f_1(x_1, x_2, \dots, x_n) = 0$ ;  
 $f_2(x_1, x_2, \dots, x_n) = 0$ ;  
...  
 $f_m(x_1, x_2, \dots, x_n) = 0$ ;

б)  $Q(x_1, x_2, \dots, x_n) = 0$   
за умовою виконання обмежень;  
 $f_1(x_1, x_2, \dots, x_n) = 0$ ;  
 $f_2(x_1, x_2, \dots, x_n) = 0$ ;  
...  
 $f_m(x_1, x_2, \dots, x_n) = 0$ .

в)  $Q(x_1, x_2, \dots, x_n) \rightarrow \infty$   
за умовою виконання обмежень;  
 $f_1(x_1, x_2, \dots, x_n) = 0$ ;  
 $f_2(x_1, x_2, \dots, x_n) = 0$ ;  
...  
 $f_m(x_1, x_2, \dots, x_n) = 0$ .

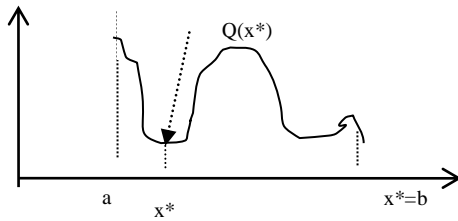
**4. Що є ознакою багатокритеріальної задачі оптимізації?**

а) кількість цільових функцій, що оптимізується, більше двох;

б) кількість цільових функцій, що оптимізується, більше однієї;

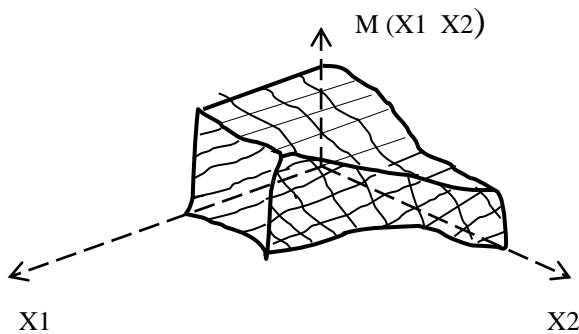
в) кількість цільових функцій, що оптимізується, є випадковою величиною.

**5. На рисунку зображено:**



- а) локальний мінімум;
- б) глобальний максимум;
- в) глобальний мінімум;
- г) мінімакс;
- д) точка розриву 2 роду.

**6. На рисунку зображено:**



- а) двовимірну цільову функцію у вигляді поверхні відгуку;
- б) сегмент екстремальної вибірки;
- в) знаряддя праці стародавньої людини;
- г) тривимірну цільову функцію у вигляді поверхні відгуку.

**7. Роботу алгоритму знаходження мінімуму методом повного перебору можна описати математичною залежністю ( $Y_i$  - чергове значення функції):**

а) 
$$Y_{MIN} = \begin{cases} Y_i, & \text{якщо } Y_i > Y_{MIN}; \\ Y_{MIN}, & \text{якщо } Y_i \leq Y_{MIN}; \end{cases}$$

$$\text{б) } Y_{MIN} = \begin{cases} Y_i, & \text{якщо } Y_i < Y_{MIN}; \\ Y_{MIN}, & \text{якщо } Y_i \geq Y_{MIN}; \end{cases}$$

$$\text{в) } Y_{MIN} = \begin{cases} Y_i, & \text{якщо } Y_i \neq Y_{MIN}; \\ Y_{MIN}, & \text{якщо } Y_i = Y_{MIN}. \end{cases}$$

## 8. Чим викликана похибка методу повного перебору?

а) комп'ютер обчислює дискретні значення функції при відповідних дискретних значеннях параметра, а реальний екстремум може бути між ними;

б) порівняння чергового значення функції з мінімальним проводиться приблизно;

в) великою кількістю обчислювальних операцій при значній кількості порівнянь.

## 9. Метод золотого перетину припускає:

а) ділення інтервалу пошуку екстремуму точками, координати яких обчислюються за певним законом, обчислення значення цільової функції  $Q(x)$  у них, порівняння значень і відкидання тієї частини інтервалу, на якій відсутній екстремум;

б) розбиття на часткові відрізки всього інтервалу, на якому шукається екстремум, і на кожному з часткових відрізків наближену заміну цільової функції  $Q(x)$  многочленом невисокого ступеня;

в) виходячи з довільного початкового вектора з точками, координати яких обчислюються за певним законом, будується ітераційний процес;

г) за координатами точок масиву  $\{x_i\}$  обчислюються коефіцієнти невідомих, що входять до цільової функції  $Q(x)$ .

**10. Назвіть основні етапи процесу знаходження мінімуму функції методу золотого перетину:**

- а) 1. На інтервалі  $[a, b]$  визначаємо значення точок  $x_1, x_2$ .  
2. Обчислюємо значення  $Q(x_1), Q(x_2)$ .  
3. Якщо  $Q(x_1) \leq Q(x_2)$ , то для подальшого розгляду залишаємо  $[a, x_2]$ .  
4. Якщо  $Q(x_1) > Q(x_2)$ , то для подальшого розгляду залишаємо  $[x_1, b]$ .

Процес ділення триває, поки довжина інтервалу невизначеності не стане менше заданої точності  $\epsilon$ ;

- б) 1. На інтервалі  $[a, b]$  визначаємо значення точок  $x_1, x_2$ .  
2. Обчислюємо значення  $Q(x_1), Q(x_2)$ .  
3. Якщо  $Q(x_1) \leq Q(x_2)$ , то для подальшого розгляду залишаємо  $[x_1, b]$ .  
4. Якщо  $Q(x_1) > Q(x_2)$ , то для подальшого розгляду залишаємо  $[a, x_2]$ .

Процес ділення триває, поки довжина інтервалу невизначеності не стане менше заданої точності  $\epsilon$ ;

- в) 1. На інтервалі  $[a, b]$  визначаємо значення точок  $x_1, x_2$ .  
2. Обчислюємо значення  $Q(x_1), Q(x_2)$ .  
3. Якщо  $Q(x_1) \leq Q(x_2)$ , то для подальшого розгляду залишаємо  $[a, x_2]$ .  
4. Якщо  $Q(x_1) > Q(x_2)$ , то для подальшого розгляду залишаємо  $[x_1, b]$ .

Процес ділення триває, поки довжина інтервалу невизначеності не стане рівною 0.

**11. Виберіть визначення унімодальної функції:**

а) функція, задана на  $[a, b]$ , називається унімодальною, якщо існує єдина точка  $x^*$  мінімуму  $Q(x)$

$$Q(x^*) = \text{MIN}(Q(x)), \\ a \leq x \leq b$$

і якщо для будь-яких точок  $x_1, x_2 \in [a, b]$  виконуються співвідношення

$$x^* \leq x_1 \leq x_2 \Rightarrow Q(x^*) \leq Q(x_1) \leq Q(x_2);$$

$$x^* \geq x_1 \geq x_2 \Rightarrow Q(x^*) \leq Q(x_1) \leq Q(x_2);$$

б) функція, задана на  $[a, b]$ , називається унімодальною, якщо існує єдина точка  $x^*$  мінімуму  $Q(x)$ ,

$$Q(x^*) = \text{MIN}(Q(x)),$$

$$a \leq x \leq b$$

і якщо для будь-яких точок  $x_1, x_2 \in [a, b]$  виконуються співвідношення

$$x^* \geq x_1 \geq x_2 \Rightarrow Q(x^*) \leq Q(x_1) \leq Q(x_2);$$

$$x^* \leq x_1 \leq x_2 \Rightarrow Q(x^*) \leq Q(x_1) \leq Q(x_2);$$

в) функція, задана на  $[a, b]$ , називається унімодальною, якщо існує єдина точка  $x^*$  мінімуму  $Q(x)$ ,

$$Q(x^*) = \text{MIN}(Q(x)),$$

$$-\infty < x < +\infty$$

і якщо для будь-яких точок  $x_1, x_2 \in [a, b]$  виконуються співвідношення

$$x^* \leq x_1 \leq x_2 \Rightarrow Q(x^*) \leq Q(x_1) \leq Q(x_2);$$

$$x^* \geq x_1 \geq x_2 \Rightarrow Q(x^*) \leq Q(x_1) \leq Q(x_2).$$

**12. Гарантувати застосовність методу золотого перетину можна тільки для:**

- а) безперервних функцій;
- б) унімодальних функцій;
- в) багатоекстремальних функцій;
- г) кусочно-безперервних функцій.

## Тема 7. Чисельне розв'язання диференціальних рівнянь

### 1. Чим викликана похибка усікання при обчисленні похідної за формулами чисельного диференціювання?

а) похибка усікання викликана заміною даної функції  $f(x)$  інтерполяційним багаточленом  $P_n(x)$ ;

б) похибка усікання викликана неточним завданням вхідних значень даної функції  $f(x)$ ;

в) похибка усікання викликана неточним завданням початкових і граничних даних для вхідної функції  $f(x)$ .

### 2. Чисельне розв'язання методом Ейлера задачі Коші для звичайних диференціальних рівнянь - це:

а) у методі Ейлера розв'язок  $y(x)$  диференціального рівняння  $y' = f(x, y)$  знаходиться як границя послідовності значень функцій  $y_n(x)$ , які обчислюються за рекурентною формулою

$$y_n(x) = y_0 + \int_{x_0}^{x_n} (f(x, y_{n-1}(x))) dx + c ;$$

б) будується система рівновіддалених точок  $x_i = x_0 + i \cdot h$  ( $i = 0, 1, 2, \dots$ ).

Обчислення значень  $y(x_i)$ , які є розв'язком диференціального рівняння  $y' = f(x, y)$ , здійснюються у два етапи. На першому етапі знаходяться проміжні значення

$y_i = y_i + \alpha h f(x_i, y_i)$  з шагом  $\alpha h$ , на другому етапі -  $y_{i+1} = y_i + (1 - \sigma) h f(x_i, y_i) + \sigma h f(x_i + \alpha h, y_i)$ , де  $\alpha > 0$ ,  $\sigma > 0$  - параметри, значення яких обумовлені з міркувань точності;

в) Будується система рівновіддалених точок  $x_i = x_0 + i \cdot h$  ( $i = 0, 1, 2, \dots$ ) при достатньо малому кроці  $h$ . Приблизні значення  $y(x_i)$ , які є розв'язками диференціального рівняння  $y' = f(x, y)$ , обчислюються послідовно за формулою

$$y_{i+1} = y_i + h \cdot f(x_i, y_i).$$



### **3. Назвіть області застосування формул чисельного диференціювання:**

а) чисельне диференціювання найчастіше застосовується, коли потрібно визначити припустиму похибку аргументів за припустимою похибкою функції;

б) до чисельного диференціювання найчастіше прибігають, коли доводиться обчислювати значення функції в проміжних точках, при цьому дана функція задана в табличному вигляді й аналітичне подання функції невідомо;

в) до чисельного диференціювання найчастіше прибігають, коли доводиться обчислювати похідні від функцій, які задані за допомогою таблиці, або коли безпосереднє диференціювання функції є заважким.

### **4. Яка система має назву динамічної?**

а) система, у якій не буває перехідних процесів;

б) система, яка рухається в просторі;

в) система, у якій мають місце перехідні процеси;

г) система, яка діє в умовах динамічних навантажень.

### **5. Яка система має назву детермінованої?**

а) система, на поведінку якої можуть впливати випадкові зовнішні фактори;

б) система, на поведінку якої не впливають випадкові зовнішні фактори;

в) система, поведінка якої залежить від стану зовнішнього середовища;

г) система, поведінка якої не залежить від вхідних параметрів.

## 6. Різницева апроксимація задачі Коші для звичайного диференціального рівняння (ДР) першого порядку - це:

а) вихідним пунктом при різницевій апроксимації є обчислення похибки  $\epsilon$ . Вважається, що похибку  $\epsilon$  можна подати у вигляді інтерполяційного полінома  $n$ -го ступеня. Якщо різницева схема стійка, то зростання будь-якого збурювання обмежене, отже, можна взяти інтеграл Фур'є від даного полінома. У такому випадку похибка апроксимації прагне до нуля при зменшенні різницевої сітки;

б) вихідним пунктом при різницевій апроксимації є заміна області безперервної зміни аргументу деякою кінцевою множиною точок, що належать цій області (різницева сітка). Ця дискретна модель середовища описується сітковими функціями, які визначені у вузлах сітки. ДР замінюються відповідними кінцево-різницеvими співвідношеннями. У підсумку досліджувана задача Коші замінюється або, як кажуть, апроксимується системою різницеvих рівнянь - різницеvою схемою;

в) заданий відрізок  $[a, b]$  замінюється системою часткових відрізків  $[x_i, x_{i+1}]$  рівної довжини, яка має назву різницеvої сітки. Відстань між кінцями відрізків  $x_{i+1} - x_i = h$  є одиничною довжиною сітки. На кожному відрізку  $[x_i, x_{i+1}]$  здійснюється чисельне розв'язання ДР. Загальний розв'язок на  $[a, b]$  обчислюється як сума окремих розв'язків по всіх часткових відрізках.

## 7. Яка математична модель описує перехідні процеси?

- а) звичайні диференціальні рівняння;
- б) алгебраїчні рівняння;
- в) степені поліноми;
- г) тригонометричні функції.

**8. Яким чином можливо визначити в математичному вигляді залежність змінних системи від часу?**

- а) розв'язати систему алгебраїчних рівнянь;
- б) розв'язати систему диференціальних рівнянь;
- в) продиференціювати перехідну функцію;
- г) знайти корені системи лінійних рівнянь.

**9. У формулі розкладання функції в ряд Тейлора**

$y_{i+1} = y_i + h \cdot y'_i + \frac{h^2}{2!} y''_i + \frac{h^3}{3!} y'''_i + \dots$  усі члени ряду, які містять похідні другого порядку й вище, відкидаються:

- а) через їхню малість;
- б) неможливості підбрання похідної 2-го порядку;
- в) неможливості підбрання похідних вищих порядків.

**10. Який вираз описує перехідний процес у системі з однією змінною?**

- а)  $y = ax^2 + bx + c$  ;
- б)  $\frac{dy}{dx} = f(x, y)$  ;
- в)  $y = \int_a^b f(x, y) dx$  ;
- г)  $\frac{d^2y}{dx^2} = F(x, y, z)$  .

**11. Чисельне рішення диференціального рівняння - це:**

- а) аналітична залежність  $y=f(x)$ ;
- б) значення функції в довільних точках;
- в) аналітична залежність  $y = f'(x)$ ;
- г) значення функції у вузлових точках.

**12. Щоб розв'язати звичайне диференціальне рівняння, необхідно знати:**

- а) значення залежної змінної та (або) похідних при деяких значеннях незалежної змінної;
- б) значення незалежної змінної та (або) похідних при деяких значеннях залежної змінної;
- в) значення залежної змінної та (або) похідних при деяких значеннях залежної змінної.

**13. Для задачі Коші додаткові умови задаються:**

- а) при декількох значеннях незалежних змінних;
- б) у точках, де 1-ша похідна дорівнює 0;
- в) при одному значенні незалежної змінної.

**14. Для моделювання динамічних процесів найчастіше використовуються:**

- а) алгебраїчні рівняння та системи алгебраїчних рівнянь;
- б) трансцендентні рівняння та системи трансцендентних рівнянь;
- в) диференціальні рівняння та системи диференціальних рівнянь;
- г) тригонометричні рівняння та системи тригонометричних рівнянь.

**15. Для крайової задачі додаткові умови задаються:**

- а) при двох або більше значеннях незалежної змінної;
- б) при одному значенні незалежної змінної;
- в) при двох або більше значеннях залежної змінної.

## ВІДПОВІДІ НА ТЕСТОВІ ПИТАННЯ

### Тема 1. Загальні питання математичного моделювання

Питання №	1	2	3	4	5	6	7	8	9	10
Відповідь	а	Б	а	б	а	в	а	в	а	в

### Тема 2. Комп'ютерне моделювання й обчислювальний експеримент.

Питання №	1	2	3	4	5	6	7	8	9	10	11	12
Відповідь	б	б	а	а	Б	а	в	г	а	а	а	а

### Тема 3. Розв'язок нелінійних рівнянь

Питання №	1	2	3	4	5	6	7	8	9	10
Відповідь	в	А	а	А	а	б	б	а	б	а

### Тема 4. Методи чисельного інтегрування

Питання №	1	2	3	4	5	6	7	8	9	10	11	12
Відповідь	а	в	в	б	Б	в	а	в	б	а	а	в

### Тема 5. Метод динамічного програмування .

Питання №	1	2	3	4	5	6	7	8	9
Відповідь	а	А	б	А	а	б	а	в	в

### Тема 6. Методи оптимізації.

Питання №	1	2	3	4	5	6	7	8	9	10	11	12
Відповідь	а	а	а	а	б	а	б	а	а	а	а	б

### Тема 7. Чисельне рішення диференційних рівняння

Питання №	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Відповідь	а	в	Б	г	б	б	а	б	а	б	г	а	в	в	а

## ПИТАННЯ ДЛЯ КОНТРОЛЮ

### 1. Особливості побудови математичних моделей

1. Що необхідно для побудови математичної моделі?
2. Від яких факторів залежить математичний опис моделі?
3. Чим визначається точність результатів, які отримані при аналізі моделі?
4. Які припущення використовуються при створенні математичної моделі? Наведіть приклад.
5. Чи визначається математична модель однозначно досліджуванним об'єктом? Відповідь поясніть.
6. У якому випадку математичну модель побудувати найпростіше?
7. Яка математична модель називається гіпотетичною?
8. Назвіть основний критерій істинності гіпотетичної математичної моделі.

### 2. Комп'ютерне моделювання й обчислювальний експеримент

1. Визначення моделі.
2. Використання принципів теорії подібності в моделюванні.
3. Основні поняття моделювання.
4. Типи моделей.
5. Дійсні моделі.
6. Ідеальні моделі.
7. На які типи поділяються аналітичні моделі?
8. Безперервні та дискретні моделі.
9. Статичні та динамічні моделі.
10. Що необхідно зробити для побудови математичної моделі?
11. У чому полягає суть комп'ютерного моделювання?
12. Як перевірити адекватність математичної моделі реальному об'єкту?
13. Охарактеризуйте точні методи розв'язання інженерних задач.

14. Охарактеризуйте чисельні методи розв'язання інженерних задач.

15. Чим пояснюється зростання інтересу до чисельних методів розв'язання інженерних задач з появою ЕОМ?

### 3. Чисельні методи розв'язання нелінійних рівнянь

1. У яких задачах виникає необхідність відшукування коренів нелінійних рівнянь?

2. Наведіть приклади алгебраїчної й трансцендентної функції. Поясніть, чим вони відрізняються.

3. Поясніть поняття «корінь рівняння». Наведіть приклади.

4. Що означає ізолювати корінь?

5. Якою повинна бути функція, що має корені на відрізку  $[a,b]$ ?

6. Перелічіть чисельні методи розв'язання елементарних рівнянь.

7. Сформулюйте ідею методу половинного ділення.

8. Як обчислити координати середини відрізка  $[a,b]$ , на якому шукається корінь у методі половинного ділення?

9. Наведіть схему алгоритму уточнення коренів за методом половинного ділення.

10. У чому полягає ідея методу простої ітерації?

11. Який вигляд має многочлен ступеня  $m$ ?

12. Чому метод Ньютона (*метод дотичних*) належить до градієнтних методів?

13. Яка умова повинна виконуватися, щоб забезпечити швидку збіжність методу Ньютона (*методу дотичних*)?

14. За якою рекурентною формулою реалізується ітераційний процес наближення до кореня в методі Ньютона (*методі дотичних*)?

15. Наведіть формулу, за якою здійснюється чисельне диференціювання для обчислення похідних на кожному кроці в модифікованому методі Ньютона (*методі січних*).

#### **4. Комп'ютерне імітаційне моделювання. Статистичне імітаційне моделювання**

1. Що являє собою *імітаційне моделювання*?
2. Назвіть основні переваги *імітаційного моделювання*.
3. У яких випадках рекомендується використовувати *імітаційне моделювання*?
4. Які недоліки має *імітаційне моделювання*?
5. Охарактеризуйте імовірнісні аналітичні моделі.
6. Що таке статистична модель випадкового процесу?
7. Дайте визначення поняттю «*статистичне імітаційне моделювання*».
8. У чому сутність методу Монте-Карло?
9. З яких етапів складається методика статистичного моделювання?
10. *Метод середини квадрата*. У чому його сутність?
11. Запишіть "магічні числа", що використовуються в *лінійному конгруентному методі*.
12. Запишіть формулу одержання випадкових чисел, запропоновану Гріном.
13. Запишіть формулу одержання випадкових чисел квадратичним методом, запропоновану Р. Ковеєм.
14. Якщо  $x_1^2 = 0.06135529$ , то чому дорівнює  $x_2$ ?
15. Що необхідно зробити для знаходження об'єктивних і стійких характеристик випадкового процесу?

#### **5. Комп'ютерне моделювання і розв'язання лінійних багатовимірних систем**

1. Який вигляд має система лінійних рівнянь (СЛР) у матричній формі?
2. Які методи розв'язання СЛР за типом отриманих результатів вам відомі?
3. Назвіть точні методи рішення СЛР.
4. Сформулюйте обов'язкову умову існування єдиного розв'язку СЛР.
5. Яка ідея лежить в основі метода Гаусса?
6. Які етапи містить у собі метод Гаусса?



7. Який елемент має назву «головний» в методі Гаусса?
8. Що таке «прямий хід» методу Гауса?
9. Надайте блок-схему «прямого ходу» методу Гаусса.
10. Що таке «зворотній хід» методу Гауса?
11. Надайте укрупнену блок-схему методу Гаусса.
12. Надайте блок-схему пошуку ненулевого головного елемента.
13. Яким чином проводиться перевірка результатів обчислення методом Гаусса.
14. Надайте блок-схему перевірки результатів визначення розв'язку СЛР методом Гаусса.
15. Які недоліки методу Гаусса ви можете визначити?
16. Наведіть приклад розв'язання трьох лінійних рівнянь (у загальному вигляді).

## **6. Моделювання багатовимірних нелінійних систем**

1. Пояснить, чому математична модель може бути описана системою нелінійних рівнянь?
2. Що є розв'язком системи нелінійних рівнянь?
3. Які методи розв'язання систем нелінійних рівнянь вам відомі?
4. Від чого залежить ефективність ітераційних методів розв'язання систем нелінійних рівнянь?
5. Що таке область збіжності?
6. У чому сутність методу простих ітерацій?
7. У яких випадках використовується метод простих ітерацій?
8. Наведіть схему алгоритму методу простих ітерацій.
9. У чому сутність методу Ньютона розв'язання системи нелінійних рівнянь?
10. Яка ідея лежить в основі метода Ньютона?
11. З яких етапів складається метод Ньютона?
12. Які переваги має метод Ньютона порівняно з іншими методами розв'язання систем нелінійних рівнянь?
13. Яка матриця називається Якобіаном?
14. Які методи знаходження матриці Якобі вам відомі?
15. Наведіть укрупнену блок-схему алгоритму методу Ньютона.

## **7. Комп'ютерне моделювання при обробці дослідних даних**

1. Який тип функції отримує науковець під час експерименту?
2. Які основні задачі обробки дослідницьких даних вам відомі?
3. У чому полягає задача інтерполяції?
4. Що таке вузли інтерполяції?
5. У якому вигляді шукається функція, яка наближає задану табличну функцію?
6. Яким умовам повинна відповідати допоміжна функція?
7. Яка мета знаходження інтерполяційного многочлена?
8. Як побудувати інтерполяційний многочлен в явному вигляді?
9. Сутність інтерполяції за Лагранжем.
10. Наведіть схему алгоритму інтерполяції за Лагранжем?
11. У чому сутність інтерполяції за Ньютоном?
12. Що таке розділені різниці?
13. Наведіть схему алгоритму інтерполяції за Ньютоном.
14. У чому сутність сплайн-інтерполяції?
15. Що таке апроксимація даних?
16. Які два основних способи апроксимації вам відомі, у чому їх суттєва різниця?
17. Сутність методу найменших квадратів.
18. Наведіть укрупнену схему методу найменших квадратів.

## **8. Чисельні методи інтегрування**

1. Наведіть приклади задач, для розв'язання яких необхідно знаходження визначеного інтеграла.
2. У чому сутність чисельних методів інтегрування?
3. Наведіть порядок обчислення інтеграла чисельними методами.
4. Які чисельні методи інтегрування вам відомі?
5. Поясніть сутність методу прямокутників.
6. У чому різниця між методом лівих прямокутників і правих прямокутників?

7. Яку точність мають методи чисельного інтегрування?
8. Наведіть блок-схему методу прямокутників.
9. У чому сутність методу трапецій?
10. Наведіть блок-схему методу трапецій.
11. У чому сутність методу Сімпсона?
12. Наведіть блок-схему методу Сімпсона.
13. У чому геометричний зміст методів чисельного інтегрування.
14. Які особливості має вибір кількості частин розподілу (розбиття) в методі Сімпсона?

### **9. Чисельні методи вирішення диференціальних рівнянь першого порядку**

1. Поясніть поняття «динамічна система».
2. Поясніть поняття «динамічна детермінована система».
3. Коли слід використовувати аналітичні, а коли чисельні методи розв'язання задачі Коші?
4. Наведіть приклади динамічних систем.
5. Які ви знаєте типи задач, розв'язок яких моделює поведінка динамічних систем?
6. Що являє собою результат чисельного розв'язання диференціального рівняння?
7. Наведіть загальний вигляд диференціального рівняння.
8. Як записати нормальну форму диференціального рівняння в нормальній формі?
9. Що є загальним розв'язанням звичайного диференціального рівняння?
10. Як здійснюється виділення частинного розв'язку диференціального рівняння з сімейства загальних?
11. Запишіть формулу Ейлера для розв'язання диференціального рівняння.
12. У чому джерело похибки при інтегруванні ОДУ методом Ейлера?
13. Яка похибка методів Ейлера?
14. Наведіть формулу обчислення числа кроків інтегрування рівняння на  $[a,b]$  з кроком  $h$ .
15. Запишіть розкладання функції  $y(x)$  в ряд Тейлора.

## МОДЕЛЮВАННЯ РОБОТИ ПУНКТУ ПЕРЕСТАНОВКИ ВАГОНІВ

*Задача.* У пункті перестановки вагонів для заміни візків состави подаються кожні 50-70 хв. Заміна візків проводиться одночасно для всього состава. Час, витрачений бригадою на заміну візків для одного состава складає 40-60 хв. У пункті перестановки вагонів працює одна бригада, состави подаються на одну платформу. Розробити модель заміни візків составів і проаналізувати роботу пункту перестановки вагонів виходячи з того, що тривалість зміни складає 10 год.

Задача складається з синхронізації двох процесів - перший процес подача составів до пункту перестановки вагонів, другий процес - безпосередньо заміна візків.

Розглянемо фактори, що забезпечують роботу пункту перестановки вагонів.

*Перший* фактор ( $x_i$ ) - час надходження составу до пункту перестановки вагонів. Состави надходять за рівномірним законом розподілу в інтервалі 50-70 хв.

У задачі для формування часу надходження кожного состава на сортувальну гірку ( $x_i$ ) та часу розформування кожного состава ( $y_i$ ) потрібно сформуувати послідовності випадкових чисел за рівномірним законом розподілу. На практиці для отримання дійсного значення випадкової величини  $x$ , рівномірно розподіленої на інтервалі від  $a$  до  $b$ , використовують формулу.

$$x = (b - a) * \text{RND} + a,$$

де  $x$  - випадкова величина, одержана за рівномірним законом розподілу;

RND - дійсне число від 0 до 1;

$a$  та  $b$  - відповідно початок і кінець інтервалу розподілу.

У нашому випадку

$$a=50 \text{ хв } b=70 \text{ хв},$$

тому

$$x = (70 - 50) * \text{RND} + 50.$$

*Другий* фактор ( $y_i$ ) - час, що необхідно затратити на заміну візків. Процес заміни візка також підпорядкований рівномірному закону розподілу в інтервалі 40-60 хв.

Тому

$$x = (60 - 40) * \text{RND} + 40.$$

*Третій* фактор ( $K$ ) - тривалість зміни складає

$$K = 10 \text{ год} = 10 * 60 \text{ хв.}$$

*Четвертий* фактор ( $n$ ) - кількість складів, що можливо обробити за планом, на цій ділянці розраховується як тривалість зміни, поділена на мінімальний час одного з процесів ( $N = 10 * 60 / 40 = 75$ ).

Для ефективного управління процесом розформування складів треба визначити:

- час надходження кожного складу на сортувальну гірку ( $x_i$ );
- час розформування кожного складу ( $y_i$ );
- час початку і завершення розформування кожного складу ( $t_{ni}, t_{ki}$ );
- час очікування розформування кожного складу ( $w_i$ );
- максимальний час розформування складу ( $u_{\max}$ );
- мінімальний час розформування складу ( $u_{\min}$ );
- номер складу, що розформувався довше всіх ( $i_{\max}$ );
- номер складу, що розформувався менше всіх ( $i_{\min}$ );
- номер складу, що очікував розформування довше всіх ( $i_{\max}$ );
- номер складу, що очікував розформування менше всіх ( $i_{\min}$ );
- середній час розформування складів ( $u_{\text{ср}}$ );
- середній час очікування розформування складів ( $w_{\text{ср}}$ );
- сумарний час очікування всіма складами  $w$ ;
- кількість складів, що надійшли на сортувальну гірку за зміну ( $k_{ol}$ ) (табл. Д.3.1).



- максимальний час розформування состава ( $y_{\max} \leftrightarrow \{y_i\}_{\max}$ );
- мінімальний час розформування составу ( $y_{\min} \leftrightarrow \{y_i\}_{\min}$ ).
- номер составу, що розформовувався довше всіх ( $i_{\max} \leftrightarrow \{y_i\}_{\max}$ );
- номер составу, що розформовувався менше всіх ( $i_{\min} \leftrightarrow \{y_i\}_{\min}$ );
- номер составу, що очікував розформування довше всіх ( $i_{\max} \leftrightarrow \{w_i\}_{\max}$ );
- номер составу, що очікував розформування менше всіх ( $i_{\min} \leftrightarrow \{w_i\}_{\min}$ );

- середній час розформування составів ( $y_{\text{ср}} = \frac{1}{kol} \sum_{i=0}^{kol-1} y_i$ );

- середній час очікування розформування составів ( $w_{\text{ср}} = \frac{1}{kol} \sum_{i=0}^{kol-1} w_i$ );

- сумарний час очікування всіма складами  $w = \sum_{i=0}^{kol-1} w_i$ ;

- кількість составів, що надійшли на сортувальну гірку за зміну ( $kol$ ).

Складемо перелік ідентифікаторів (табл. Д.3.2), які необхідні для побудови моделі.

Таблиця Д.3.2

## Ідентифікатори

Іденти-фікатор	Опис ідентифікатора	Тип даних	Формат виведення
a1, b1	границі інтервалу очікування інтервалу часу прибуття составів	INTEGER	##, ##
A2, b2	границі інтервалу очікування інтервалу часу обробки составів	INTEGER	##, ##
N	передбачувана кількість модельованих составів	INTEGER	##
K	кількість хвилин у робочій зміні (час моделювання)	INTEGER	###
I	номер оброблюваного состава	INTEGER	##
Xi	часи подачі состава на обробку	SINGLE	###.#
Yi	час обробки состава	SINGLE	###.#
Ti	час прибуття состава на обробку	SINGLE	###.#
Tni	час початку обробки поданого состава	SINGLE	###.#
tki	час закінчення обробки поданого состава	SINGLE	###.#
wi	час простою состава	SINGLE	###.#
vi	час простою обслуговуючої бригади	SINGLE	###.#
kol	кількість составів у даному змодельованому процесі	INTEGER	##
ymax	Найбільший час обробки состава	SINGLE	###.#
ymin	Найменший час обробки состава	SINGLE	###.#
iymax	номер состава, із найбільшим часом обробки	INTEGER	##
iymin	номер состава з найменшим часом обробки	INTEGER	##
wmax	Найбільший час простою состава	SINGLE	###.#
iymax	номер состава з найбільшим часом простою состава	INTEGER	###.#
wmin	Найменший час простою состава	SINGLE	###.#
iymin	номер состава з найменшим часом простою состава	INTEGER	##
sy	сумарний час обробки составів	SINGLE	###.#
ysr	середній час обробки составів	SINGLE	###.#
sw	сумарний час простою составів	SINGLE	###.#
kolw	кількість простоїв составів	INTEGER	##
wsr	середнє значення часу простоїв составів	SINGLE	###.#



1. Рисунок транспортного процесу і вихідні дані (параметри) індивідуального завдання

Побудуємо та опишемо часові діаграми транспортного процесу.

На часових осях відображена послідовність подій, які моделюються (рис. Д.3.1).

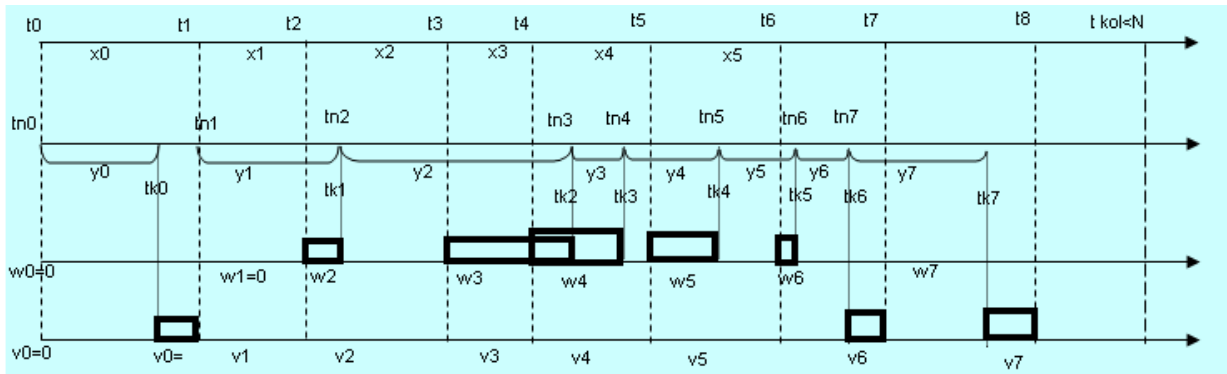


Рис. Д.3.1. Часові співвідношення часу прибуття составів ( $t_i$ ), обробки составів ( $y_i$ ), простою составів ( $w_i$ ), простою бригади ( $v_i$ )

Перша вісь визначає моменти прибуття состава на обробку,  $t_0, t_1, t_2, t_3, \dots$  - це моменти часу, коли перший, другий, третій, ... состави прибувають на сортувальну гірку. Між моментами  $t_i$  і  $t_{i+1}$  визначено інтервал часу  $x$ . Це випадкове число, розподілене за рівномірним законом розподілу на інтервалі 12-16 хв. Таким чином, для отримання моментів часу прибуття всіх составів на обробку  $t_0, t_1, t_2, t_3, \dots$  слід згенерувати ряд випадкових чисел  $x$  із заданими  $a = 12$  і  $b = 16$  і записати їх у масив з урахуванням співвідношення:  $t_i = t_i + x$ .

Друга вісь показує час обробки кожного состава. Величини  $y_0, y_1, y_2, \dots$  складають випадкові, рівномірно розподілені величини на інтервалі 12-16 хв. За логікою подій процесу, що моделюється, інтервал часу  $y_i$  ( $i=0 \div 75$ ) повинен починатися строго або в момент  $t_i$  ( $i=0 \div 75$ ), якщо бригада вільна і може зайнятися обслуговуванням  $i$ -го состава, або в момент часу закінчення обслуговування попереднього состава  $t_{i-1} + y_{i-1}$ , якщо бригада у момент часу  $t_i$  була зайнята.

Третя вісь відображує час простою составів перед обробкою. Цей час визначається як різниця між часом початку обробки поточного состава і його прибуттям на станцію:  $w_i = (t_{i-1} + y_{i-1} + w_{i-1}) - t_i$ , якщо  $t_{i-1} + y_{i-1} + w_{i-1} > t_i$ ;  $w_i = 0$  в інших випадках.

Час початку обробки поточного состава визначається моментом його прибуття (якщо бригада була вільною), у цьому випадку  $w_i = 0$  або дорівнює різниці між моментом завершення обробки попереднього состава (якщо бригада була зайнята попереднім составом) і моментом прибуття на обробку поточного. Момент завершення обробки попереднього состава визначається моментом його прибуття  $t_{i-1}$  плюс час простою перед обробкою (у разі зайнятості бригади  $w_{i-1}$ ) і плюс час обробки цього состава  $y_{i-1}$ .

Четверта вісь показує періоди часу  $v_i$  ( $i=0 \div 75$ ) простою обслуговуючої бригади. Цей час визначається як інтервали часу між завершенням обробки попереднього состава і прибуттям наступного состава:  $v_i = t_i - (t_{i-1} + y_{i-1} + w_{i-1})$ , у випадку, якщо  $t_i > t_{i-1} + y_{i-1} + w_{i-1}$ , інакше  $v_i = 0$ .

Представимо дані для побудови імітаційної моделі транспортного процесу у вигляді системи рівнянь і запишемо її в таблицю.

## Результати досліджень

Графік подачі составів матиме вигляд, як на рис Д.3.2.

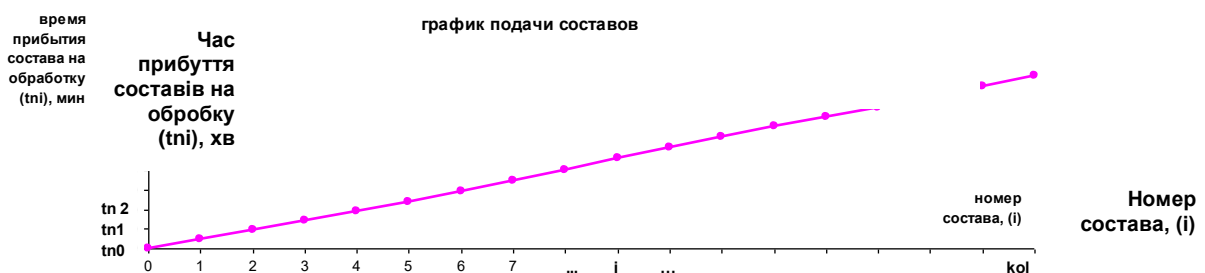


Рис. Д.3.2. Графік подачі составів

Алгоритм розв'язання задачі подано на рис. Д.3.3.

Результат моделювання наведено у таблиці Д.3.3.

Таблиця Д.3.3

Результат моделювання

№ состава (i)	Час подачі состава (xi)	Час обробки состава (yi)	При- буття состава (ti)	Початок обробки (tni)	Закін- чення обробки (tki)	Очіку- вання обробки составами (wi-1)	Простої бригади по составам (vi-1)
0	24	22	0	0	22	0	0
1	16	29	24	24	53	2	0
2	17	30	40	53	83	0	13
3	16	30	57	83	113	0	26
4	17	23	73	113	136	0	40
5	18	23	90	136	205	0	46
6	23	29	108	159	188	0	51
7	22	23	131	188	211	0	57
8	21	24	153	211	235	0	58
9	20	25	174	235	260	0	61
10	19	26	194	260	286	0	66
11	20	27	213	286	313	0	73
12	20	28	233	313	341	0	80
13	20	26	253	341	367	0	88
14	20	26	273	367	393	0	94
15	21	24	293	393	417	0	100
16	19	25	314	417	442	0	103
17	21	25	333	442	467	0	109
18	19	24	354	467	491	0	113

- максимальний час розформування состава ( $u_{\max} = 30$  хв);
- мінімальний час розформування состава ( $u_{\min} = 22$  хв);
- номери составів, що розформувалися довше всіх ( $i_{\max} = 2, 3$ );
- номери составів, що розформувалися менше всіх ( $i_{\min} = 1$ );
- номери составів, що очікували розформування довше всіх ( $i_{\max} = 0$ );
- номери составів, що очікували розформування менше всіх ( $i_{\min} = 1-18$ );
- середній час розформування составів ( $u_{\text{ср}} \approx 26$  хв);
- середній час очікування розформування составів ( $w_{\text{ср}} = 2$  хв);
- сумарний час очікування всіма составами  $w = 2$  хв;
- кількість составів, що надійшли на сортувальну гірку за зміну ( $k_{\text{ол}} = 19$ ).

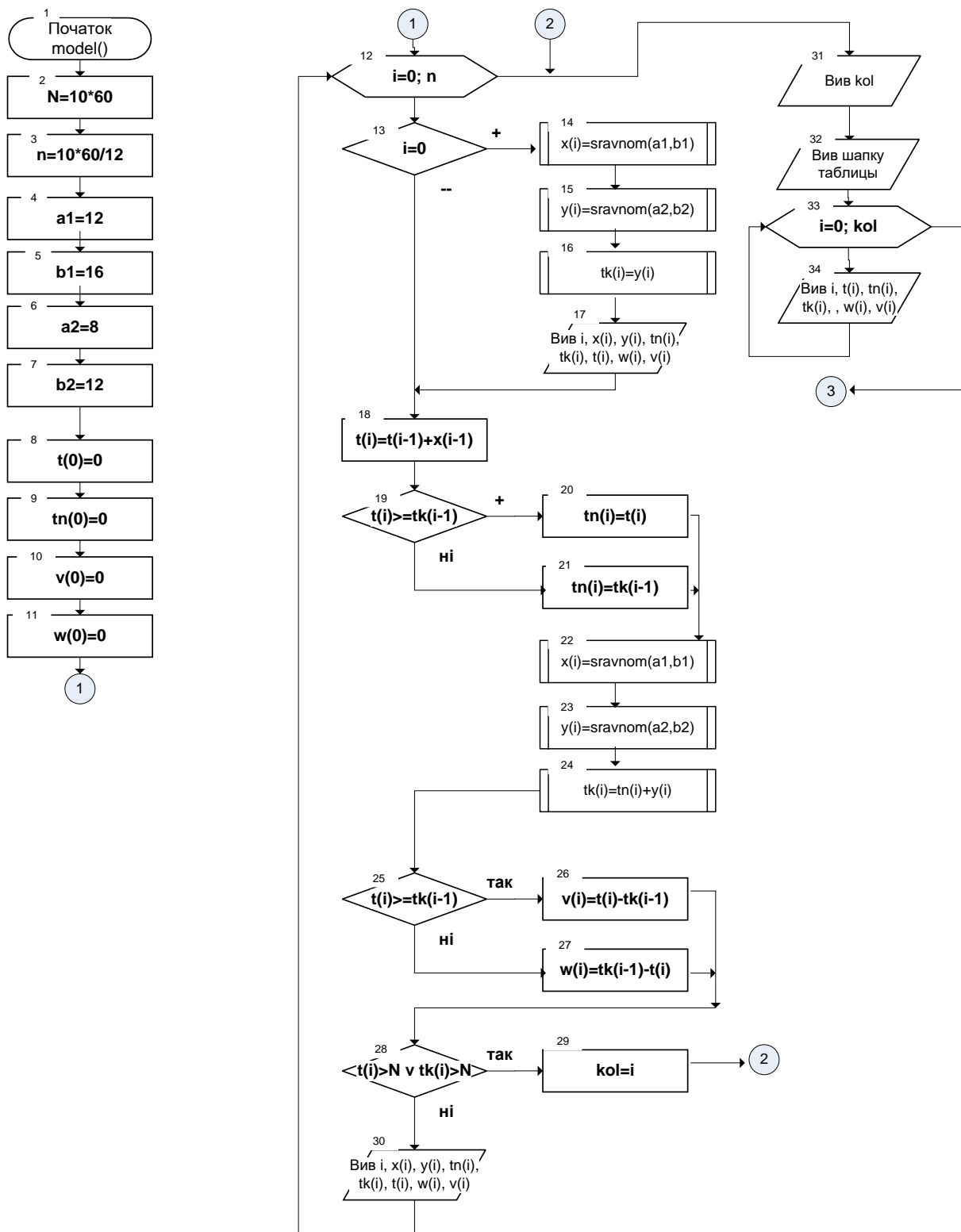


Рис. Д.3.3. Схема алгоритму (функція model()) (початок)

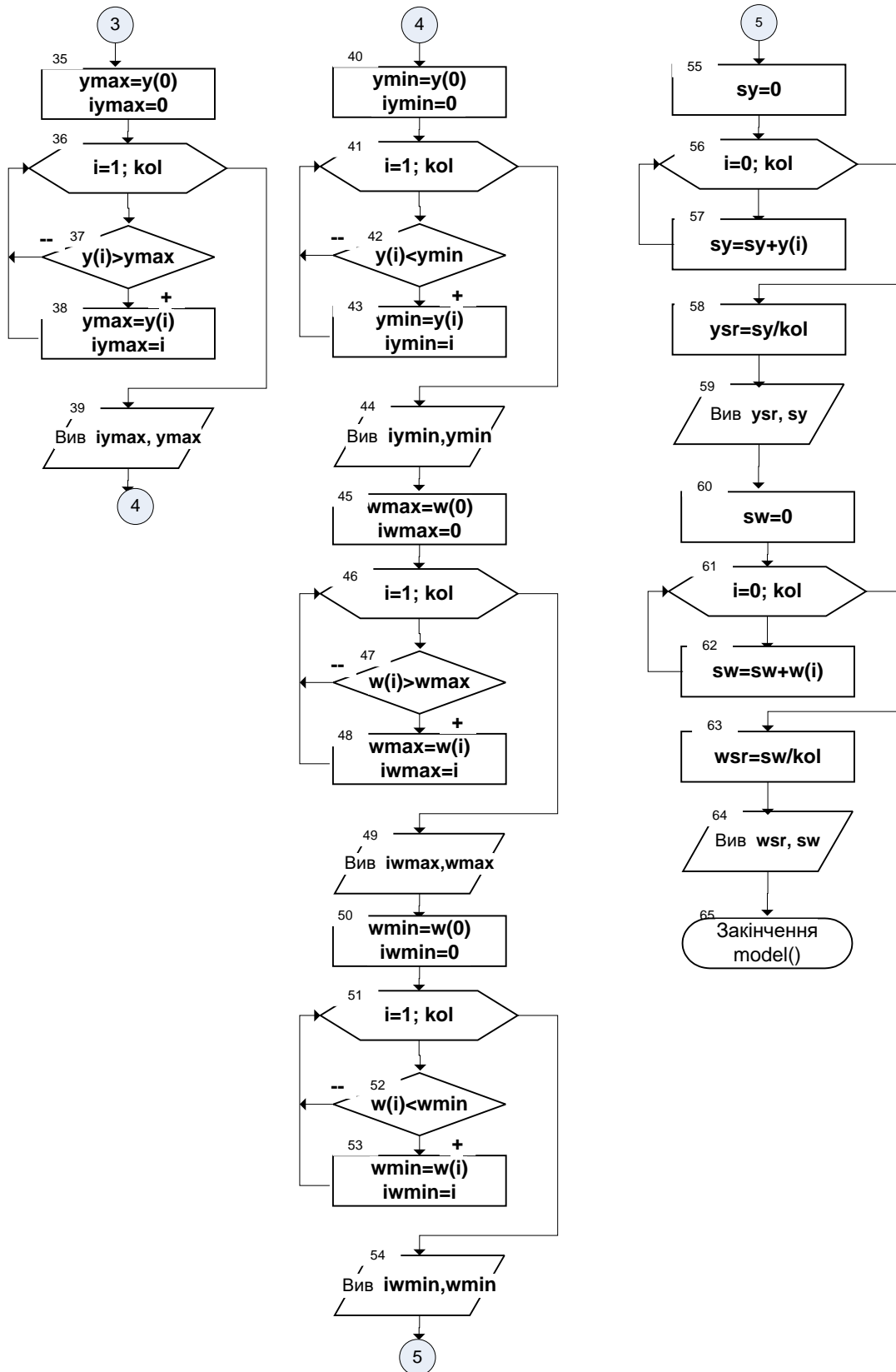


Рис. Д.3.3. Схема алгоритму (функція model()) (закінчення)

## Опис алгоритму

В алгоритмі цикл, що складається з блоків 4, 5, 12, 13, 14, 22, 28 формує випадкові величини - моменти прибуття составів на обробку. Цикл, який складається з блоків 6, 7, 12, 13, 15, 23, 28 визначає випадкові величини - часи обробки кожного состава. Цикл, що складається з блоків 12-31, перевіряє умови надходження і обробки составів, визначає час незайнятості бригади ( $v_i$ ) або простою составів ( $w_i$ ), часи початку обробки поточного состава і сумарний час простою. У циклі (блоки 32-34) виводяться на друк такі дані: час початку і закінчення обробки кожного состава (окрім нульового) і дані оформляються у вигляді таблиці.

У блоках 2, 3 визначається тривалість зміни у хвилинах і передбачувана кількість составів. У блоці 8 заданий  $t_0 = 0$  - вважаємо, що перший состав поступив на обробку в початковий, нульовий момент часу. У блоках 10, 11  $v_0=0$  і  $W_0=0$ , тобто в початковий момент бригада без затримки приступає до обробки першого состава  $i$ , відповідно, час простою першого состава дорівнює 0. У блоках 9, 16 - обчислюється час початку подачі составів на обробку і встановлюється час закінчення обробки нульового состава. У блоці 17 виведення на друк даних по нульовому составу.

У блоці 35 припускається, що найбільший час обробки состава дорівнює  $u_0$  і його номер 0. У блоках 36, 37, 38 перебираються (блок 36) состави, що залишилися, і якщо знаходиться (блок 37) такий состав, час обробки якого ( $u_i$ ) більший, ніж передбачуваний час обробки, то фіксуємо (блок 36) знайдений найбільший час обробки і номер состава. Блок 39 - виведення на друк найбільшого часу обробки і номера состава.

У блоці 40 припускається, що найменший час обробки состава дорівнює  $u_0$  і його номер 0. У блоках 41, 42, 43 перебираються (блок 36) состави, що залишилися, і, якщо знаходиться (блок 42) такий состав, час обробки якого ( $u_i$ ) менше, ніж передбачуваний час обробки, то фіксуємо (блок 43) знайдений найбільший час обробки і номер состава. Блок 44 - виведення на друк найменшого часу обробки і номера состава.

У блоці 45 припускається, що найменший час простою состава дорівнює  $w_0$  і його номер 0. У блоках 46, 47, 48 перебираються (блок 46) состави, що залишилися, і якщо знаходиться (блок 47) такий состав, час простою якого ( $w_i$ ) більше, ніж передбачуваний, то фіксуємо (блок 48) знайдений найбільший час простою і номер цього состава. Блок 49 - виведення на друк найбільшого часу простою состава і номера цього состава.

У блоці 50 припускається, що найбільший час простою состава дорівнює  $w_0$  і його номер 0. У блоках 51, 52, 53 - перебираються (блок 51) состави, що залишилися, і якщо знаходиться (блок 52) такий состав, час простою якого ( $w_i$ ) менше, ніж передбачуваний, то фіксуємо (блок 53) знайдений найменший час простою і номер цього состава. Блок 54 - виведення на друк найменшого часу простою состава і його номер.

У блоках 55, 56, 57 накопичується сума часу обробки составів. У блоці 58 вираховується середній час обробки потоку составів. Блок 59 - виведення на друк загального та середнього часів обробки составів.

У блоках 60, 61, 62 накопичується сума часу простоїв составів. У блоці 63 вираховується середній час простоїв потоку составів. Блок 64 - виведення на друк загального та середнього часів простоїв составів (рис. Д.3.4).

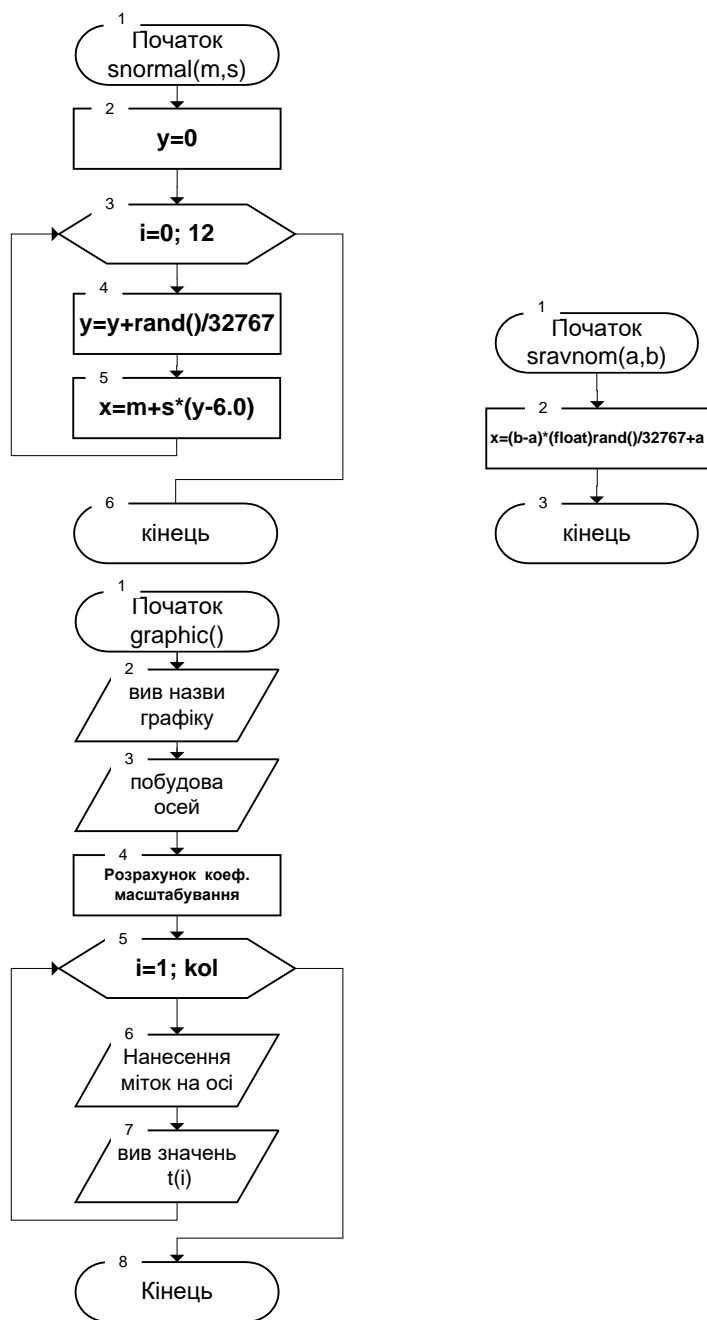


Рис. Д.3.4. Схеми алгоритмів функцій

snormal();      sravnom();      graphic()

Функція snormal() - у блоках 2-5 формуємо випадкову змінну  $x$  для визначення часу обробки составів ( $y_i$ ).

Функція sravnom() - у блоці 2 формуємо випадкову змінну  $x$  для визначення часів подачі составів ( $x_i$ ).



Текст BASIC-програми:

```
DECLARE SUB model ()
DECLARE SUB graphic (kol, tk())
DECLARE FUNCTION snormal! (m AS INTEGER, s AS INTEGER)
DECLARE FUNCTION sravnom! (a AS INTEGER, b AS INTEGER)
DO
    CLS
    LOCATE 2, 9: PRINT "1. Побудова моделі"
    LOCATE 3, 9: PRINT "2. Вихід"
    LOCATE 4, 9: INPUT "Ваш вибір? "; v

    SELECT CASE v
        CASE 1
            CALL model
        CASE 2
            GOTO m1
    END SELECT

LOOP WHILE (1)

m1: CLS : PRINT "Програма завершила роботу"

SUB graphic (kol, tk())
    SCREEN 12
    DIM i AS INTEGER
    PRINT "          ГРАФІК ПРИБУТТЯ ВАГОНІВ"

    ' Побудова координатних осей
    LINE (10, 450)-(630, 450)
    LINE (630, 450)-(620, 440)
    LINE (630, 450)-(620, 460)

    LINE (34, 30)-(34, 470)
    LINE (34, 30)-(24, 40)
    LINE (34, 30)-(44, 40)

    ' Розрахунок коефіцієнтів масштабування
    kx = 600 / kol
    ky = 450 / tk(kol)

    ' Виведення графіка та нанесення міток на вісь
    FOR i = 1 TO kol
        xd = kx * i
        yd = 480 - ky * tk(i) + 20
        PSET (FIX(xd), FIX(yd))
        LINE (FIX(xd), 620)-(FIX(xd), 620 + 10)
    NEXT i
```

```
SLEEP
SCREEN 0
END SUB
```

```
SUB model
```

```
  N = 50
```

```
  Nsn = 2 * 60
```

```
  DIM x(N) AS SINGLE
```

```
  DIM y(N) AS SINGLE
```

```
  DIM t(N) AS SINGLE
```

```
  DIM tn(N) AS SINGLE
```

```
  DIM tk(N) AS SINGLE
```

```
  DIM w(N) AS SINGLE
```

```
  DIM v(N) AS SINGLE
```

```
  DIM i AS INTEGER, vib AS INTEGER
```

```
  DIM s1 AS INTEGER, m1 AS INTEGER, a AS INTEGER, b AS INTEGER
```

```
  DIM maxy AS SINGLE, miny AS SINGLE
```

```
    jy = 0: jw = 0
```

```
    maxny = 0: minny = 0: maxnw = 0: minnw = 0
```

```
    sy = 0: sw = 0
```

```
  ' RANDOMIZE TIMER
```

```
    m1 = 16: s1 = 4: a = 12: b = 16
```

```
    t(1) = 0: tn(1) = 0: w(1) = 0: v(1) = 0
```

```
  CLS
```

```
  PRINT "У результаті роботи були отримані такі дані"
```

```
  PRINT "i   x   y   tn   tk   t   w   v "
```

```
  FOR i = 1 TO N
```

```
  IF i = 1 THEN
```

```
    x(i) = snormal(m1, s1)
```

```
    y(i) = sravnom(a, b)
```

```
    tk(i) = y(i)
```

```
  PRINT USING "|###| #####.# |#####.# | #####.# | #####.# | #####.# | #####.#|"; i; x(i); y(i);  
tn(i); tk(i); t(i); w(i); v(i)
```

```
  ELSE
```

```
    t(i) = t(i - 1) + x(i - 1)
```

```
    IF t(i) >= tk(i - 1) THEN tn(i) = t(i) ELSE tn(i) = tk(i - 1)
```

```
    x(i) = snormal(m1, s1)
```

```
    y(i) = sravnom(a, b)
```

```
    tk(i) = tn(i) + y(i)
```

```
  IF t(i) >= tk(i - 1) THEN v(i) = t(i) - tk(i - 1) ELSE w(i) = tk(i - 1) - t(i)
```

```
  IF t(i) >= Nsn OR tk(i) >= Nsn THEN kol = i: PRINT USING "|###| #####.# |#####.# |  
#####.# | #####.# | #####.# | #####.#|"; i; x(i); y(i); tn(i); tk(i); t(i); w(i); v(i):  
EXIT FOR
```

```
  PRINT USING "|###| #####.# |#####.# | #####.# | #####.# | #####.# | #####.#|"; i; x(i); y(i);  
tn(i); tk(i); t(i); w(i); v(i)
```

END IF

```
IF i = 20 OR i = 40 OR i = 60 THEN INPUT ""; qqq
NEXT i
PRINT USING "kol = ## "; kol
INPUT ""; qqq
```

CLS

PRINT "РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ПРИКЛАДІ  
ПОБУДОВИ"

PRINT " ІМІТАЦІЙНОЇ МОДЕЛІ ТРАНСПОРТНОГО ПРОЦЕСУ "

PRINT " Розробник: Іванов І.І. група 3-Ш-Л"

PRINT "|-----|"

PRINT "| РЕЗУЛЬТАТИ МОДЕЛЮВАННЯ |"

PRINT "|-----|"

PRINT "| N | прибуття | початок | закінчення | очікування | простою |"

PRINT "| | состава | обробки | обробки | обробки | бригади |"

PRINT "|-----|"

FOR i = 1 TO kol

PRINT USING "| ## | #####.# | #####.# | #####.# | #####.# | #####.# |"; i; x(i); tn(i);  
tk(i); w(i); v(i)

IF i = 20 OR i = 40 OR i = 60 THEN INPUT "", qqq

NEXT i

PRINT "|-----|"

INPUT ""; qqq

FOR i = 1 TO kol

sy = sy + y(i) ' сумарний час обробки составів

sw = sw + w(i) ' сумарний час очікування обробки

IF y(i) > 0 THEN jy = jy + 1 ' кількість оброблених составів

IF w(i) > 0 THEN jw = jw + 1 ' кількість составів, що очікують обробку

IF y(i) > y(maxny) THEN

maxny = i

ELSEIF y(i) < y(minny) THEN

minny = i

END IF

IF w(i) > w(maxnw) THEN

maxnw = i

ELSEIF w(i) < w(minnw) AND w(i) > 0 THEN

minnw = i

END IF

NEXT i

PRINT "Проведемо аналіз результатів моделювання:"

PRINT USING "Кількість оброблених составів ### "; jy

PRINT USING "Час, що затрачено на їх обробку, склав #####.# "; sy

PRINT USING "Середній час обслуговування составів #####.#"; sy / jy

PRINT

PRINT USING "Кількість составів, що очікують обробку ### "; jw

PRINT USING "Час, що затрачений на очікування #####.# "; sw

```

PRINT USING "Середній час очікування склав #####.#"; sw / jw
PRINT
PRINT USING "Найбільший час обслуговування склав #####.# (вагон номер ###)";
y(maxny); maxny
PRINT USING "Найменший час обслуговування склав #####.# (вагон номер ###)";
y(minny); minny
PRINT USING "Найбільший час простою склав #####.# "; maxpw
PRINT USING "Найменший час простою склав #####.# "; minpw
INPUT ""; qq

```

```

LOCATE 23, 1: INPUT "Побудувати графік? (так - 1 / ні - 0)? ", vib
IF vib = 1 THEN CALL graphic(kol, tk())
SCREEN 0
INPUT ""; qq
END SUB

```

```

FUNCTION snormal! (m AS INTEGER, s AS INTEGER)
y = 0
DIM x AS SINGLE
' RANDOMIZE TIMER
FOR i = 0 TO 12 - 1
y = y + RND
NEXT i
x = m + s * (y - 6)
snormal! = x
END FUNCTION

```

```

FUNCTION sravnom! (a AS INTEGER, b AS INTEGER)
DIM x AS SINGLE
' RANDOMIZE TIMER
x = (b - a) * RND + a
sravnom! = x
END FUNCTION

```



Таблиця Д.3.1

## Імітаційна модель транспортного процесу

№ состава (i)	0	1	2	3	4	5	...	i	...
Час подачі состава (x <sub>i</sub> )	x <sub>0</sub>	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	...	x <sub>i</sub>	...
Час обробки состава (y <sub>i</sub> )	y <sub>0</sub>	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>	y <sub>4</sub>	y <sub>5</sub>	...	y <sub>i</sub>	...
Прибуття состава (t <sub>i</sub> )	t <sub>0</sub> =0	t <sub>1</sub> =t <sub>0</sub> +x <sub>0</sub>	t <sub>2</sub> =t <sub>1</sub> +x <sub>1</sub>	t <sub>3</sub> =t <sub>2</sub> +x <sub>2</sub>	t <sub>4</sub> =t <sub>3</sub> +x <sub>3</sub>	t <sub>5</sub> =t <sub>4</sub> +x <sub>4</sub>	...	t <sub>i</sub> =t <sub>i-1</sub> +x <sub>i-1</sub>	...
Початок обробки (tn <sub>i</sub> )	tn <sub>0</sub> =0	tn <sub>1</sub> =					...	$tn_i = \begin{cases} tn_{i-1} + y_{i-1} & \text{npu } t_i < tk_{i-1} \\ t_i & \text{npu } t_i \geq tk_{i-1} \end{cases}$	...
Закінчення обробки (tk <sub>i</sub> )	tk <sub>0</sub> =y <sub>0</sub>	tk <sub>1</sub> =tn <sub>1</sub> +y <sub>1</sub>	tk <sub>2</sub> =tn <sub>2</sub> +y <sub>2</sub>	tk <sub>3</sub> =tn <sub>3</sub> +y <sub>3</sub>	tk <sub>4</sub> =tn <sub>4</sub> +y <sub>4</sub>	tk <sub>5</sub> =tn <sub>5</sub> +y <sub>5</sub>	...	$tk_i = \begin{cases} tn_i + y_i & \text{npu } w_i = 0 \\ tn_i + y_i + w_i & \text{npu } w_i > 0 \end{cases}$	...
Очікування обробки составами (w <sub>i-1</sub> )	w <sub>0</sub> =0						...	$w_i = \begin{cases} 0 & \text{npu } t_i \geq tk_{i-1} \\ tn_i - t_i & \text{npu } t_i < tk_{i-1} \end{cases}$	...
Простої бригади по складах (v <sub>i-1</sub> )	v <sub>0</sub> =0						...	$v_{i-1} = \begin{cases} t_i - tk_{i-1} & \text{npu } t_i \geq tk_{i-1} \\ 0 & \text{npu } t_i < tk_{i-1} \end{cases}$	...