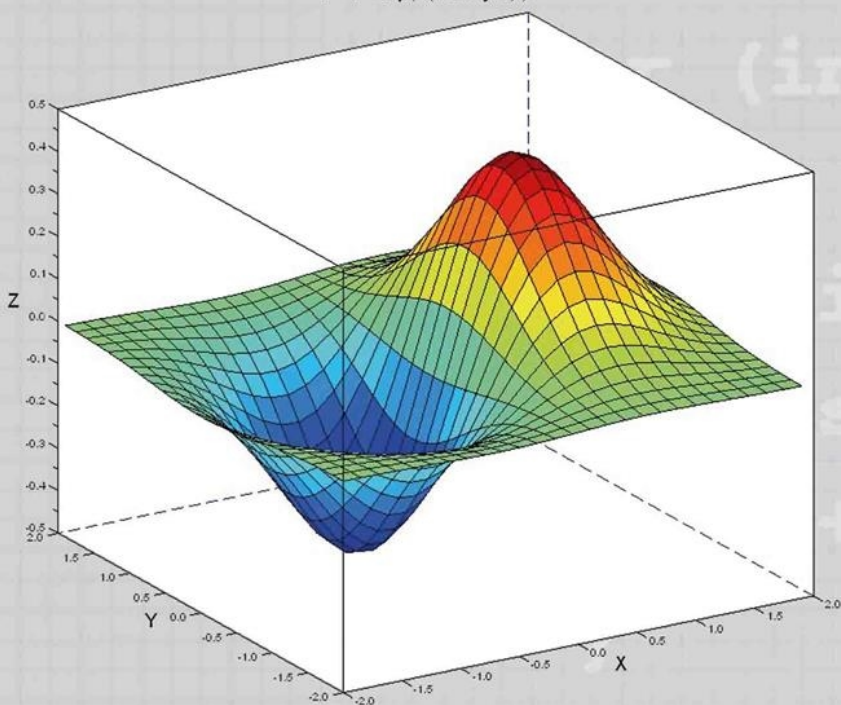


В.Г. Мусіяка

# ОСНОВИ ЧИСЛОВИХ МЕТОДІВ

$$z = x \cdot \exp(-x^2 + y^2)$$



**Міністерство освіти і науки України  
Дніпровський національний університет  
імені Олеся Гончара**

*Присвячено 100-річчю  
Дніпровського  
національного університету  
імені Олеся Гончара*

**В.Г. Мусіяка**

## **ОСНОВИ ЧИСЛОВИХ МЕТОДІВ**

*Ухвалено на вченій раді  
протокол № 8 від 22 грудня 2016 р.*

Дніпро  
2017

УДК 519.6(075.8)  
ББК 22.19я73  
М91

Рецензенти: д-р техн. наук, проф. *В.Л. Красовський*  
д-р фіз.-мат. наук, проф. *В.І. Кузьменко*

**Мусяка В.Г.**

М91 Основи числових методів [Текст] підручник / В.Г. Мусяка. —  
Дніпро : ЛІРА, 2017. – 256 с.  
ISBN 978-966-383-938-7

Викладено основи класичних числових методів, застосованих у наближенні функцій, числовому диференціюванні та інтегруванні, під час розв'язання алгебричних, диференціальних та інтегральних рівнянь. Запропоновано задачі для самостійного розв'язування та навчальні Scilab-програми, які реалізують деякі числові методи, викладені у підручнику.

Для студентів природничих і інженерних факультетів вищих навчальних закладів, а також для тих, хто бажає поглибити власні знання з основ класичних числових методів.

**УДК 519.6(075.8)**  
**ББК 22.19я73**

ISBN 978-966-383-938-7

© Мусяка В.Г., 2017

## ПЕРЕДМОВА

---

Бурхливий розвиток сучасної науки і техніки ставить перед дослідниками і проєктантами все складніші проблеми, що, природно, потребує створення досконаліших математичних моделей. На жаль, час застосування точних (аналітичних) методів для їх вивчення, по суті, завершився ще у ХХ ст. Натомість основним інструментарієм у дослідженні й проєктуванні стали числові методи, для застосування яких практично не існує обмежень. Отже, вивчення майбутніми фахівцями числових методів є нагальна необхідність.

У підручнику розглянуто основи класичних числових методів, застосованих під час розв'язання коректно поставлених математичних задач, до яких зводяться прикладні задачі (задачі інженерної та конструкторської практики).

Детальний виклад розглянутих у підручнику числових методів і багатьох інших можна знайти у працях, наведених у списку рекомендованої літератури. Зазначимо, що це далеко не повний перелік існуючого масиву літератури з числових методів та їх застосування. Це — найпоширеніші джерела, якими тією чи іншою мірою скористався автор у процесі створення підручника. Найбільший вплив на зміст, побудову структурно-логічної схеми та способів викладення матеріалу у цьому підручнику справила надзвичайно змістова книга М.М. Каліткіна.

Зрозуміло, що застосування числових методів, особливо у випадку вивчення складних математичних моделей, зумовлює обов'язкове послуговування ЕОМ.

Якщо при цьому потрібно розв'язувати одиничну задачу на ЕОМ, то доцільно скористатись існуючою стандартною програмою (вбудованою функцією) або ж власноруч скласти програму одним із найпростіших числових методів. Крім того, у разі розв'язання *складної прикладної задачі* її властивості потрібно попередньо вивчити на простій математичній моделі за допомогою добре перевірених методів, що, природно, також може спонукати до використання вбудованих функцій.

Загалом існує безліч задач алгебри, геометрії та аналізу, *за допомогою яких моделюють прикладні задачі* та для яких розроблено вбудовані функції для ЕОМ. Це задачі наближення функцій, обчислення інтегралів невисокої кратності, задачі лінійної алгебри, задача Коші й лінійні крайові задачі для звичайних диференціальних рівнянь, задачі обробки інформації та ін.

У додатках підручника наведено навчальні Scilab-програми, які реалізують деякі числові методи, викладені у виданні.



Автор щиро вдячний В.О. Насекан за прискіпливу й ретельну роботу над текстом підручника, яка сприяла суттєвому його поліпшенню.

## ВСТУП

Задачі, які поширені у розрахунковій та конструкторській практиці, розв'язують різними способами. Один із них — *проведення експерименту*. Виготовивши зразок конструкції і провівши випробування чи дослідження, одержуємо потрібні дані. Якщо ж вони незадовільні, можна виготовити інший зразок і повторити дослідження. На жаль, цей шлях може виявитися повільним, до того ж занадто дорогим. Інший спосіб — *аналіз конструкції або явища математичними методами*. Побудувавши *математичну модель* досліджуваних конструкції чи явища, яка охоплює найважливіші ознаки або ті, що цікавлять нас, (*постановка задачі*), знаходимо розв'язок одержаної математичної задачі. Цей шлях може виявитися значно доцільнішим.

З урахуванням складності реальних явищ чи конструкцій досить трудомісткою роботою в постановці задачі є вдалий вибір зв'язків та характеристик явища, істотних для цієї задачі, які слід формалізувати і включити до математичної моделі.

Наприклад, прогин гнучкої прямолінійної пружної балки  $u$  за чистого згину (коли відношення довжини прольоту балки до її висоти більше ніж 10) визначають за таким рівнянням:

$$\frac{d^2 u}{dx^2} \left[ 1 + \left( \frac{du}{dx} \right)^2 \right]^{-1,5} = \frac{M}{EI(x)}. \quad (\text{В.1})$$

Розв'язку цього нелінійного звичайного диференціального рівняння 2-го порядку в елементарних функціях не існує. Зауважимо, що хоч рівняння (В.1) й одержано для чистого згину, тобто поперечні сили не враховані, але ним інколи можна скористатися також і для дослідження поперечного згину.

Якщо прогин  $u$  невеликий порівняно з довжиною балки, рівняння (В.1) істотно спрощується, бо тоді можна знехтувати величиною  $(du/dx)^2$ , унаслідок чого одержимо лінійне звичайне диференціальне рівняння такого вигляду:

$$\frac{d^2 y}{dx^2} = \frac{M}{EI(x)}. \quad (\text{В.2})$$

Це — наближене рівняння ( $u \approx y$ ) згину балки, яке інтегрується точно, якщо інтеграл від правої частини можна взяти в елементарних функціях.

З цього досить простого прикладу видно, що для точнішої й складнішої математичної моделі аналітичний розв'язок часом одержати важко або й зовсім неможливо. Тому, як правило, основними методами розв'язання прикладних задач для повніших і точніших моделей є числові.

Загалом переважно більшість математичних моделей, які доводиться розв'язувати, умовно можна сформулювати так:

$$A(u, x) = 0, \quad (\text{В.3})$$

де  $u$  — невідомий розв'язок, а величина  $x$  та функція (оператор)  $A$  — відомі.

Зауважимо, що тут  $u$  і  $x$  можуть бути числами, їх масивами або функціями однієї чи багатьох змінних із деякого класу.

У разі розв'язування (В.3) числовими або наближеними методами  $A$ ,  $u$ ,  $x$  замінюють відповідно на  $B$ ,  $y$ ,  $\xi$ , зручніші для обчислень, унаслідок чого й приходимо до задачі

$$B(y, \xi) = 0. \quad (\text{В.4})$$

Природно, заміна має бути такою, щоб розв'язки (В.3) і (В.4) були близькими, тобто  $u \approx y$ .

Зрозуміло, що одержаний числовий розв'язок задачі (В.4) матиме похибки, основними джерелами яких є математична модель, вихідні дані, метод розв'язування та округлення. Перші два джерела породжують так звану *неусувну похибку*, яка матиме місце навіть за точних обчислень. Тому треба намагатися, щоб останні два джерела не збільшували похибку розв'язку. Як правило, слід вибирати такий математичний метод, щоб його похибка була в 2–5 разів менша за неусувну. Ураховуючи, що сучасні ЕОМ працюють з десятьма і більше значущими цифрами, похибка округлення за відсутності систематичних похибок буде істотно менша за похибку методу, а тому нею можна нехтувати порівняно з неусувною та похибкою методу.

У підручнику розглянуто лише *коректно поставлені задачі* типу (В.3), тобто такі, в яких для довільних  $x$  з деякого класу розв'язок  $u$  існує, єдиний і стійкий за вихідними даними.

Існування розв'язку, а особливо його єдиність, потрібні тому, що числовий метод завжди приводить до єдиного розв'язку. Стосовно стійкості задачі за вихідними даними, то вона має місце лише тоді, коли розв'язок неперервно залежить від вихідних даних, тобто коли для  $|\Delta x| \rightarrow 0$  маємо  $|\Delta u| \rightarrow 0$ . Якщо ж ця умова не виконується, задача є нестійка за вихідними даними.

Для розв'язання некоректно поставлених задач розроблено так звані *методи регуляризації*. Згідно з ними вихідну задачу замінюють на коректно поставлену, що містить деякий параметр, у разі прямування його до нуля розв'язок останньої наближається до розв'язку вихідної задачі.

Зауважимо, що числові методи майже завжди потребують застосування ЕОМ, що в свою чергу спонукає до розробки алгоритму й побудови програми. Отже, перед будь-яким дослідником стоїть проблема вибору програмного забезпечення. Зазвичай надають перевагу сучасним вільно поширюваним пакетам. Саме таким є пакет наукових програм Scilab з потужним *відкритим* середовищем для інженерних і наукових розрахунків.

Scilab дозволяє працювати з елементарними функціями і великим числом спеціальних функцій, має потужні засоби роботи з матрицями (у тому числі і з деякими символьними), поліномами, проводити числові розрахунки і розв'язання задач лінійної алгебри, оптимізації і симуляції, могутні статистичні функції, а також засоби для побудови графіків і роботи з ними. Вищенаведене й зумовило вибір Scilab для розв'язання прикладних задач числовими методами.

У додатках розміщено 40 навчальних програм на Scilab, які реалізують деякі числові методи, викладені у підручнику, та коротку інформацію про елементи мови пакета Scilab.



# Розділ 1

## НАБЛИЖЕННЯ ФУНКЦІЙ

---

Задача наближення функцій полягає в заміні складної аналітичної або таблично заданої функції  $y(x)$  на зручнішу для обчислень апроксимувальну функцію  $\varphi(x)$ , яка б для всіх необхідних нам значень аргументу задовольняла співвідношення  $y(x) \approx \varphi(x)$ . Цього досягають введенням вектора вільних параметрів  $C = (C_1, C_2, \dots, C_n)$  до  $\varphi(x)$  [тобто  $\varphi(x) \equiv \varphi(x; C)$ ] і його визначенням із деякої взятої умови близькості функцій  $y(x)$  та  $\varphi(x)$ . Зазвичай вважають, що графік таблично заданої функції є плавна крива.

Необхідність у застосуванні наближення функцій виникає у разі обробки експериментальних даних або у випадку числового розв'язання задачі, тобто для відновлення значень деякої функції в точках, що нас цікавлять, за її відомими значеннями.

### 1.1. Інтерполяція

---

Якщо за умову близькості  $\varphi(x)$  і  $y(x)$  взяти умову їх збіжності в  $n$  точках (вузлах інтерполяції)  $x_i$  ( $i = \overline{1, n}$ ), то вектор вільних параметрів  $C$  визначають як результат розв'язання системи алгебричних рівнянь

$$\varphi(x_i; C) = y(x_i) \equiv y_i \quad (i = \overline{1, n}). \quad (1.1)$$

Це так звана *лагранжова* інтерполяція.

Якщо  $C_i$  ( $i = \overline{1, n}$ ) входять до  $\varphi(x)$  лінійно, то інтерполяцію називають *лінійною*, у протилежному випадку — *нелінійною*.

### 1.1.1. Лінійна інтерполяція

Задача визначення складових вектора  $C$  суттєво спрощується, якщо  $\varphi(x)$  — узагальнений поліном, тобто

$$\varphi(x; C) = \sum_{k=1}^n C_k \varphi_k(x), \quad (1.2)$$

де  $\varphi_k(x)$  — система лінійно незалежних функцій.

Підставивши (1.2) в (1.1), матимемо таку систему лінійних алгебричних рівнянь для визначення вектора вільних параметрів  $C$ :

$$\sum_{k=1}^n C_k \varphi_k(x_i) = y_i \quad (i = \overline{1, n}). \quad (1.1')$$

Для існування та єдиності розв'язку задачі інтерполяції визначник СЛАР (1.1') не має перетворюватися на нуль (зрозуміло, що при цьому серед вузлів інтерполяції не може бути однакових).

Найпростішою для обчислень системою функцій є алгебричні поліноми  $\varphi_k(x) = x^k$  ( $k = \overline{0, n}$ ). Для зручності в цьому разі вузли інтерполяції також нумерують від 0 до  $n$ .

**Інтерполяційний поліном Лагранжа.** Ж. Лагранж запропонував алгебричний інтерполяційний поліном брати таким:

$$\varphi(x; C) \equiv P_n(x) = \sum_{k=0}^n y_k L_k^{(n)}(x).$$

Очевидно, тут у вузлах інтерполяції  $x_i$  ( $i = \overline{0, n}$ ) має бути така залежність:

$$L_k^{(n)}(x_i) = \delta_{ik} \equiv \begin{cases} 1 & (i = k) \\ 0 & (i \neq k). \end{cases}$$

Покладаючи

$$L_k^{(n)}(x) = C_k \prod_{i \neq k} (x - x_i),$$

одержимо, що для всіх вузлів, крім  $x_k$ ,  $L_k^{(n)}(x_i) = 0$ .

З умови

$$L_k^{(n)}(x_k) = 1;$$

$$C_k = 1 / \prod_{i \neq k} (x_k - x_i).$$

Отже, можна вважати, що

$$y(x) \approx \varphi(x; C) \equiv P_n(x) = \sum_{k=0}^n y_k \left( \prod_{i \neq k} \frac{x - x_i}{x_k - x_i} \right). \quad (1.3)$$

Інтерполяційний поліном у вигляді (1.3) називають *інтерполяційним поліномом Лагранжа*.

Для  $n=1$  і  $n=2$  інтерполяційні поліноми Лагранжа (1.3) відповідно мають вигляд

$$P_1(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0};$$

$$P_2(x) = y_0 \frac{x - x_1}{x_0 - x_1} \frac{x - x_2}{x_0 - x_2} + y_1 \frac{x - x_0}{x_1 - x_0} \frac{x - x_2}{x_1 - x_2} + y_2 \frac{x - x_0}{x_2 - x_0} \frac{x - x_1}{x_2 - x_1}.$$

Іноколи ці інтерполяційні поліноми називають відповідно лінійним і квадратичним.

Із (1.3) видно, що інтерполяційний поліном Лагранжа явно залежить від  $y_k$ , а це іноді зручно, але, на жаль, зі збільшенням вузлів усі доданки в ньому потрібно перераховувати.

**Приклад 1.1.** Побудуємо інтерполяційний поліном Лагранжа  $P_2(x)$  для функції  $y(x) = \sin \pi x / 6$  і знайдемо за його допомогою значення  $\sin \pi / 4$ , тобто обчислимо  $P_2(3/2)$ . Далі, посилаючись на приклад 1.1, писатимемо прикл. 1.1.

Якщо за вузли взяти точки  $x = 0, 1, 3$ , де функція відповідно набуває значень  $y = 0, 1/2, 1$ , і скористатися формулою для  $P_2(x)$ , одержимо, що

$$y(x) \approx P_2(x) = x(7 - x) / 12.$$

Отже,  $P_2(3/2) = 11/16 = 0,6875$ .

Оскільки табличне значення  $y(3/2) = \sqrt{2}/2 = 0,7071$ , то фактична похибка інтерполяції становитиме

$$y(3/2) - P_2(3/2) = 0,0196.$$

**Інтерполяційний поліном Ньютона.** Для його побудови введемо так звані поділені різниці:

$$y(x_i, x_j) = \frac{y(x_i) - y(x_j)}{x_i - x_j}; \quad y(x_i, x_j, x_k) = \frac{y(x_i, x_j) - y(x_j, x_k)}{x_i - x_k}, \dots$$

Із запису поділених різниць випливає, що вони мають розмірність похідних, тому поділені різниці інколи використовують як наближені їх значення.

Розглянемо поділені різниці алгебричного полінома  $n$ -го степеня  $P(x)$  (тут для спрощення індекс  $n$  біля  $P$  опущено). Якщо від  $P(x)$  відняти  $P(x_0)$ , одержимо знову алгебричний поліном  $n$ -го степеня, який стає нулем за  $x = x_0$ . Зрозуміло, що він ділиться на  $(x - x_0)$  без остачі.

Отже, перша поділена різниця  $P(x, x_0) = \frac{P(x) - P(x_0)}{x - x_0}$  є поліномом  $(n - 1)$ -го степеня відносно  $x$  та  $x_0$ . Очевидно, що тоді друга поділена різниця  $P(x, x_0, x_1)$  буде поліномом  $(n - 2)$ -го степеня і т. д.

Можна довести, що  $n$ -та поділена різниця —  $P(x, x_0, x_1, \dots, x_{n-1})$  буде поліномом нульового степеня, тобто сталою, оскільки поліном було поділено  $n$  разів на біноми вигляду  $(x - x_i)$  ( $i = 0, n - 1$ ). Тоді очевидно, що поділені різниці, починаючи з  $(n + 1)$ -ї, дорівнюють нулю.

Розв'язуючи першу поділену різницю відносно  $P(x)$ , одержимо

$$P(x) = P(x_0) + (x - x_0)P(x, x_0).$$

Виконавши ті ж самі дії з другою поділеною різницею і знайшовши  $P(x, x_0)$ , матимемо

$$P(x) = P(x_0) + (x - x_0)[P(x, x_0) + (x - x_1)P(x, x_0, x_1)]$$

і т. д.

З огляду на те що поділені різниці, починаючи з  $(n + 1)$ -ї, нульові, ланцюг виразів буде скінченним, тобто можна записати:

$$P(x) = P(x_0) + \sum_{k=1}^n P(x_0, x_1, \dots, x_k) \prod_{i=0}^{k-1} (x - x_i).$$

З умови збіжності інтерполяційний поліном із заданою функцією та її поділеною різницею у вузлах інтерполяції приходимо до такого виразу для інтерполяційного полінома Ньютона:

$$y(x) \approx \varphi(x; C) \equiv P_n(x) = y(x_0) + \sum_{k=1}^n y(x_0, x_1, \dots, x_k) \prod_{i=0}^{k-1} (x - x_i). \quad (1.4)$$

Наведені тут інтерполяційні поліноми Лагранжа та Ньютона застосовують для таблиць зі змінним кроком.

Для обчислення функції  $y(x)$  за допомогою інтерполяційного полінома Ньютона потрібно скласти таблицю її поділених різниць вигляду (табл. 1.1), а потім наближені значення  $y(x)$  знаходити за формулою (1.4).

*Таблиця 1.1*

$x_0$	$y(x_0)$		
$x_1$	$y(x_1)$	$y(x_0, x_1)$	$y(x_0, x_1, x_2)$
$x_2$	$y(x_2)$	$y(x_1, x_2)$	

Зауважимо, що на практиці перевагу віддають інтерполяційному поліному Ньютона, оскільки збільшення в ньому кількості вузлів не потребує перерахування всіх доданків у (1.4) та й порядок нумерації вузлів при цьому у ньому довільний.

**Приклад 1.2.** Побудуємо інтерполяційний поліном Ньютона для цієї самої функції  $y(x) = \sin \pi x / 6$  за використання тих самих вузлів, що й у разі побудови інтерполяційного полінома Лагранжа у прикл. 1.1.

Поділені різниці функції  $y(x)$  наведено в табл. 1.2.

*Таблиця 1.2*

$i$	$x_i$	$y(x_i)$	$y(x_i, x_{i+1})$	$y(x_i, x_{i+1}, x_{i+2})$
0	0	0		
1	1	1/2	1/2	
2	3	1	1/4	-1/12

Підставляючи знайдені поділені різниці в (1.4) та користуючись верхнім скісним рядком цієї таблиці, одержимо вираз

$$y(x) \approx P_2(x) = 0 + x(1/2) + x(x-1)(-1/12) = x(7-x)/12.$$

Отже, інтерполяційні поліноми Лагранжа та Ньютона збігаються, що й повинно бути. Реалізацію розрахунків за інтерполяційним поліномом Ньютона наведено у дод. 1.

$$\text{Тут } \delta = [y(1,5) - P_2(1,5)] / y(1,5) \cdot 100 = 2,77\%$$

*Зауваження.* Якщо під час інтерполяції за допомогою інтерполяційного полінома Ньютона число вузлів фіксоване, то розрахунки краще вести за схемою Горнера:

$$y(x) \approx y(x_0) + (x - x_0)(y(x_0, x_1) + (x - x_1)(y(x_0, x_1, x_2) + \dots)),$$

що може зменшити витрати машинного часу на обчислення.

Якщо ж точність обчислень контролюється візуально, то краще скористатися формулою (1.4), обмежуючись доданками, які більші за абсолютною величиною від заданої похибки обчислень  $\varepsilon$ .

За сталого кроку таблиці ( $h = x_{i+1} - x_i = \text{const}$ ) доцільніше застосовувати таку формулу:

$$y(x_0 + th) \approx y(x_0) + \sum_{k=1}^n t(t-1)\dots[t-(k-1)] \Delta^k y_0 / k!,$$

де

$$\Delta^k y_0 = k! y(x_0, x_1, \dots, x_k) h^k; \quad t = \frac{x - x_0}{h} \quad \text{або} \quad x = x_0 + th.$$

Крім розглянутих інтерполяційних поліномів, застосовують також інші, наприклад, інтерполяційні поліноми Ерміта, Гаусса, але як для ручних, так і для машинних розрахунків зручніше використовувати формулу (1.4).

**Залишковий член (похибка) інтерполяції інтерполяційним поліномом Ньютона.** У разі візуального контролю розрахунків за формулою (1.4) похибка інтерполяції приблизно дорівнює першому відкинутому доданку (*апостеріорна оцінка*).

Якщо ж функція  $y(x)$  задана аналітично, то оцінити похибку можна ще до початку розрахунків (*априорно*).

Позначимо похибку інтерполяції алгебричним поліномом так:

$$r_n(x) \equiv y(x) - P_n(x) = \rho(x)\omega(x) \quad \left[ \omega(x) = \prod_{i=0}^n (x - x_i) \right].$$

Зрозуміло, що  $r_n(x)$  дорівнює нулю у вузлах  $x_0, x_1, \dots, x_n$ .

Розглянемо функцію

$$q(\xi) = y(\xi) - P_n(\xi) - \rho(x)\omega(\xi),$$

де  $x$  є параметр та може мати довільне фіксоване значення.

Очевидно,  $q(\xi) = 0$  в точках  $\xi = x, x_0, x_1, \dots, x_n$ , тобто в  $n+2$  точках.

Нехай функція  $y(x)$  має  $(n+1)$ -у неперервну похідну, що, звичайно, стосується й  $q(\xi)$ . Нагадаємо, що за теоремою Ролля між двома нулями гладкої функції лежить один нуль її похідної. Послідовно застосовуючи теорему Ролля, дійдемо висновку, що між крайніми  $n+2$  нулями функції обов'язково лежить нуль її  $n+1$ -ї похідної. Проте оскільки  $P_n^{(n+1)}(x) = 0$ , то

$$q^{(n+1)}(\xi) = y^{(n+1)}(\xi) - (n+1)!\rho(x).$$

Якщо в якійсь точці  $\xi^*$ , що лежить між зазначеними вище нулями,  $q^{(n+1)}(\xi^*) = 0$ , тоді

$$\rho(x) = \frac{y^{(n+1)}(\xi^*)}{(n+1)!}.$$

Замінивши  $y^{(n+1)}(\xi^*)$  на її максимально можливе значення, одержимо таку оцінку похибки:

$$|r_n(x)| \leq |y(x) - P_n(x)| = \frac{|\omega(x)|M_{n+1}}{(n+1)!}, \quad (1.5)$$

де  $M_{n+1} = \max |y^{(n+1)}(\xi)|$ , який береться по відрізку  $[a, b]$ . Тут

$$a = \min(x_0, x_1, \dots, x_n), \quad b = \max(x_0, x_1, \dots, x_n).$$

У разі довільного розміщення вузлів інтерполяції оцінити похибку досить важко, але якщо сітка рівномірна, до того ж вузли розміщені підряд, то біля центральних вузлів відрізка  $[a, b]$  екстремуми  $\omega(x)$  невеликі, а біля крайніх вузлів дещо більші. Якщо ж точка  $x$  виходить за крайні вузли, то  $\omega(x)$  досить швидко зростає. Тому для значення  $x$ , яке розташоване поза крайніми вузлами інтерполяції (*екстраполяція*), розрахунки стають ненадійними. Отже, для зменшення похибки точку  $x$  слід розмістити ближче до центральних вузлів відрізка інтерполяції.

Можна показати, що за  $h = \text{const}$  із (1.5) можна одержати таку оцінку похибки в середині відрізка:

$$|r_n(x)| < \sqrt{\frac{2}{n\pi}} M_{n+1} \left(\frac{h}{2}\right)^{n+1} \approx 0(h^{n+1}). \quad (1.5')$$

Виходячи з цього констатуємо, що інтерполяція поліномом Ньютона забезпечує  $(n + 1)$ -й порядок точності.

**Приклад 1.3.** Знайдемо залишковий член інтерполяції (*похибку методу*) для функції  $y(x) = \sin \pi x / 6$  у точці  $x = 3/2$ , застосувавши формулу (1.5). Тут

$$y'''(x) = -\left(\frac{\pi}{6}\right)^3 \cos \frac{\pi x}{6}, \quad M_3 = \max |y'''(x)| = \left(\frac{\pi}{6}\right)^3;$$

$$\begin{aligned} |\omega(3/2)| &= |x(x-1)(x-3)|_{x=3/2} = \\ &= |(3/2)(3/2-1)(3/2-3)| = 9/8 = 1.125. \end{aligned}$$

Отже, значення залишкового члена інтерполяції у цьому разі матиме вигляд:

$$|r_{\frac{1}{2}}(3/2)| \leq \left(\frac{\pi}{6}\right)^3 \frac{9/8}{6} = 0.0269.$$

Він, як і належить, більший за фактичну похибку інтерполяції, знайденої з чотирма правильними цифрами в прикл. 1.1.



**Про застосування та збіжність інтерполяції.** Крім основної задачі інтерполяції (відновлення значень функції), її застосовують для *тестування таблиць* на наявність несистематичних похибок (друга поділена різниця поводить хаотично), *субтабулювання* (зменшення кроку існуючої таблиці) та *оберненої інтерполяції* (знаходження значення аргументу для заданого чи відомого значення функції).

Похибку алгебричної інтерполяції можна зменшити у такий спосіб: зберегти степінь інтерполяційного полінома, зменшивши крок, або зберегти крок, збільшивши кількість використовуваних вузлів. У *практичних розрахунках* збільшення  $n$  небажане (бо воно збільшує похибку), а тому для підвищення точності інтерполяції слід обмежитися 4 – 6 вузлами (точніше, вільними параметрами) і зменшити крок таблиці, не змінюючи  $n$ .

Якщо відомі значення функції та її першої похідної, то краще скористатися для відновлення функції **інтерполяційним поліномом Ерміта**:

$$H(x) = y_{i-1} + (x - x_{i-1}) \left\{ y'_{i-1} + \frac{x - x_{i-1}}{x_{i-1} - x_i} \left[ y'_{i-1} - y'(x_{i-1}, x_i) \right] + \frac{x - x_{i-1}}{x_{i-1} - x_i} \left[ y'_{i-1} - 2y(x_{i-1}, x_i) + y'_i \right] \right\} [x_{i-1} \leq x \leq x_i; i = \overline{1, N}].$$

Ця формула точно передає значення функції  $(y_{i-1}, y_i)$  та її похідних  $(y'_{i-1}, y'_i)$  у точках  $(x_{i-1}, x_i)$ , а між ними апроксимує функцію  $y(x)$  поліномом 3-го степеня.

### 1.1.2. Нелінійна інтерполяція

Зазначимо, що за рівномірного кроку сітки алгебрична інтерполяція має похибку порядку  $M_{n+1} \left(\frac{h}{2}\right)^{n+1}$ , тобто зі збільшенням степеня інтерполяційного полінома на одиницю похибка змінюється в  $\frac{hM_{n+2}}{2M_{n+1}}$  разів. Зрозуміло, якщо крок малий, то похибка зменшуватиметься, якщо ж крок великий, то похи-

бка зростатиме. Це обов'язково слід ураховувати, якщо функція швидкозростаюча.

Універсальних способів інтерполяції таких функцій немає, але для кожної функції іноді вдається знайти окремий спосіб інтерполяції. Здебільшого це *нелінійна інтерполяція*.

Для побудови зручних формул у таких випадках потрібно мати додаткову інформацію щодо якісної поведінки функції  $y(x)$ , яку часто можна одержати або з фізичного змісту  $y(x)$ , або з порівняння графіка функції  $y(x)$  з графіками добре вивчених функцій, насамперед елементарних.

Після вивчення якісної поведінки  $y(x)$  підбирають такі прямі перетворення  $\eta = \eta(y)$  та  $\xi = \xi(x)$ , щоб графік  $\eta(\xi)$  на декількох кроках сітки мало різнився від прямої (*вирівнювання*). Тоді для одержання прийнятної точності знаходження  $\eta(\xi)$  можна скористатися інтерполяційним поліномом невисокого степеня. Для обчислення будують таблицю  $\eta_i = \eta(\xi_i)$ , по ній інтерполюють, а потім *оберненою* інтерполяцією знаходять  $y = y(\eta)$ .

Визначимо деякі найважливіші аспекти нелінійної інтерполяції.

*По-перше*, майже кожна конкретна функція потребує своїх прямих перетворень  $\eta(y)$ ,  $\xi(x)$ .

*По-друге*, оцінювати точність інтерполяції на практиці слід за швидкістю зменшення доданків в інтерполяційному поліномі Ньютона для  $\eta(\xi)$ .

*По-третє*, як прямі перетворення  $\eta(y)$ ,  $\xi(x)$ , так і обернене  $y = y(\eta)$  мають бути простими, бо інакше *метод вирівнювання* буде малоєфективним у практичних розрахунках (слід застосовувати логарифмічні, експоненціальні, тригонометричні або інші елементарні функції).

*По-четверте*, якщо вирівнювальні перетворення змінних прості, то іноді  $f(x)$  можна виразити через табличні значення функції у вихідних змінних (на одному-двох кроках таблиці).

Наприклад, якщо функція  $y(x)$  близька до експоненти, то після вирівнювання за допомогою перетворень  $\xi = x$ ,  $\eta = \ln y(x)$  її можна апроксимувати двоточковим інтерполяційним поліномом Ньютона:

$$\eta \approx \eta_{i-1} + \frac{(\eta_i - \eta_{i-1})(\xi - \xi_{i-1})}{\xi_i - \xi_{i-1}} \quad (\xi_{i-1} \leq \xi \leq \xi_i).$$

Повернувшись до вихідних змінних, одержимо такий вираз:

$$y(x) \approx y_{i-1} \exp \left[ (x - x_{i-1}) \ln \left( \frac{y_i}{y_{i-1}} \right) / (x_i - x_{i-1}) \right] \\ (x_{i-1} \leq x \leq x_i). \quad (1.6)$$

*По-н'яте*, якщо у вихідних змінних інтерполяція нелінійна відносно складових вектора вільних параметрів  $C$ , а в змінних  $\xi$  і  $\eta$  — близька до лінійної, то вона мало ускладнює роботу і має назву *квазілінійної*; якщо ж не можна знайти перетворення, в яких залежність відносно вільних параметрів майже лінійна, наприклад, якщо функція близька до  $y(x) \approx a(x+b)^c$ , то виконати інтерполяцію нелегко, відтак її називають *істотно нелінійною*. Останню на практиці застосовують досить рідко.

### 1.1.3. Сплайн-інтерполяція

В інженерній практиці графік функції  $y(x_i)$  ( $i = \overline{0, N}$ ) будуть в основному за допомогою лекал. Якщо ж точки розташовані досить рідко, то беруть гнучку лінійку (spline), ставлять її на ребро і згинають так, щоб вона одночасно проходила через усі точки.

Оскільки наближене рівняння чистого згину пружного бруса сталої жорсткості має вигляд  $y^{IV}(x) = 0$ , то можна припустити, що її форма між вузлами є алгебричний поліном 3-го степеня.

Тому, очевидно, інтерполювальну функцію між кожними двома вузлами можна взяти, наприклад, таку:

$$S^{(i)}(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3 \quad (1.7) \\ [x_{i-1} \leq x \leq x_i; \quad i = \overline{1, N}].$$

Невідомі коефіцієнти  $a_i$ ,  $b_i$ ,  $c_i$ ,  $d_i$  знайдемо з умов у вузлах інтерполяції.

З огляду на те що поліноми мають збігатися з табличними значеннями функції  $y(x_i)$  ( $i = \overline{0, N}$ ) у вузлах інтерполяції, це приводить до таких рівнянь:

$$S^{(i)}(x_{i-1}) \equiv a_i = y_{i-1}; \quad (A)$$

$$S^{(i)}(x_i) \equiv a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = y_i \quad (h_i = x_i - x_{i-1}). \quad (B)$$

Оскільки цих рівнянь удвічі менше, ніж невідомих коефіцієнтів, то потрібні ще й додаткові умови (наприклад, умови неперервності 1- та 2-ї похідних у всіх точках, утім числі й у вузлах інтерполяції, тобто умови гладкості кута повороту перетину та кривизни лінійки).

Отже, умови неперервності похідних у внутрішніх вузлах ( $i = \overline{1, N-1}$ ) такі:

$$[S^{(i)}(x_i)]' = [S^{(i+1)}(x_i)]'; \quad (C)$$

$$[S^{(i)}(x_i)]'' = [S^{(i+1)}(x_i)]''. \quad (D)$$

Знайдемо вирази для похідних від сплайна  $S^{(i)}(x)$ :

$$[S^{(i)}(x)]' = b_i + 2c_i(x - x_{i-1}) + 3d_i(x - x_{i-1})^2; \quad (E)$$

$$[S^{(i)}(x)]'' = 2c_i + 6d_i(x - x_{i-1}) \quad (F)$$

і підставимо їх у вирази (C) і (D). Як наслідок для ( $i = \overline{1, N-1}$ ) матимемо такі співвідношення:

$$b_{i+1} = b_i + 2c_i h_i + 3d_i h_i^2; \quad (G)$$

$$c_{i+1} = c_i + 3d_i h_i. \quad (H)$$

Для одержання ще двох необхідних рівнянь скористаємось умовами в кінцевих вузлах. Наприклад, можна вважати кінці лінійки вільно відпущеними, що відповідає їх нульовій кривизні, тобто

$$[S^{(1)}(x_0)]'' / 2 \equiv c_1 = 0; \quad (I)$$

$$\left[ S^{(N)}(x_N) \right]'' / 2 \equiv c_N + 3d_N h_N = 0 \quad (c_{N+1} = 0). \quad (J)$$

Побудовані за таких умов кубічні сплайни називають *вільними*. Природно, що за інших відомих асимптотичних відомостей про задачу можливі й інші умови на кінцях.

Рівняння (A), (B), (G) – (J) утворюють повну СЛАР для визначення  $4N$  невідомих коефіцієнтів. Якщо цю СЛАР дещо перетворити, то її розв'язання істотно спроститься.

Очевидно,  $a_i = y_{i-1}$  ( $i = \overline{1, N}$ ). Крім того, із залежності (J) випливає, що

$$d_N = -\frac{c_N}{3h_N}, \quad (K)$$

а з виразу (H) —

$$d_i = \frac{c_{i+1} - c_i}{3h_i} \quad (i = \overline{1, N-1}). \quad (L)$$

Далі, підставивши рівняння (L) у формулу (B) і врахувавши, що  $a_i = y_{i-1}$ , одержимо такі співвідношення:

$$b_i = (y_i - y_{i-1}) / h_i - h_i(c_{i+1} + 2c_i) / 3 \quad (i = \overline{1, N-1}); \quad (M)$$

$$b_N = (y_N - y_{N-1}) / h_N - 2h_N c_N / 3 \quad (c_{N+1} = 0). \quad (N)$$

Вилучаючи тепер з (G)  $b_i$  та  $b_{i+1}$  за допомогою (M) (зазвичай, для  $b_{i+1}$  індекс потрібно збільшити на одиницю), а величину  $d_i$  — на підставі (L), прийдемо до такої СЛАР відносно лише  $c_i$ :

$$\begin{aligned} & h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = \\ & = 3 \left( \frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}} \right) \quad (i = \overline{2, N}); \quad (1.8) \\ & c_1 = c_{N+1} = 0. \end{aligned}$$

Матриця цієї СЛАР тридіагональна, тобто не дорівнюють нулю лише елементи на головній та двох сусідніх діагоналях.

Для її розв'язання (знаходження  $c_i$ ) слід скористатись методом алгебричної прогонки (див. п. 5.1.1), після чого визначити  $b_i$  та  $d_i$  з виразів (K) – (N).

Доцільно розглянути задачу про знаходження сплайна  $n$ -го степеня загального вигляду:

$$S_n^{(i)}(x) = \sum_{k=0}^n c_{ik} x^k \quad (x_{i-1} \leq x \leq x_i; i = \overline{1, N}),$$

коефіцієнти якого кусково-сталі та який у вузлах інтерполяції набуває значень заданої функції й неперервний разом зі своїми  $n - 1$  похідними.

На практиці, як правило, використовують кубічні або лінійні сплайни. Останній збігається з інтерполяційним поліномом Ньютона 1-го степеня, що відповідає заміні графіка  $y(x)$  на ламану.

Сплайн-інтерполяцію рекомендують застосовувати, коли сітка не така щільна, щоб вдаватись до лагранжової інтерполяції, і не настільки рідка, щоб скористатися нелінійною інтерполяцією.

**Приклад 1.4.** Побудуємо вільні кубічні сплайни, які апроксимують таку табличну функцію  $y(x)$  (табл. 1.3), і знайдемо за їх допомогою наближені значення заданої функції в точках  $x = -1/2, 1/2, 3/2$ .

Таблиця 1.3

$i$	0	1	2	3
$x_i$	-1	0	1	2
$y_i$	-2	1	4	15

Тут  $N = 3$  і  $h = 1$ , тобто маємо три відрізки, на яких слід побудувати вільні кубічні сплайни. Підставляючи значення  $h$  і  $y_i$  в (1.8) і враховуючи, що  $c_1 = c_4 = 0$ , одержимо таку СЛАР:

$$i = 2: 1 \times 0 + 2 \times 2 \times c_2 + 1 \times c_3 = 3 \times \left\{ \frac{4 - 1}{1} - \frac{[1 - (-2)]}{1} \right\},$$

$$i = 3: 1 \times c_2 + 2 \times 2 \times c_3 + 1 \times 0 = 3 \times \left\{ \frac{15-4}{1} - \frac{4-1}{1} \right\}$$

або  $4c_2 + c_3 = 0, \quad c_2 + 4c_3 = 24.$

Звідси знайдемо  $c_3 = 32/5; \quad c_2 = -8/5.$

Далі, урахувавши залежність (А) і підставивши  $c_2$  і  $c_3$  у вирази (К) – (N), одержимо коефіцієнти вільних кубічних сплайнів (1.7):

$$a_1 = y_0 = -2, \quad a_2 = y_1 = 1, \quad a_3 = y_2 = 4;$$

$$b_1 = \frac{1 - (-2)}{1} - 1 \times \frac{-8/5 + 0}{3} = \frac{53}{15},$$

$$b_2 = \frac{4 - 1}{1} - 1 \times \frac{\left[ \frac{32}{5} + 2 \times \frac{-8}{5} \right]}{3} = \frac{29}{15},$$

$$b_3 = \frac{15 - 4}{1} - 2 \times \frac{32}{5} \times \frac{1}{3} = \frac{101}{15};$$

$$d_1 = \frac{\frac{-8}{5} - 0}{3 \times 1} = -\frac{8}{15}, \quad d_2 = \frac{\frac{32}{5} - \frac{-8}{5}}{3 \times 1} = \frac{8}{3}, \quad d_3 = -\frac{\frac{32}{5}}{3 \times 1} = -\frac{32}{15}.$$

Обчислені за допомогою побудованих вільних кубічних сплайнів наближені значення функції  $y(x)$  у заданих точках наведені в таблиці 1.4:

Таблиця 1.4

$x$	-0,5	0,5	1,5
$y$	-0,3	1,9	8,7

Реалізацію цих розрахунків за вільними кубічними сплайнами наведено в дод. 2.

### 1.1.4. Інтерполяція функції двох змінних

Оскільки в процесі практичних розрахунків інколи виникає потреба апроксимації функції декількох змінних (наприклад,

після розв'язання параболічного чи гіперболічного рівняння), розглянемо функцію двох змінних  $u = u(x, t)$ , значення якої задано в трьох точках:

$$(x_i, t_i, u_i) \quad (i = \overline{1,3}),$$

де

$$u_i = u(x_i, t_i).$$

З аналітичної геометрії відомо, що рівняння площини, яка проходить через три задані точки, має вигляд

$$\begin{vmatrix} x - x_1 & t - t_1 & u - u_1 \\ x_2 - x_1 & t_2 - t_1 & u_2 - u_1 \\ x_3 - x_1 & t_3 - t_1 & u_3 - u_1 \end{vmatrix} = 0.$$

Розкриваючи цей визначник, одержимо інтерполяційний поліном 1-го степеня, за допомогою якого можна знайти наближене значення функції  $u = u(x, t)$ , а саме:

$$u(x, t) \approx \varphi(x, t) = A \cdot x + B \cdot t + C$$

$$(A = -D_1 / D_3, \quad B = D_2 / D_3, \quad C = D_0 / D_3), \quad (1.9)$$

де

$$D_0 = \begin{vmatrix} x_1 & t_1 & u_1 \\ x_2 & t_2 & u_2 \\ x_3 & t_3 & u_3 \end{vmatrix}; \quad D_1 = \begin{vmatrix} t_1 & u_1 & 1 \\ t_2 & u_2 & 1 \\ t_3 & u_3 & 1 \end{vmatrix};$$

$$D_2 = \begin{vmatrix} x_1 & u_1 & 1 \\ x_2 & u_2 & 1 \\ x_3 & u_3 & 1 \end{vmatrix}; \quad D_3 = \begin{vmatrix} x_1 & t_1 & 1 \\ x_2 & t_2 & 1 \\ x_3 & t_3 & 1 \end{vmatrix}.$$

**Приклад 1.5.** Знайдемо наближене значення функції  $u = u(x, t)$  у точці  $(2, 1)$ , скориставшись виразом (1.9), якщо відомі три її значення:

$$u_1 = u(0, 0) = 0; \quad u_2 = u(2, 4) = -3; \quad u_3 = u(4, -2) = 1.$$

Тут

$$D_0 = \begin{vmatrix} 0 & 0 & 0 \\ 2 & 4 & -3 \\ 4 & -2 & 1 \end{vmatrix} = 0; \quad D_1 = \begin{vmatrix} 0 & 0 & 1 \\ 4 & -3 & 1 \\ -2 & 1 & 1 \end{vmatrix} = -2;$$



$$D_2 = \begin{vmatrix} 0 & 0 & 1 \\ 2 & -3 & 1 \\ 4 & 1 & 1 \end{vmatrix} = 14; \quad D_3 = \begin{vmatrix} 0 & 0 & 1 \\ 2 & 4 & 1 \\ 4 & -2 & 1 \end{vmatrix} = -20.$$

Звідси  $u(x, t) \approx \varphi(x, t) = -0,1x - 0,7t$

Отже,  $u(2, 1) \approx -0,9$ .

Реалізацію розрахунків за (1.9) наведено в дод. 3.

## 1.2. СЕРЕДНЄ КВАДРАТИЧНЕ НАБЛИЖЕННЯ

Оскільки інтерполяція забезпечує зручну апроксимацію лише на невеликому відрізку, тобто на декількох вузлах інтерполяції, то якщо потрібно мати єдину апроксимувальну функцію  $\varphi(x; C)$  на значному відрізку або ж коли значення заданої функції  $y(x)$  знайдено неточно (скажімо, експериментально), слід вдатися до *середнього квадратичного наближення*. Воно передає головні ознаки поведінки функції  $y(x)$ , а в разі неточних даних згладжує локальні похибки (тобто апроксимує функцію  $y(x)$  у середньому).

У середньо квадратичному наближенні вектор вільних параметрів  $C$  визначають із такої умови:

$$\|y(x) - \varphi(x; C)\|_{L_2}^2 = \min. \quad (1.10)$$

Тут  $\|f\|_{L_2}^2 = (f, f)$  — норма функції  $f(x)$ ;  $(f, g)$  — скалярний добуток функцій:

на відрізку (*інтегральний*) —

$$(f, g) = \int_a^b \rho(x) f(x) g(x) dx; \quad (1.11)$$

на скінченній множині точок  $x_i$  ( $i = \overline{1, N}$ ) (*дискретний*) —

$$(f, g) = \sum_{i=1}^N \rho_i f(x_i) g(x_i) \quad (1.12)$$

$(\rho(x) > 0$  — вагова функція;  $\rho_i > 0$  — вагові коефіцієнти (ваги), зміст яких пояснено далі).

### 1.2.1. Лінійна апроксимація

Спочатку розглянемо випадок, коли  $\varphi(x; C)$  має вигляд узагальненого полінома (1.2).

**Інтегральне середнє квадратичне наближення.** Підставивши (1.2) у (1.10), отримаємо

$$\begin{aligned} \min &= \left\| y(x) - \sum_{k=1}^n C_k \varphi_k(x) \right\|_{L_2}^2 = \\ &= \int_a^b \rho(x) \left[ y(x) - \sum_{k=1}^n C_k \varphi_k(x) \right]^2 dx \equiv I(C_1, \dots, C_n). \end{aligned}$$

Прирівнявши до нуля частинні похідні від  $I(C_1, \dots, C_n)$  за всіма складовими  $C_k$ , одержимо таку СЛІАР для їх визначення:

$$\sum_{k=1}^n a_{mk} C_k = b_m \quad (m = \overline{1, n}), \quad (1.13)$$

де

$$a_{mk} = (\varphi_m, \varphi_k), \quad b_m = (y, \varphi_m) \quad [(m, k = \overline{1, n})].$$

За умови лінійної незалежності  $\{\varphi_k\}$  визначник (1.13) ненульовий, тобто інтегральне середнє квадратичне наближення існує, до того ж воно єдине.

Відомо, що лінійно незалежну систему функцій можна ортогоналізувати. Тому якщо  $\{\varphi_k\}$  — ортонормована система, то має місце співвідношення  $(\varphi_m, \varphi_k) = \delta_{mk}$ . Тоді з (1.13) випливає, що

$$C_k = (y, \varphi_k) \equiv \int_a^b \rho(x) y(x) \varphi_k(x) dx \quad (k = \overline{1, n}).$$

Отже,  $C_k$  — коефіцієнти узагальненого ряду Фур'є, тобто інтегральне середнє квадратичне наближення є відрізок узагальненого ряду Фур'є.

Якщо  $\{\varphi_k\}$  — повна ортонормована система функцій, то  $\|y(x) - \varphi\|_{L_2}^2 = \sum_{k=1}^n C_k^2$  і за  $n \rightarrow \infty$  норма похибки прямує до нуля.

Якщо ж  $\varphi_k$  неортогональні, то за  $n \rightarrow \infty$  визначник (1.13) може прямувати до нуля і ця система стає *погано обумовленою (слабкостійкою)*, тобто розв'язок одержують із великою втра-тою точності. У цьому разі брати  $n$  більше за 5 – 6 недоцільно.

Отже, якщо в  $\varphi(x; C)$  потрібно мати більшу кількість чле-нів, слід попередньо ортогоналізувати  $\{\varphi_k\}$  або ж скористатись готовими системами ортогональних функцій, такими як орто-гональні поліноми із заданою вагою, наприклад поліномами Якобі або їхніми окремими випадками — поліномами Лежандра і Чебишова.

Зауважимо, що збіжність середнього квадратичного на-ближення тим краща, чим менше особливостей у функції  $y(x)$ . Тому якщо можна виділити особливості заданої функції  $y(x)$  у вигляді нескладної функції  $y_0(x)$  і далі наближати різницю  $y(x) - y_0(x)$ , точність апроксимації можна істотно покращити за невеликої кількості коефіцієнтів.

**Приклад 1.6.** Знайдемо інтегральне середнє квад-ратичне наближення функції  $y(x) = x^2$  на відрізку  $[0, 1]$  у вигляді

$$\varphi_{\text{СКН}}(x) = C_1 + C_2x.$$

Тут

$$\varphi_1(x) = 1, \varphi_2(x) = x \quad [\rho(x) \equiv 1].$$

Скориставшись виразами (1.13), одержимо коефіцієн-ти СЛАР

$$a_{11} = \int_0^1 1^2 dx = 1; \quad a_{12} = a_{21} = \int_0^1 1x dx = \frac{1}{2};$$

$$a_{22} = b_1 = \int_0^1 x^2 dx = \frac{1}{3}; \quad b_2 = \int_0^1 x^3 dx = \frac{1}{4}.$$

Отже, тут система (1.13) набуває вигляду

$$C_1 + \frac{C_2}{2} = \frac{1}{3}; \quad \frac{C_1}{2} + \frac{C_2}{3} = \frac{1}{4}.$$

Розв'язавши цю систему, одержимо  $C_1 = -\frac{1}{6}$ ,  $C_2 = 1$ ,

тобто

$$\varphi_{\text{ІСКН}}(x) = x - \frac{1}{6}.$$

**Дискретне середнє квадратичне наближення, або метод найменших квадратів.** У цьому разі умову (1.10) можна записати у такому вигляді:

$$I(C_1, \dots, C_n) \equiv \delta^2 \sum_{i=1}^N \rho_i = \sum_{i=1}^N \rho_i \left[ y(x_i) - \sum_{k=1}^n C_k \varphi_k(x_i) \right]^2 = \min.$$

Зазвичай тут має бути  $n < N$ , бо за  $n = N$  середнє квадратичне наближення переходить в інтерполяцію. Тут, як і в разі інтегрального середнє квадратичного наближення,  $C_k$  знаходять за рівнянням (1.13), тільки скалярний добуток визначають за формулою (1.12).

Якщо метод найменших квадратів застосовують для обробки експериментальних даних, то за вагу беруть  $\rho_i = \varepsilon_i^{-2}$ , тобто чим вища точність певних даних, тим більшої ваги набуває відповідна точка. Зрозуміло, що апроксимувальна крива  $\varphi(x)$  буде ближчою до точок, значення в яких знайдено точніше. Подібні міркування доцільні і для інтегрального середнього квадратичного наближення, тобто вибирають вагову функцію  $\rho(x) > 0$  більшою для тих значень аргументу, де потрібна вища локальна точність апроксимації ( $\varphi(x)$  ближча до  $y(x)$ ).

Кількість коефіцієнтів апроксимації  $n$  істотно залежить від вигляду функції  $y(x)$ , кількості вузлів  $N$ , їх розміщення, вага та вибраної системи функцій  $\{\varphi_k\}$ .

Оптимальне число коефіцієнтів  $n_{\text{опт}}$  визначають так. Вибирають будь-яке число  $n$ , знаходять  $C_k^{(n)}$  ( $k = \overline{1, n}$ ) та дискретне середнє квадратичне відхилення:

$$\delta_n = \sqrt{\frac{\sum_{i=1}^N \rho_i [y_i - \varphi(x_i; C^{(n)})]^2}{\sum_{i=1}^N \rho_i}}, \quad (1.14)$$

відтак порівнюють його із заданою або бажаною точністю  $\varepsilon$ . Якщо  $\delta_n \gg \varepsilon$ , то вибраного числа коефіцієнтів  $n$  замало для опису  $y(x)$  і  $n$  потрібно збільшити. Якщо  $\delta_n \ll \varepsilon$ , то коефіцієнти недостовірні й  $n$  слід зменшити. Якщо ж  $\delta_n \approx \varepsilon$ , то  $n$  — оптимальне.

На практиці розрахунки починають з  $n=1$ , коли напевне  $\delta_1 \gg \varepsilon$ , і збільшують число коефіцієнтів доти, доки не почне виконуватись умова  $\delta_n \approx \varepsilon$ . Якщо при цьому  $n \ll N$ , то  $\varphi(x)$  вибрано вдало, а якщо ж  $n_{\text{опт}} \approx N$ , то апроксимувальну функцію  $\varphi(x)$  потрібно взяти в іншому вигляді.

Зазначимо, що в разі інтегрального середнє квадратичного наближення середнє квадратичне відхилення знаходять за формулою

$$\delta = \sqrt{\frac{\int_a^b \rho(x) [y(x) - \varphi(x; C)]^2 dx}{\int_a^b \rho(x) dx}}. \quad (1.15)$$

Якщо взяти за  $\varphi_k(x) = x^{k-1}$  ( $k = \overline{1, n+1}$ ), то тут коефіцієнти СЛАР (1.13) визначають за такими формулами:

$$a_{mk} = \sum_{i=1}^N \rho_i x_i^{m+k-2}; \quad b_m = \sum_{i=1}^N \rho_i y_i x_i^{m-1} \quad [(m, k = \overline{1, n})]. \quad (1.16)$$

**Приклад 1.7.** Нехай функція  $y(x) = x^2$  задана таблично, тобто у вузлах  $x_i = (i-1)h$   $\left[ i = \overline{1, N}; h = \frac{1}{(N-1)} \right]$  на відрізку  $[0, 1]$ . Знаходимо дискретне середнє квадратичного наближення у вигляді лінійної функції

$$\varphi_{\text{ДСКРН}}(x) = C_1 + C_2x \quad (\rho_i \equiv 1).$$

У разі  $N = 4$  значення  $y(x)$  такі (табл. 1.5):

Таблиця 1.5

$i$	1	2	3	4
$x_i$	0	1/3	2/3	1
$y_i$	0	1/9	4/9	1

Скориставшись формулами (1.16), одержимо коефіцієнти системи (1.13):

$$a_{11} = N = 4;$$

$$a_{12} = a_{21} = \sum_{i=1}^4 x_i = 0 + \frac{1}{3} + \frac{2}{3} + 1 = 2;$$

$$a_{22} = \sum_{i=1}^4 x_i^2 = 0 + \frac{1}{9} + \frac{4}{9} + 1 = \frac{14}{9};$$

$$b_1 = \sum_{i=1}^4 y_i x_i^0 = \sum_{i=1}^4 x_i^2 = \frac{14}{9};$$

$$b_2 = \sum_{i=1}^4 y_i x_i = \sum_{i=1}^4 x_i^3 = 0 + \frac{1}{27} + \frac{8}{27} + 1 = \frac{4}{3}.$$

Отже, маємо СЛАР для знаходження коефіцієнтів  $C_1$  і  $C_2$ :

$$4C_1 + 2C_2 = \frac{14}{9}; \quad 2C_1 + \frac{14C_2}{9} = \frac{4}{3}.$$

Розв'язок цієї системи такий:

$$C_1 = -\frac{1}{9}; \quad C_2 = 1.$$

Отже,

$$\varphi_{\text{ДСКН}}(x) = x - 1/9.$$

Реалізацію розрахунків за методом найменших квадратів наведено у дод. 4.

Скориставшись формулою (1.14), знайдемо дискретне середнє квадратичне відхилення, попередньо обчислюючи значення знаменника та чисельника підкореневого

дробу в ньому  $\delta_{\text{ДСКН}} = \sqrt{\frac{A}{B}}$ :

$$B = \sum_{i=1}^4 \rho_i = 4; \quad A = \sum_{i=1}^4 \left[ x_i^2 - \left( x_i - \frac{1}{9} \right) \right]^2 = \left[ 0 - \left( 0 - \frac{1}{9} \right) \right]^2 + \\ + \left[ \frac{1}{9} - \left( \frac{1}{3} - \frac{1}{9} \right) \right]^2 + \left[ \frac{4}{9} - \left( \frac{2}{3} - \frac{1}{9} \right) \right]^2 + \left[ 1 - \left( 1 - \frac{1}{9} \right) \right]^2 = \frac{4}{81};$$

$$\delta_{\text{ДСКН}} = 1/9.$$

Якщо взяти  $N = 100$ , то значення вільних параметрів у  $\varphi(x)$  та дискретне середнє квадратичного відхилення будуть такі:

$$C_1 = -0,1650, \quad C_2 = 1; \quad \delta_{\text{ДСКН}} = 0,0760.$$

*Зауваження.* У разі аналітичного задання функції  $y(x) = x^2$  середнє квадратичне відхилення за апроксимації її на відрізок  $[0, 1]$  лінійною функцією

$$\varphi_{\text{СКН}}(x) = x - 1/6$$

знайдемо за формулою (1.15), як і раніше, попередньо обчисливши значення знаменника та чисельника підкореневого дробу в ньому:

$$A = \int_0^1 \rho(x) [y - \varphi(x)]^2 dx = \int_0^1 1 \times \left[ x^2 - \left( x - \frac{1}{6} \right) \right]^2 dx = \frac{1}{180};$$

$$B = \int_0^1 \rho(x) dx = \int_0^1 1 dx = 1; \quad \delta_{\text{ІСКН}} = \sqrt{\frac{A}{B}} = \sqrt{\frac{1}{180}} \approx 0,0745.$$

Із порівняння сталих  $C_1$ ,  $C_2$  і середніх квадратичних відхилень для інтегрального і дискретного середнє квадратичного наближень випливає, що зі збільшенням числа вузлів  $N$  у дискретному середнє квадратичному наближенні останнє прямує до інтегрального.

### 1.2.2. Нелінійна апроксимація

Лінійна залежність від  $C_k$ , якщо  $\varphi(x)$  має вигляд узагальненого алгебричного полінома, часто не відповідає характеру поведінки  $y(x)$ , бо алгебричний поліном високого степеня швидко зростає за  $|x| \rightarrow \infty$ . Оскільки середнє квадратичне наближення проводиться на великому інтервалі зміни аргументу, застосування тут нелінійної залежності  $\varphi(x)$  від коефіцієнтів  $C$  має бути ще доцільнішим, ніж у разі інтерполяції.

У випадку **квазілінійної залежності** підбирають такі вирівнювальні змінні  $\eta(y)$ ,  $\xi(x)$ , щоб графік вихідної функції  $y(x)$  у нових змінних  $\eta(\xi)$  був близьким до прямої, що уможливить використання результатів п. 1.2.1 для визначення вектора вільних параметрів  $C$ . Щоправда, у нових змінних слід звертати особливу увагу на вибір ваги для одержання прийнятної точності апроксимації у первинних змінних.

Найчастіше застосовують **дробово-лінійну (раціональну) залежність (апроксимація Паде)**  $\varphi(x)$  від вільних параметрів

$$y(x) \approx \varphi(x) = P_n(x) / Q_m(x) = \frac{\sum_{k=0}^n \alpha_k x^k}{\sum_{q=0}^m \beta_q x^q} \quad (\beta_0 = 1). \quad (1.17)$$

Крім того, іноді застосовують і відношення узагальнених поліномів.

Апроксимація Паде дає змогу передати полюси заданої функції  $y(x)$  (ім відповідатимуть нулі знаменника  $Q_m(x)$ ) потріб-



ної кратності), а іноді й відтворити асимптотичну поведінку  $y(x)$  у разі  $x \rightarrow \infty$  за рахунок відповідного вибору різниці  $n - m$  (якщо  $y(\infty) = \text{const} \neq 0$ , то беруть  $n = m$ ).

Зрозуміло, що тут уже квадрат похибки

$$\left\| y(x) - \frac{P_n}{Q_m} \right\|_{L_2}^2$$

не буде квадратичною формою відносно невідомих коефіцієнтів, тому знайти їх нелегко. Якщо ж його записати у вигляді

$$\| Q_m(x)y(x) - P_n(x) \|_{L_2}^2 = \min,$$

то зліва уже матимемо квадратичну форму коефіцієнтів  $\alpha_k, \beta_q$ .

Отже, перепишемо вихідну задачу в іншому вигляді:

$$\left\| y(x) - \frac{P_n}{Q_m} \right\|_{L_2}^2 = \int_a^b \tilde{\rho}(x) [Q_m(x)y(x) - P_n(x)]^2 dx = \min,$$

$$\tilde{\rho}(x) = \frac{\rho(x)}{Q_m^2(x)}.$$

Щоправда, тепер у цій задачі замість ваги  $\rho(x)$  скористалися вагою  $\tilde{\rho}(x)$

Визначимо коефіцієнти (вільні параметри), удаючись до такого ітераційного процесу:

$$\tilde{\rho}^{(s)}(x) = \frac{\rho(x)}{[Q_m^{(s-1)}(x)]^2};$$

$$\int_a^b \tilde{\rho}^{(s)}(x) [Q_m^{(s)}(x)y(x) - P_n^{(s)}(x)]^2 dx = \min,$$

$$[s = 1, 2, \dots; Q_m^{(0)}(x) \equiv 1].$$

На кожній ітерації вага  $\tilde{\rho}(x)$  відома і задача легко розв'язується, бо при цьому коефіцієнти слабо залежать від вибору ваги й ітерації швидко збігаються.

Очевидно, тоді  $\alpha_k$  ( $k = \overline{0, n}$ ),  $\beta_q$  ( $q = \overline{1, m}$ ) можна знайти як результат розв'язання такої СЛАР ( $M = n + m$ ):

$$\sum_{j=0}^M a_{ij}^{(s)} C_j^{(s)} = b_i^{(s)} \quad (i = \overline{0, M}; s = 1, 2, \dots). \quad (1.18)$$

Тут позначено:

$$a_{ij}^{(s)} = (\varphi_i, \varphi_j)^{(s)}; \quad b_i^{(s)} = (y, \varphi_i)^{(s)};$$

$$(f, g)^{(s)} = \int_a^b \tilde{\rho}^{(s)}(x) f(x) g(x) dx;$$

$$\tilde{\rho}^{(s)}(x) = \frac{\rho(x)}{[Q_m^{(s-1)}(x)]^2} \quad [Q_m^{(0)}(x) \equiv 1];$$

$$C_j^{(s)} = \alpha_j^{(s)}, \quad \varphi_j(x) = x^j \quad (j = \overline{0, n});$$

$$C_j^{(s)} = -\beta_{j-n}^{(s)}, \quad \varphi_j(x) = y(x) x^{j-n} \quad (j = \overline{n+1, M}).$$

Розрахунки уможливають виконання умови

$$|C_j^{(s+1)} - C_j^{(s)}| \leq \varepsilon \quad (j = \overline{0, M}).$$

Цей метод має назву методу *ітерованої ваги*.

Зазначимо, що за допомогою апроксимації Паде можна задовільно апроксимувати навіть функції з розривами  $[y(x) \equiv |x|]$ .

**Приклад 1.8.** Знайдемо апроксимацію Паде для функції  $y(x) = \ln(1+x)$  на відрізку  $[-0,5; 1]$  у вигляді

$$\Phi_{\text{Паде}}(x) = \frac{\alpha x}{1 + \beta x}.$$

Тут формули (1.18) за  $\rho(x) \equiv 1$ ,  $M = 2$  набувають вигляду

$$\sum_{j=1}^M a_{ij}^{(s)} C_j^{(s)} = b_i^{(s)} \quad (i = \overline{1, M}; s = 1, 2, \dots),$$

де

$$a_{ij}^{(s)} = (\varphi_i, \varphi_j)^{(s)}, \quad b_i^{(s)} = (y, \varphi_i)^{(s)} \quad (i = \overline{1, M});$$

$$(f, g)^{(s)} = \int_{-0.5}^1 \tilde{\rho}^{(s)}(x) f(x) g(x) dx;$$

$$\tilde{\rho}^{(s)}(x) = \frac{1}{[1 - C_2^{(s-1)} x]^2} [Q_m^{(0)}(x) \equiv 1];$$

$$C_1^{(s)} = \alpha^{(s)}; \quad \varphi_1(x) = x;$$

$$C_2^{(s)} = -\beta^{(s)}; \quad \varphi_2(x) = x \cdot y(x) = x \cdot \ln(1 + x).$$

Якщо інтеграли в цих виразах обчислити за узагальненою формулою середніх (3.4) за  $N = 100$ , то для  $s = 1$  СЛАР матиме такий вигляд:

$$0.3750C_1^{(1)} + 0.1643C_2^{(1)} = 0.3025;$$

$$0.1643C_1^{(1)} + 0.1171C_2^{(1)} = 0.1105,$$

звідки

$$C_1^{(1)} = 1.0209; \quad C_2^{(1)} = -0.4886.$$

Якщо сталі  $C_1^{(s)}$  і  $C_2^{(s)}$  знаходити з похибкою  $\varepsilon = 10^{-5}$ , то після чотирьох ітерацій одержимо такі значення:

$$C_1^{(4)} = 1.0230; \quad C_2^{(4)} = -0.4950.$$

Отже,

$$\alpha \approx 1.0230; \quad \beta \approx 0.4950$$

і апроксимація Паде має такий вигляд:

$$\varphi_{\text{Падє}}(x) = \frac{1,0230x}{1 + 0,4950x}.$$

Із порівняння наближеного

$$\varphi_{\text{Падє}}(0,5) = 0,4100$$

і точного

$$\ln(1,5) = 0,4055$$

значень заданої функції в точці  $x = 0,5$  впливає, що похибка апроксимації становить 1,12 %. Якщо ж покласти  $\ln(1+x) \approx x/(1+0,5x)$ , то похибка обчислення у тій самій точці приблизно дорівнюватиме 5 %.

Реалізацію розрахунків за (1.18) наведено в дод. 5.

### 1.3. РІВНОМІРНЕ НАБЛИЖЕННЯ

У рівномірному наближенні вектор вільних параметрів  $C$  знаходять з такої умови:

$$\|y(x) - \varphi(x; C)\|_c \equiv \max_{[a,b]} |y(x) - \varphi(x; C)| = \min, \quad (1.19)$$

де мінімум відшуковують на множині функцій  $\varphi(x)$ .

Слід зазначити, що для досить гладких функцій  $y(x)$  середнє квадратичне наближення забезпечує таку саму точність, що й рівномірне наближення і лише для не досить гладких  $y(x)$  середнє квадратичне наближення збігається повільно (або й взагалі не збігається), але й рівномірне наближення тоді також збігається повільно, а отже, застосовувати його значно важче.

*На практиці* в основному застосовують середнє квадратичне наближення, але для поліпшення його збіжності виділяють особливості у вигляді  $y_0(x)$  і апроксимують різницю  $y(x) - y_0(x)$ . Найчастіше рівномірне наближення використовують для обчислення вбудованих функцій на ЕОМ, де потрібна висока точність за невеликого числа членів суми.

Задачу знаходження коефіцієнтів рівномірного наближення (1.19) не можна розв'язати за скінченне число дій, але її можна розв'язати за допомогою ітераційного процесу, якщо чебишовську норму  $\|\cdot\|_c$  розглядати як межу  $\|\cdot\|_{L_p}$  за умов  $p \rightarrow \infty$  та  $\rho(x) \equiv 1$ .

Отже, задачу знаходження коефіцієнтів рівномірного наближення

$$\|y(x) - \varphi(x; C)\|_{L_p}^p = \int_a^b |y(x) - \varphi(x; C)|^p dx = \min$$

у просторі  $L_p$  розв'язують *ітеруванням ваги*:

$$\int_a^b \rho^{(s)}(x) [y(x) - \varphi^{(s+1)}(x; C)]^2 dx = \min;$$

$$\rho^{(s)}(x) = |y(x) - \varphi^{(s)}(x; C)|^{p-2}$$

$$(s = 0, 1, 2, \dots; p = 2, 3, \dots).$$

Тоді на кожній ітерації матимемо задачу на знаходження середнє квадратичного наближення, тобто тут виникає подвійний межовий перехід, що потребує громіздких числових розрахунків. Для їх спрощення межові процеси можна об'єднати в один, якщо на першій ітерації по  $s$  покласти  $p = 2$ , на другій —  $p = 4$ , на третій —  $p = 6$  і т.д. Це, у свою чергу, зумовлює таку задачу:

$$\int_a^b \rho^{(s)}(x) [y(x) - \varphi^{(s+1)}(x)]^2 dx = \min, \quad (1.20)$$

де  $\rho^{(s)}(x) = [y(x) - \varphi^{(s)}(x)]^{2s}$  ( $\rho^{(0)}(x) \equiv 1$ ;  $s = 0, 1, 2, \dots$ ).

Якщо апроксимувальну функцію  $\varphi(x;C)$  взяти у вигляді узга-  
гального полінома (1.2) і підставити її в (1.20), одержимо та-  
ку СЛАР для визначення  $C_k$  :

$$\sum_{k=1}^n a_{mk}^{(s)} C_k^{(s+1)} = b_m^{(s)} \quad (m = \overline{1, n}), \quad (1.21)$$

де

$$a_{mk}^{(s)} = (\varphi_m \cdot \varphi_k)^{(s)}; \quad b_m^{(s)} = (y, \varphi_m)^{(s)};$$

$$(f, g)^{(s)} = \int_a^b \rho^{(s)}(x) f(x) g(x) dx;$$

$$\rho^{(s)}(x) = \left[ y(x) - \sum_{k=1}^n C_k^{(s)} \varphi_k(x;C) \right]^{2s}$$

$$(\rho^{(0)}(x) \equiv 1; \quad s = 0, 1, 2, \dots).$$

**Приклад 1.9.** Знайдемо рівномірне наближення  
функції  $y(x) = x^2$  на відрізку  $[0,1]$  у такому вигляді:

$$\varphi_{PH}(x) = C_1 + C_2 x.$$

Тут  $\varphi_1(x) = 1; \quad \varphi_2(x) = x.$

Оскільки на першій ітерації ( $s = 0$ ) вагова функція  
має вигляд  $\rho^{(0)}(x) \equiv 1$ , то знайдемо інтегральне середнє  
квадратичне наближення (див. прикл. 1.7), тобто

$$\varphi_{PH}^{(1)}(x) = x - 1/6.$$

На другій ітерації ( $s = 1$ ) коефіцієнти знайдемо з та-  
ких виразів:

$$a_{mk}^{(1)} = \int_0^1 \rho^{(1)}(x) x^{m+k-2} dx;$$

$$b_m^{(1)} = \int_0^1 \rho^{(1)}(x) x^2 x^{m-1} dx [m.k = 1,2];$$

$$\rho^{(1)}(x) = [x^2 - (x - 1/6)]^2.$$

Провівши всі необхідні розрахунки, матимемо такі вирази для коефіцієнтів СЛАР

$$a_{11}^{(1)} = \frac{1}{180}; \quad a_{12}^{(1)} = a_{21}^{(1)} = \frac{1}{360}; \quad a_{22}^{(1)} = \frac{8}{21 \times 180};$$

$$b_1^{(1)} = \frac{8}{21 \times 180}; \quad b_2^{(1)} = \frac{9}{28 \times 180}.$$

Помноживши всі коефіцієнти на 180, остаточно прийдемо до такої СЛАР:

$$C_1^{(2)} + \frac{C_2^{(2)}}{2} = \frac{8}{21}; \quad \frac{C_1^{(2)}}{2} + 8 \frac{C_2^{(2)}}{21} = \frac{9}{28},$$

розв'язок якої має вигляд

$$C_1^{(2)} = -5/42; \quad C_2^{(2)} = 1.$$

Отже,

$$\varphi_{PH}^{(2)}(x) = x - 5/42 = x - 0,1190,$$

а точний розв'язок

$$\varphi_{PH}(x) = x - 1/8 = x - 0,125$$

тобто похибка визначення  $C_1^{(2)}$  менша за 5 %.

Реалізацію розрахунків за (1.21) наведено в дод. 6.



**Задачі для самостійного розв'язування**

1. Скориставшись заданими значеннями функції  $y(x) = 2^x$ :

$$y(-1) = 1/2, y(0) = 1, y(1) = 2, y(2) = 4,$$

побудуйте інтерполяційний поліном Ньютона, записавши його у вигляді алгебричного полінома 3-го степеня, вільні кубічні сплайни та знайдіть наближене значення в точці  $x = 1/2$  обома способами.

2. Визначте апостеріорну й апостеріорну похибки інтерполяції для вправи 1.

3. Розглянувши інтерполяцію квадратичними сплайнами, запропонуйте економний спосіб визначення їхніх коефіцієнтів.

4. За даними вправи 1 знайдіть лінійний поліном, коефіцієнти якого визначте методом найменших квадратів.

5. Побудуйте лінійний поліном вигляду  $\varphi(x) = C_1 + C_2x$  для апроксимації функції  $y(x) = x^{1/2}$  на відріжку  $[0, 1]$ , коефіцієнти якого визначте за допомогою інтегрального середнього квадратичного та рівномірного наближення.



## Розділ 2

# ЧИСЛОВЕ ДИФЕРЕНЦІЮВАННЯ

---

Якщо похідні функції від  $y(x)$  важко або неможливо знайти аналітично (наприклад, функція задана таблично), її наближають легкообчислюваною апроксимувальною функцією  $\varphi(x)$  і далі покладають

$$y^{(k)}(x) \approx \varphi^{(k)}(x).$$

**Приклад 2.1.** Знайдемо першу похідну функції  $y(x) = \ln(1+x)$ , скориставшись апроксимацією Паде цієї функції з прикл. 1.7:

$$\varphi_{\text{Паде}}(x) = \frac{\alpha x}{1 + \beta x} \quad (\alpha = 1,0230; \beta = 0,4950).$$

Очевидно, тоді

$$\varphi'_{\text{Паде}}(x) = \frac{\alpha}{(1 + \beta x)^2}.$$

Обчисливши значення цієї похідної в точці  $x = 0,4$  за наближеною і точною формулою

$$y'(x) = \frac{1}{1+x},$$

матимемо

$$\varphi'_{\text{Паде}}(0,4) = 0,7128 \text{ і } y'(0,4) = 0,7143.$$

Знайдене наближене значення похідної різниться від точного на 0,2 %.

### 2.1. Поліноміальні формули

---

Якщо функція  $y(x)$  задана таблично, за  $\varphi(x)$  можна взяти інтерполяційний поліном Ньютона, тобто покласти

$$y(x) \approx \varphi(x) = y(x_0) + \sum_{m=1}^n y'(x_0, x_1, \dots, x_m) \prod_{i=0}^{m-1} (x - x_i),$$

і, замінивши в ньому  $x - x_i$  на  $\xi_i$ , одержати вирази для похідних

$$\begin{aligned}\varphi(x) &= y(x_0) + \xi_0 y(x_0, x_1) + \xi_0 \xi_1 y(x_0, x_1, x_2) + \dots; \\ \varphi'(x) &= y(x_0, x_1) + (\xi_0 + \xi_1) y(x_0, x_1, x_2) + \dots; \\ \varphi''(x) &= 2y(x_0, x_1, x_2) + \dots\end{aligned}$$

Загальна формула матиме такий вигляд:

$$\varphi^{(k)}(x) = k! \left[ y(x_0, \dots, x_k) + \left( \sum_{i=0}^k \xi_i \right) y(x_0, \dots, x_{k+1}) + \dots \right]. \quad (2.1)$$

Залишаючи в сумі (2.1) певне число доданків, одержимо наближені вирази для похідних. Якщо ж залишити один член, то

$$\begin{aligned}y'(x) \approx y(x_0, x_1) &= \frac{y(x_0) - y(x_1)}{x_0 - x_1}; \dots; \\ \frac{1}{k!} y^{(k)}(x) \approx y(x_0, \dots, x_k) &= \sum_{p=0}^k y_p \prod_{\substack{i=0 \\ i \neq p}}^k (x_p - x_i)^{-1}.\end{aligned} \quad (2.2)$$

Ці формули розраховані на *нерівномірну сітку*.

Оцінимо похибку числового диференціювання. Якщо відстані між вузлами малі, то похибка приблизно дорівнює першому відкинутому члену й у разі використання вузлів  $x_i$  ( $i = \overline{0, n}$ ) до цього члена входить поділена різниця  $y(x_0, x_1, \dots, x_{n+1})$ , яка згідно з (2.2) приблизно дорівнює  $\frac{y^{(n+1)}(x)}{(n+1)!}$ . Перед нею стоїть сума добутоків різних множників

$\xi_i$ ; кожен добуток має  $n+1-k$  множників, а сума —  $C_{n+1}^k$  доданків. Звідси випливає оцінка похибки (2.1) з  $n+1$  вузлом:

$$\left| r_n^{(k)}(x) \right| \leq \max \left| y^{(n+1)} \right| \max_i \left| \xi_i \right|^{(n+1-k)} / [(n+1-k)!]. \quad (2.3)$$

Якщо ж сітка рівномірна ( $\max |\xi_i| = h = \text{const}$ ), то (2.3) можна записати так:

$$\left| r_n^{(k)}(x) \right| < \max \left| y^{(n+1)} \right| \left[ \frac{enh}{n+1-k} \right]^{n+1-k} \approx O(h^{n+1-k}). \quad (2.4)$$

Отже, порядок точності формули (2.1) відносно кроку сітки дорівнює числу залишених у ній членів. Тому *мінімальне число вузлів для обчислення  $k$ -ї похідної дорівнює  $k+1$ , приводить до формул (2.2) і забезпечує перший порядок точності.*

Якщо скористатись 4 – 6 вузлами, то навіть для добре складеної таблиці на докладній сітці за допомогою числового диференціювання з прийнятною точністю можна визначити першу та другу похідні, а третю та четверту — лише задовільною.

Зауважимо, що вільні кубічні сплайни забезпечують визначення першої та другої неперервних похідних, причому з доброю точністю. Якщо ж табульовані і функція, і її перша похідна, то краще будувати і диференціювати інтерполяційний поліном Ерміта.

## 2.2. Симетричні формули

З урахуванням того що на практиці найчастіше використовують рівномірні сітки, формули (2.1) можна істотно спростити, а точність підвищити, якщо скористатись для розрахунків симетрично розміщеними вузлами. Наприклад, візьмемо три сусідні вузли  $x_{i-1}$ ,  $x_i$ ,  $x_{i+1}$  і знайдемо першу та другу похідні в середньому вузлі.

Виразивши у формулах (2.2) поділені різниці через вузлові значення функції, одержимо такі співвідношення:

$$\begin{aligned} y'(x_i) &= \frac{y_{i+1} - y_{i-1}}{2h} + O(h^2); \\ y''(x_i) &= \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + O(h^2). \end{aligned} \quad (2.5)$$

Для одержання *априорної* оцінки точності цих формул скористаємось розкладанням функції  $y(x)$  у ряд Тейлора, вважаючи, що  $y(x)$  має неперервну четверту похідну. Для цього виразимо значення функції у вузлах  $x_{i\pm 1}$  через значення функції та її похідних у вузлі  $x_i$ :

$$y(x_{i\pm 1}) = y(x_i \pm h) = y_i \pm hy'_i + \frac{h^2 y''_i}{2} \pm \frac{h^3 y'''_i}{6} + \frac{h^4 y^{IV}(\eta_{\pm})}{24}$$

$$[y(x_i) = y_i],$$

де  $\eta_-$  — деяка точка інтервалу  $(x_{i-1}, x_i)$ ;  $\eta_+$  — деяка точка інтервалу  $(x_i, x_{i+1})$ .

Підставивши ряд Тейлора у вираз другої похідної в (2.5), одержимо

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = y''_i + \frac{h^2 [y^{IV}(\eta_+) + y^{IV}(\eta_-)]}{24} \approx y''_i + O(h^2), \quad (2.6)$$

що підтверджує зроблену раніше оцінку й уточнює величину залишкового члена, який дорівнює  $\frac{h^2 y^{IV}(\eta)}{12}$ .

### 2.3. Правило Рунге

З виразу (2.6) випливає, що похибка формул (2.5) для чотирьох рази диференційованої функції має вигляд

$$r = h^2 \psi(\eta),$$

де  $\eta$  — якась точка поблизу вузла  $x_i$ .

Якщо  $y^{IV}(x)$  ліпшицово неперервна, то оцінку неважко уточнити:

$$r = h^2 \psi(\eta) + O(h^3).$$

Оцінимо похибку та уточнення наближених значень функції  $Z(x)$ , якщо воно пов'язане з наближеним  $\zeta(x;h)$ , знайденим на рівномірній сітці з кроком  $h$ , співвідношенням

$$Z(x) = \zeta(x;h) + r(x). \quad (A)$$

Нехай похибка обчислення функції визначається формулою

$$r(x) = Z(x) - \zeta(x;h) = h^p \psi(x) + O(h^{p+1}). \quad (B)$$

Провівши розрахунки за цією наближеною формулою в тій самій точці  $x$ , але на сітці з кроком  $gh$ , одержимо значення

$\zeta(x;gh)$ , пов'язане з точним таким співвідношенням:

$$Z(x) - \zeta(x;gh) = (gh)^p \psi(x) + O((gh)^{p+1}). \quad (C)$$

Зрозуміло, що  $O((gh)^{p+1}) \approx O(h^{p+1})$ . Маючи два розрахунки на різних сітках, знайдемо оцінку похибки. Для цього від (C) віднімемо (B):

$$r(x) \approx h^p \psi(x) = \frac{\zeta(x;h) - \zeta(x;gh)}{g^p - 1} + O(h^{p+1}). \quad (2.7)$$

Це так звана *перша формула Рунге*.

Отже, розрахунок на другій сітці дає змогу оцінити похибку розрахунку на першій сітці. Якщо знайдену похибку виключити з (A), одержимо точніше значення шуканої функції за *другою формулою Рунге*:

$$\tilde{Z}(x) \approx \zeta(x;h) + \frac{\zeta(x;h) - \zeta(x;gh)}{g^p - 1}. \quad (2.8)$$

У разі числового диференціювання знайдене з кроком  $h$  значення  $k$ -ї похідної можна уточнити за формулою (2.8):

$$\tilde{y}^{(k)}(x) \approx \varphi^{(k)}(x;h) + \frac{\varphi^{(k)}(x;h) - \varphi^{(k)}(x;gh)}{g^p - 1}. \quad (2.9)$$

Тут для (2.5)  $p = 2$ .

**Приклад 2.2.** Нехай функція  $y(x) = \ln(1+x)$  задана таблично (табл. 2.1):

Таблиця 2.1

$x$	0,0000	0,2000	0,4000	0,6000	0,8000
$y$	0,0000	0,1823	0,3365	0,4700	0,5878

і потрібно обчислити  $y'(0,4)$ . Skorистаємось першою з формул (2.5), взявши спочатку крок 0,2:

$$y'(0,4;0,2) \approx \frac{0,4700 - 0,1823}{2 \times 0,2} = 0,7193.$$

Потім збільшимо крок удвічі і візьмемо точки  $x = 0,0$  і  $x = 0,8$ :

$$y'(0,4;0,4) \approx \frac{0,5878 - 0,0000}{2 \times 0,4} = 0,7348.$$

Тепер уточнимо значення  $y'(0,4)$  за формулою (2.9) (нагадаємо, що тут  $p = 2$ ):

$$\tilde{y}'(0,4) \approx 0,7193 + \frac{(0,7193 - 0,7348)}{2^2 - 1} = 0,7140.$$

Порівнявши це значення з точним  $y'(0,4) = 0,7143$ , бачимо, що  $\delta = [|y'(0,4) - \tilde{y}'(0,4)| / y'(0,4)] \cdot 100 = 0,04 \%$ .

Реалізацію цих розрахунків наведено в дод. 7.

## 2.4. Квазірівномірні сітки

Якщо функція за деяких значень аргументу *різко змінюється*, то для одержання задовільних результатів під час розрахунків (наприклад, у разі числового диференціювання) крок сітки потрібно зменшувати. Водночас на відрізках аргументу, де функція змінюється повільно, навпаки, крок можна взяти більшим. Тому вдало підібрана нерівномірною сітка може істотно полегшити розв'язання задачі. До нерівномірних сіток вдаються також під час числового інтегрування та розв'язання диференціальних і інтегральних рівнянь.

Важливу роль тут відіграють так звані *квазірівномірні сітки*. Для їх побудови беруть двічі неперервно диференційовані функції  $x = \xi(t)$ , які перетворюють відрізок  $0 \leq t \leq 1$  на відрізок  $a \leq x \leq b$  так, щоб кожній сітці  $x_i^{(N)}$  відповідала рівномірною сітка  $t_i^{(N)} = \frac{i}{N}$  ( $i = \overline{0, N}$ ), причому на цьому відрізку  $\xi'(t) \geq \varepsilon > 0$ , а  $\xi''(t)$  — обмежена. Так, наприклад, перетворення  $x = \alpha t \operatorname{tg} \frac{\pi t}{2}$  за  $0 \leq t < 1$  дає змогу побудувати квазірівномірні сітки на півпрямій (параметр  $\alpha$  керує сіткою; чим він менший, тим щільніша сітка за  $x \rightarrow 0$ ; останній інтервал цієї сітки нескінченний).

Для скупчення квазірівномірної сітки  $x$  слід скупчити рівномірну сітку  $t_i$  (збільшити  $N$ ) і по ній обчислити нову сітку.

Середину інтервалу на квазірівномірній сітці визначають так:  $x_{i+1/2} = \xi \left[ \left( i + \frac{1}{2} \right) / N \right]$ . Похідні на квазірівномірній сітці обчислюються або простіше, або точніше, ніж на довільній нерівномірній сітці. Для підвищення точності і на квазірівномірній сітці можна користуватись правилом Рунге.

Квазірівномірні сітки часто використовують для детальнішого дослідження особливостей розв'язання за допомогою невеликого числа вузлів.

## **2.5. Числове диференціювання швидкозмінних функцій**

---

Для числового диференціювання *швидкозмінних* функцій доцільна квазілінійна інтерполяція, її виконують за допомогою нескладних вирівнювальних змінних  $\xi(x)$ ,  $\eta(y)$ , похідні яких  $\xi'_x(x)$ ,  $\eta'_y(y)$  знаходять точно. Очевидно, що для шуканої першої похідної має місце співвідношення

$$y'_x = \eta'_\xi \frac{\xi'_x}{\eta'_y}.$$

Тут числово знаходять лише похідну  $\eta'_\xi$ .

Якщо у вихідних змінних  $x$ ,  $y$  сітка була рівномірною або квазірівномірною, то зазвичай вона квазірівномірна й у вирівнювальних змінних, бо вирівнювальне перетворення на обмеженому відрізку майже завжди має потрібні властивості похідних. У цьому разі результат можна уточнювати за правилом Рунге.

Другу і старші похідні швидкозмінних функцій важко знаходити числово, оскільки дворазове диференціювання інтерполяційного полінома Ньютона ненадійне.

## **2.6. Регуляризація числового диференціювання**

---

Похибка числового диференціювання складається з похибки апроксимації  $|r^{(k)}|$ , тобто залишкового члена формул [(2.3)

або (2.4)] (похибка методу), і похибки округлень у разі розрахунків на ЕОМ  $|\rho^{(k)}|$  (правильніше сказати мажоранти цієї неусувної похибки). Якщо  $|r^{(k)}|$  зі зменшенням кроку сітки  $h$  для досить гладких функцій зменшується, то похибка округлень  $|\rho^{(k)}|$  за дуже малих  $h$  збільшується. Тому сумарна похибка числового диференціювання  $|r^{(k)}| + |\rho^{(k)}|$  може зменшуватися зі зменшенням  $h$  лише до деякого межового значення, після чого подальше зменшення  $h$  не підвищуватиме точності результатів.

Викладений спосіб визначення оптимального кроку  $h_{\text{опт}}$  є деякий спосіб регуляризації числового диференціювання (так звана *регуляризація за кроком*). Інший спосіб — диференціювання попередньо згладженої кривої, причому згладжування виконується методом найменших квадратів.



### Задачі для самостійного розв'язування

1. Застосувавши поліноміальну формулу (2.1), обчисліть першу та другу похідні функції  $y(x)$  в точці  $x = 60$  з першим і другим порядком точності, якщо відомі такі її значення:

$$y(50) = 1,6990; \quad y(55) = 1,7404; \quad y(60) = 1,7782;$$

$$y(65) = 1,8129; \quad y(70) = 1,8448.$$

2. Скориставшись симетричними формулами (2.5), обчисліть значення першої та другої похідних у точці  $x = 60$  для функції  $y(x)$  з попередньої вправи, уточнивши їх за правилом Рунге.

3. Для функції  $y(x)$  з першої вправи за допомогою методу найменших квадратів знайдіть коефіцієнти апроксимувальної функції  $\varphi(x) = a + bx + cx^2$  та обчисліть за нею значення першої та другої похідних у точці  $x = 60$ .



## Розділ 3

### ЧИСЛОВЕ ІНТЕГРУВАННЯ

У практичних розрахунках нерідко виникає потреба в обчисленні визначених інтегралів:

$$F = \int_a^b \rho(x)f(x)dx \quad (\rho(x) > 0), \quad (3.1)$$

де функція  $f(x)$  неперервна на відрізку  $[a, b]$ ; вагова функція  $\rho(x)$  неперервна на інтервалі  $(a, b)$ .

До числового інтегрування вдаються, коли інтеграл неможливо виразити через елементарні функції або ж коли функцію  $f(x)$  задано таблично, а також коли внаслідок інтегрування одержано незручний для користування вираз. Тоді  $f(x)$  наближають такою апроксимувальною функцією  $\varphi(x; C)$ , інтеграл від якої беруть в елементарних функціях.

Найчастіше підінтегральну функцію  $f(x)$  замінюють деяким узагальненим поліномом. Тоді внаслідок лінійності такої апроксимації відносно параметрів  $C_i$  функцію  $f(x)$  можна записати так:

$$f(x) = \sum_{i=0}^n f(x_i)\varphi_i(x) + r(x), \quad (3.2)$$

де  $r(x)$  — залишковий член апроксимації.

Підставляючи вираз (3.2) у формулу (3.1), одержимо загальну формулу числового інтегрування — квадратурну формулу:

$$F = \sum_{i=0}^n C_i f(x_i) + R;$$
$$C_i = \int_a^b \rho(x)\varphi_i(x)dx; \quad R = \int_a^b \rho(x)r(x)dx. \quad (3.3)$$

де  $x_i$  — вузли;  $C_i$  — ваги;  $R$  — похибка або залишковий член квадратурної формули.

Отже, інтеграл наближено замінено на суму, подібну до інтегральної, причому як вузли, так і коефіцієнти (ваги) квадратурної формули не залежать від функції  $f(x)$ . Зазвичай інтерполяційний поліном (3.2) не обов'язково має бути *лагранжовим*.

### 3.1. Поліноміальні формули

Розглянемо найліпше вивчену заміну підінтегральної функції  $f(x)$  — заміну на алгебричний поліном, причому коли  $\rho(x) \equiv 1$ .

#### 3.1.1. Формула середніх

Якщо на відрізку  $[a, b]$  взяти єдиний вузол квадратурної формули  $x_0$ , то підінтегральна функція  $f(x)$  апроксимується поліномом нульового степеня — сталою  $f(x_0)$ . Ураховуючи, що симетричне розміщення вузлів у числовому диференціюванні привело до підвищення точності, за вузол  $x_0$  візьмемо середню відрізка інтегрування  $\bar{x} = \frac{a+b}{2}$ . Замінивши наближено площу криволінійної трапеції на площу прямокутника з висотою  $f(\bar{x})$  та основою  $(b-a)$ , одержимо *формулу середніх*:

$$F = \int_a^b f(x) dx \approx f(\bar{x})(b-a) \quad (\bar{x} = \frac{a+b}{2}).$$

Це найпростіша квадратурна формула. Розклавши  $f(x)$  у ряд Тейлора в околі точки  $\bar{x}$ ,

$$f(x) = f(\bar{x}) + (x - \bar{x})f'(\bar{x}) + \frac{(x - \bar{x})^2 f''(\bar{x})}{2} + \dots$$

і підставивши цей відрізок ряду в інтеграл, одержимо значення похибки формули середніх:

$$R = \int_a^b f(x)dx - f(\bar{x})(b-a) \approx \frac{(b-a)^3 f''(\bar{x})}{24}.$$

Очевидно, формула середніх є точна для лінійної підінтегральної функції  $f(x)$ , бо тоді  $f''(x) = 0$ .

Природно, що точність формули середніх для довільної  $f(x)$  можна підвищити, скориставшись докладнішою сіткою  $x_i$  ( $i = 0, N$ ):

$$F \approx \sum_{i=1}^N (x_i - x_{i-1}) f(\bar{x}_i);$$

$$R \approx \frac{1}{24} \sum_{i=1}^N (x_i - x_{i-1})^3 f''(\bar{x}_i) \left( \bar{x}_i = \frac{x_{i-1} + x_i}{2} \right).$$

Це так звана *узагальнена формула середніх*.

У разі рівномірної сітки  $x_i - x_{i-1} = h = (b-a) / N$  ці формули набудуть такого вигляду:

$$F_h^C \approx h \sum_{i=1}^N f_{i-1/2}; \quad f_{i-1/2} = f\left(x_i - \frac{h}{2}\right); \quad (3.4)$$

$$R \approx \frac{h^3}{24} \sum_{i=1}^N f''(\bar{x}_i) \approx \frac{h^2}{24} \int_a^b f''(x) dx \approx O(h^2). \quad (3.5)$$

Наведені оцінки  $R$  справедливі, якщо існує неперервна  $f''(x)$ ; якщо ж вона кусково-неперервна, має місце лише мажорантна оцінка:

$$R \leq \frac{(b-a)h^2 M_2}{24} \left( M_2 = \max_{[a,b]} |f''(x)| \right).$$

### 3.1.2. Формула трапецій

Замінімо функцію  $f(x)$  на відрізку  $[a, b]$  інтерполяційним поліномом першого степеня з вузлами  $x = a$ ,  $x = b$ , що відповідає заміні кривої  $f(x)$  січною. Тоді значення шуканого інтеграла — площу криволінійної трапеції — можна наближено

замінити площею трапеції з висотою  $(b - a)$  та основами  $f(a)$  і  $f(b)$ . Отже, формула трапецій матиме вигляд

$$F = \int_a^b f(x)dx \approx (b - a) \frac{f(a) + f(b)}{2}.$$

Залишковий член формули трапецій відрізняється від залишкового члена формули середніх числовим коефіцієнтом, тобто замість  $(1/24)$  буде  $(-1/12)$ .

На докладнішій сітці  $x_i$  ( $i = \overline{0, N}$ ) одержимо *узагальнену формулу трапецій*:

$$F \approx \frac{1}{2} \sum_{i=1}^N h_i (f_i + f_{i-1})$$

$$[h_i = x_i - x_{i-1}, f_i = f(x_i)].$$

На рівномірній сітці вона стає такою:

$$F_h^T \approx h \left( \frac{f_0 + f_N}{2} + \sum_{i=1}^{N-1} f_i \right). \quad (3.6)$$

Формула трапецій буде точною для лінійної підінтегральної функції з тієї самої причини, що й формула середніх.

### 3.1.3. Формула парабол (Сімпсона)

Ураховуючи, що головні члени похибок у формулі середніх та формулі трапецій одного порядку, але різних знаків, можна одержати точнішу квадратурну формулу. Для цього скомбінуємо ці формули так, щоб головний член сумарної похибки цих квадратурних формул перетворився на нуль, тобто

$$F \approx (2F^C + F^T) / 3 = \{2f(\bar{x})(b - a) + (b - a)[f(a) + f(b)] / 2\} / 3.$$

У такий спосіб дійдемо *формули парабол*:

$$F \approx \frac{b - a}{6} \left[ f(a) + 4f\left(\frac{a + b}{2}\right) + f(b) \right].$$

Геометрично це означає заміну графіка підінтегральної функції  $f(x)$  квадратичною параболою, яка проходить через точки  $x = a, \frac{a+b}{2}, b$ . Доведено, що формула парабол є точна для кубічної підінтегральної функції  $f(x)$ , оскільки в похибку формули парабол  $R$  входить  $f^{IV}(x)$ , а вона для такої підінтегральної функції дорівнює нулю.

*Узагальнена формула парабол* для рівномірної сітки має вигляд ( $N = 2K$ )

$$F_h^{\text{II}} \approx \frac{b-a}{6K} \left( f_0 + f_{2K} + 4 \sum_{i=1}^K f_{2i-1} + 2 \sum_{i=1}^{K-1} f_{2i} \right). \quad (3.7)$$

Ця формула має такий залишковий член:

$$R \leq -\frac{h^4(b-a)}{180} M_4 \approx O(h^4), \quad M_4 = \max_{[a,b]} |f^{IV}(x)|, \quad (3.8)$$

тобто формула парабол має четвертий порядок похибки, а числовий коефіцієнт досить малий. Через ці обставини формула парабол дає добру точність за відносно невеликого числа вузлів, якщо  $f^{IV}(x)$  не дуже велика.

Зазначимо, що всі розглянуті формули — окремі випадки квадратурних формул Ньютона — Котеса з рівновіддаленими вузлами, в яких підінтегральна функція  $f(x)$  замінено на інтеполяційний полінома Лагранжа.

### 3.1.4. Правило Рунге

Із вищенаведених виразів для залишкових членів квадратурних формул випливає, що до них можна застосовувати правило Рунге, як це було в п. 2.3. Нехай точне значення інтеграла  $F$  пов'язане з наближенням  $F_h$ , знайденим на сітці з кроком  $h$ , так:

$$F = F_h + h^p C + O(h^{p+2}).$$

Для формул середніх (трапецій) і парабол відповідно  $p = 2$  і  $p = 4$ .

Якщо цей вираз порівняти з виразом (2.7), то можна за аналогією записати першу формулу Рунге для інтегрування (оцінка похибки числового інтегрування):

$$R \approx h^p C = \frac{F_h - F_{gh}}{g^p - 1} + O(h^{p+2}). \quad (3.9)$$

Тут  $F_{gh}$  — наближене значення  $F$ , знайдене з кроком  $gh$ .

За аналогією з виразом (2.8) можна записати й другу формулу Рунге, за допомогою якої уточнюють значення інтеграла:

$$\tilde{F} \approx F_h + \frac{F_h - F_{gh}}{g^p - 1}. \quad (3.10)$$

**Приклад 3.1.** Обчислимо інтеграл  $F = \int_0^1 \ln(1+x) dx$ .

Його точне значення легко визначити за формулою Ньютона — Лейбніца

$$F = |(1+x) \ln(1+x) - (1+x)|_0^1 = 2 \ln 2 - 1 = 0,3863.$$

Скориставшись наведеними в табл. 3.1 значеннями підінтегральної функції, знайдемо значення  $F$  за формулами трапецій (3.6), середніх (3.4) і парабол (3.7).

Таблиця 3.1

$x$	0,0000	0,2500	0,5000	0,7500	1,0000
$f$	0,0000	0,2231	0,4055	0,5596	0,6931

Спочатку зробимо це з кроком  $h = 1$ :

$$F_{1,0}^T = 1 \times \frac{f(0) + f(1)}{2} = \frac{0 + 0,6931}{2} = 0,3465;$$

$$F_{1,0}^C = 1 \times f(0,5) = 0,4055.$$

Далі візьмемо  $h = 0,5$ :

$$F_{0,5}^T = 0,5 \times \left[ \frac{0 + 0,6931}{2} + 0,4055 \right] = 0,3760;$$

$$F_{0,5}^C = 0,5 \times (0,2231 + 0,5596) = 0,3914;$$

$$\begin{aligned} F_{0,5}^П &= \frac{1-0}{6} \times [f(0) + 4f(0,5) + f(1)] = \\ &= \frac{0 + 4 \times 0,4055 + 0,6931}{6} = 0,3858. \end{aligned}$$

Для  $h = 0,25$  наведемо остаточні результати:

$$F_{0,25}^T = 0,3837; \quad F_{0,25}^C = 0,3876; \quad F_{0,25}^П = 0,3863.$$

Уточнимо наближене значення інтеграла за правилом Рунге:

$$\tilde{F}_{0,25}^C \approx F_{0,25}^C + \frac{F_{0,25}^C - F_{0,5}^C}{2^2 - 1}$$

$$= 0,3876 + \frac{0,3876 - 0,3914}{2^2 - 1} = 0,3863.$$

Другий доданок у попередньому виразі — це оцінка похибки інтегрування за формулою середніх ( $-0,0038$ ).

Реалізацію цих розрахунків наведено у дод. 8.

*Зауваження.* У процесі побудови вбудованої функції числового інтегрування для ЕОМ з автоматичним вибором кроку інтегрування, як правило, крок зменшують удвічі й перевіряють, чи досягнуто задану точність  $\epsilon$ , за використання формули (3.9), тобто у разі виконання умови  $|R| \leq \epsilon$ , обчислення припиняють.

### 3.1.5. Квадратурна формула Гаусса

У розглянутих квадратурних формулах вибирали вузли й знаходили ваги, не послугуючись тим самим усіма можливостями загальної квадратурної формули (3.3). Гаусс звернув увагу, що вона має  $2n$  невідомі параметри  $c_i$  та  $x_i$ , тобто саме стільки, скільки параметрів має алгебричний поліном степеня  $m = 2n - 1$ . Він запропонував підбирати ці параметри так, щоб

квадратурна формула була точною для підінтегральної функції  $f(x)$  у вигляді полінома степеня, не вищого за  $m$ .

Спочатку для спрощення розглянемо відрізок  $[-1, 1]$ , тобто на підставі (3.3) матимемо

$$\Phi = \int_{-1}^1 \rho(\xi) f(\xi) d\xi \approx \sum_{i=1}^n \gamma_i f(\xi_i).$$

Отже, знайдемо параметри  $\gamma_i$  і  $\xi_i$  з таких умов:

$$\sum_{i=1}^n \gamma_i \xi_i^j = \int_{-1}^1 \rho(\xi) \xi^j d\xi \quad (j = \overline{0, m}). \quad (A)$$

Це система  $2n$  нелінійних алгебричних рівнянь відносно  $\gamma_i$ ,  $\xi_i$  ( $i = \overline{1, n}$ ).

Для подальшого спрощення покладатимемо  $\rho(\xi) \equiv 1$ .

За  $n = 1$  одержимо  $m = 1$  і система (A) набуде вигляду

$$j = 0: \quad \gamma_1 = \int_{-1}^1 d\xi = 2; \quad j = 1: \quad \gamma_1 \xi_1 = \int_{-1}^1 \xi d\xi = 0;$$

із другого рівняння знайдемо, що  $\xi_1 = 0$ , тобто дійшли відомої формули середніх для відрізка  $[-1, 1]$ :

$$\Phi = \int_{-1}^1 f(\xi) d\xi \approx 2f(0),$$

яка є точна для будь-якого полінома 1-го степеня.

За  $n = 2$  система (A) буде мати такий вигляд ( $m = 3$ ):

$$j = 0: \quad \gamma_1 + \gamma_2 = 2; \quad j = 1: \quad \gamma_1 \xi_1 + \gamma_2 \xi_2 = 0;$$

$$j = 2: \quad \gamma_1 \xi_1^2 + \gamma_2 \xi_2^2 = \frac{2}{3}; \quad j = 3: \quad \gamma_1 \xi_1^3 + \gamma_2 \xi_2^3 = 0.$$

Розв'язавши цю систему, знайдемо, що

$$\gamma_1 = \gamma_2 = 1; \quad \xi_2 = -\xi_1 = \sqrt{3}/3 = 0,5773502692,$$

тобто маємо квадратурну формулу



$$\Phi = \int_{-1}^1 f(\xi) d\xi \approx f\left(\frac{-1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right),$$

яка є точна для будь-якого полінома 3-го степеня.

За довільного  $n$  як вузли квадратурної формули Гаусса беруть нулі поліномів Лежандра

$$P_n(\xi) = \frac{1}{2^n n!} \frac{d^n (\xi^2 - 1)^n}{d\xi^n},$$

а ваги цієї квадратурної формули визначають так:

$$\gamma_i = \frac{2}{(1 - \xi_i^2) [P_n'(\xi_i)]^2} \quad (i = \overline{1, n}).$$

Маючи значення ваг  $\gamma_i$  та вузлів  $\xi_i$  на відрізку  $[-1, 1]$ , значення інтеграла на довільному відрізку  $[a, b]$  обчислюють за такою квадратурною формулою Гаусса:

$$F \approx \frac{b-a}{2} \sum_{i=1}^n \gamma_i f(x_i); \quad (3.11)$$

$$x_i = \frac{a+b}{2} + \frac{(b-a)\xi_i}{2} \quad (i = \overline{1, n}).$$

Похибка квадратурної формули Гаусса має вигляд

$$\max |R| \approx \frac{b-a}{2.5\sqrt{n}} \left(\frac{b-a}{3n}\right)^{2n} M_{2n}$$

$$\left( M_{2n} = \max_{[a,b]} |f^{(2n)}(x)| \right).$$

Зауважимо, що, починаючи з  $n = 4$ , і вузли, і ваги є ірраціональні числа, а кінці  $a$  і  $b$  ніколи не входять до вузлів.

Іншими прикладами квадратурної формули типу Гаусса є формули Чебишова, Ерміта.

Згадані формули і квадратурна формула Маркова (тут обидві межі є вузли інтегрування) розраховані на одержання ви-

сокої точності уже за невеликого числа вузлів  $n \approx 4 \dots 10$ , а тому для цих квадратурних формул узагальнених формул не будуть.

**Приклад 3.2.** Обчислимо інтеграл із прикл. 3.1 за квадратурною формулою Гаусса ( $n = 2$ ).

Тут

$$a = 0, b = 1, x_i = \frac{1+0}{2} + \frac{(1-0)\xi_i}{2} \quad (i = \overline{1,2}),$$

тобто

$$x_1 = 0,2113, x_2 = 0,7887, \gamma_1 = \gamma_2 = 1.$$

Тоді

$$\begin{aligned} F &\approx \frac{b-a}{2} [\gamma_1 f(x_1) + \gamma_2 f(x_2)] = \\ &= \frac{1-0}{2} \times [1 \times \ln(1+0,2113) + 1 \times \ln(1+0,7887)] = 0,3866. \end{aligned}$$

### 3.1.6. Інтегрування за допомогою сплайнів

Очевидно, сплайни мають бути ефективними не лише в апроксимації, а і в числовому інтегруванні. Так, якщо відрізок інтегрування розбити на  $N$  підвідрізків і для них побудувати вільні кубічні сплайни (1.7), а потім підставити їх до (3.1) за умови  $\rho(x) \equiv 1$ , одержимо такий вираз:

$$F \approx \sum_{i=1}^N \left( a_i h_i + \frac{b_i h_i^2}{2} + \frac{c_i h_i^3}{3} + \frac{d_i h_i^4}{4} \right) \quad (h_i = x_i - x_{i-1}).$$

Для практичних розрахунків цю формулу можна переписати так:

$$F \approx \frac{1}{2} \sum_{i=1}^N h_i (f_{i-1} + f_i) - \frac{1}{12} \sum_{i=1}^N h_i^3 (c_i + c_{i+1}) \quad (c_1 = c_{N+1} = 0). \quad (3.12)$$

У цій формулі перший член збігається з формулою трапецій, а другий є поправка до формули трапецій, яку забезпечує використання вільних кубічних сплайнів.

### 3.1.7. Вибір квадратурних формул та їх збіжність

Стосовно вибору квадратурної формули, то, очевидно, за існування неперервної четвертої похідної від підінтегральної функції  $f(x)$  краще користуватись формулою парабол, за існування лише неперервної другої похідної та аналітичного задання  $f(x)$  — формулою середніх, а за табличного — формулою трапецій. Для підінтегральної функції  $f(x)$  високої гладкості найзручнішою є формула Гаусса.

Оскільки узагальнені формули середніх і трапецій за змістом є інтегральні суми, вони мають збігатися до точного значення інтеграла для довільної неперервної функції. Сказане має місце і для формули парабол, оскільки вона — комбінація формул середніх і трапецій). Можна довести збіжність і квадратурної формули Гаусса.

Швидкість збіжності квадратурної формули визначають оцінкою залишкового члена, тобто якщо  $R = O(h^p)$ , квадратурну формулу називають збіжною з  $p$ -порядком збіжності (нагадаємо, що для формул середніх і трапецій  $p = 2$ , для формули парабол  $p = 4$ ). Зазвичай підінтегральна функція при цьому повинна мати похідну, яка входить до залишкового члена.

Доведено, що числове інтегрування стійке за вихідними даними, хоча квадратурні формули нестійкі відносно похибок округлення, але ця нестійкість слабка і проявляється лише за розрахунків з малою кількістю правильних цифр.

## 3.2. Нестандартні квадратурні формули

### 3.2.1. Розривні та знаковмінні функції

У прикладних задачах трапляються випадки, коли підінтегральна функція  $f(x)$  та її похідні — кусково-неперервні на  $[a, b]$ , але в точках розриву наявні усі потрібні однобічні похідні.

Тоді для обчислення відповідного інтеграла відрізок  $[a, b]$  розбивають на підвідрізки неперервності  $f(x)$  та деякого числа її  $p$  нижчих похідних, застосовують квадратурна формула  $q$ -го

порядку точності ( $q \leq p$ ) на цих підвідрізках, а потім результати підсумовують. Якщо одночасно й однаково згупувати сітки на всіх підвідрізках неперервності, порядок точності буде  $q$ , як і для неперервних досить гладких функцій. Тоді, очевидно, можна підвищити порядок точності до  $p$  за правилом Рунге.

Якщо  $f(x)$  має велике число нулів (або екстремумів — нулів похідних), крок інтегрування слід брати істотно меншим за відстань між сусідніми нулями функції та її похідних.

### 3.2.2. Нелінійні квадратурні формули

Нагадаємо, що застосування нелінійної апроксимації в наближенні функцій підвищувало точність розрахунків, особливо для швидкозмінних функцій. В інтегруванні вирівнювальні змінні підбирають так, щоб уже два вільні параметри забезпечували задовільну апроксимацію. З цією метою на відрізку  $[a, b]$  вводять сітку й на кожному підвідрізку сітки підінтегральну функцію замінюють нелінійною інтерполяційною формулою, параметри якої можна виразити через табличні значення функції.

Наприклад, якщо функція  $f(x)$  близька до експоненти, то після вирівнювання її можна апроксимувати за допомогою виразу (1.6), інтегрування якого в кожному інтервалі  $x_{i-1} \leq x \leq x_i$  приводить до квадратурної формули

$$F = \int_a^b f(x) dx \approx \sum_{i=1}^N (x_i - x_{i-1}) (f_i - f_{i-1}) / \ln(f_i / f_{i-1}).$$

Зауважимо, що одержана квадратурна формула дещо нагадує узагальнену формулу трапецій. Однак ця формула для функцій неекспоненціального характеру не може забезпечувати добру точність, бо, як і в нелінійній апроксимації, тут немає універсальних вирівнювальних перетворень.

### 3.2.3. Змінна межа інтегрування

Часом виникає потреба в обчисленні інтеграла

$$F(x) = \int_a^x \rho(\xi) f(\xi) d\xi.$$

Зрозуміло, що його можна обчислювати для кожного значення верхньої межі  $x$  як інтеграл зі сталими межами за однією з квадратурних формул, що особливо невигідно в разі великої кількості значень  $x$ . Краще вибрати деяку грубішу сітку і скласти таблицю значень інтеграла на цій сітці  $F_n = F(x_n)$  за квадратурною формулою високої точності, а тоді користуватись такою формулою:

$$F(x) = F_n + \int_{x_n}^x \rho(\xi)f(\xi)d\xi \quad (x_n \leq x < x_{n+1}).$$

Тут інтеграл уже можна обчислювати за простими квадратурними формулами.

Крім того, маючи таблицю  $F(x_n)$ , можна знаходити  $F(x)$  інтерполяцією за цією таблицею. Природно, маючи і похідну інтеграла  $F'(x) = \rho(x)f(x)$ , узагалі краще скористатись інтерполяційним поліномом Ерміта.

### 3.2.4. Невласні інтеграли

*Інтеграл з нескінченними межами* можна обчислювати за допомогою різних прийомів.

*Заміною змінної* інтегрування іноді можна межі інтегрування перетворити на скінченні. Наприклад, для інтеграла

$$F = \int_a^{\infty} f(x)dx \quad (a > 0) \tag{3.13}$$

заміна  $x = \frac{a}{1-t}$  перетворює півпряму  $[a, \infty)$  на відрізок  $[0, 1]$ .

Тепер, якщо підінтегральна функція разом із деяким числом похідних залишається обмеженою, то застосовують стандартні квадратурні формули.

Далі, можна, *обрізавши верхню межу* в (3.13), тобто вибравши досить велике  $b$  (щоб значення інтеграла  $\int_b^{\infty} f(x)dx$  було меншим за допустиму похибку розрахунків  $\epsilon$ ), інтеграл  $\int_a^b f(x)dx$  обчислити за стандартними квадратурними форму-

лами. Проте оскільки біля верхньої межі  $b$  підінтегральна функція  $f(x)$  мала, тут особливо вигідно скористатись квазірівномірними сітками (див. п.2.4). Можна також застосувати квадратурні формули *типу Гаусса* або *нелінійні*.

На практиці трапляються інтеграли, в яких межі інтегрування скінченні, але  $f(x)$  *стає нескінченною в деяких точках відрізка  $[a, b]$* . Наприклад, поблизу особливої точки  $x_{от}$  підінтегральна функція така:  $|f(x)| \leq M|x - x_{от}|^\alpha$ , де  $-1 < \alpha < 0$ . Очевидно, особливі точки розбивають відрізок інтегрування на відрізки неперервності підінтегральної функції.

Розглянемо прийоми обчислення інтегралів, в яких *особливі* одна або обидві межі.

Іноді *підінтегральну функцію можна розбити на суму*, тобто

$$f(x) = \varphi(x) + \psi(x),$$

де  $\varphi(x)$  — обмежена функція, а  $\psi(x)$  можна інтегрувати аналітичними методами.

Тоді інтеграл від  $\varphi(x)$  обчислюють за квадратурними формулами, а від  $\psi(x)$  — точно. Зазвичай підінтегральну функцію розбивають на суму в якнайпростішому вигляді.

Якщо ж підінтегральну функцію *можна записати як добуток*, тобто  $f(x) = \varphi(x)\rho(x)$ , де  $\varphi(x)$  — обмежена, а  $\rho(x)$  — невід'ємна й інтегровна на відрізку, то інколи вдається розглядати  $\rho(x)$  як вагову і скористатись квадратурні формули типу Гаусса.

Крім того, можна побудувати *нестандартні квадратурні формули*, які явно враховують характер особливостей. Причому добре побудована нестандартна квадратурна формула матиме один і той самий порядок точності на рівномірних і нерівномірних сітках.

### 3.3. Кратні інтеграли

Розглянемо  $K$ -вимірний інтеграл вигляду

$$I = \int_G f(x) dx, \quad (3.14)$$

де  $x = (x_1, x_2, \dots, x_K)$ , а  $G$  —  $K$ -вимірна область інтегрування.

Далі для спрощення в основному розглядатимемо подвійні інтеграли, тобто  $K = 2$ .

### 3.3.1. Метод комірок

Метод комірок — найпростіший для обчислення кратних інтегралів. Спочатку розглянемо випадок прямокутної області інтегрування:

$$G = \{x_i \in [a_i, b_i]; i = 1, 2\}.$$

Аналогічно формулі середніх замінимо підінтегральну функцію на її значення у центрі ваги прямокутника у результаті матимемо такий вираз:

$$I = \int_{a_2}^{b_2} \int_{a_1}^{b_1} f(x) dx \approx Sf(\bar{x}) \quad (3.15)$$

$$\left[ S = (b_1 - a_1)(b_2 - a_2), \bar{x}_i = \frac{a_i + b_i}{2} (i = 1, 2) \right].$$

Для підвищення точності область  $G$  розіб'ємо на прямокутні комірки і в кожній з них для обчислення кратних інтегралів застосовуємо формулу (3.15), позначивши через  $S^{(j)}$  та  $\bar{x}^{(j)} = (\bar{x}_1^{(j)}, \bar{x}_2^{(j)})$  відповідно площу та координати центра ваги комірки. Тоді

$$I = \int_{a_2}^{b_2} \int_{a_1}^{b_1} f(x) dx \approx \sum_j S^{(j)} f(\bar{x}^{(j)}). \quad (3.16)$$

Сума праворуч від знака  $\approx$  подібна до інтегральної, тому для неперервної  $f(x)$  вона збігається зі значенням інтеграла, якщо периметри всіх комірок прямують до нуля.

Можна довести, що похибка (3.15) має вигляд (аналогічно формулі середніх)

$$R \approx \frac{S}{24} \sum_{i=1}^2 (b_i - a_i)^2 f_{,ii},$$

а в разі *узагальненої формули методу комірок* (3.16), якщо сторони прямокутника розбиті відповідно на підвідрізки

$$h_i = \frac{(b_i - a_i)}{N_i} \quad (i = 1, 2),$$

$$R \approx \frac{1}{24} \sum_{i=1}^2 h_i^2 \int_G f_{,ii} dx \approx O(h_1^2 + h_2^2).$$

Отже, (3.16) має другий порядок точності, тобто можна скористатись правилом Рунге, якщо зазвичай кроки за осями зменшувати в одне й те саме число разів, тобто залишати сталим  $h_1/h_2$ .

Зрозуміло, що формули (3.15) і (3.16) можна поширити і на область  $G$  складнішої форми.

Очевидно, (3.15) буде точною для підінтегральної функції  $f = Ax_1 + Bx_2 + C$ , тобто вона відповідає апроксимації  $f(x)$  площиною. Вона також буде точною для згадуваних підінтегральних функцій у разі довільної області  $G$ , якщо  $S$  — площа  $G$ , а  $\bar{x}$  — координати центра її ваги, які знаходять за формулами

$$S = \int_G dx; \quad \bar{x}_1 = \frac{1}{S} \int_G x_1 dx; \quad \bar{x}_2 = \frac{1}{S} \int_G x_2 dx.$$

Зазвичай це досить легко зробити, коли область  $G$  простої форми (трикутник, прямокутник, круг). Сказане стосується й (3.16), якщо  $G$  обмежена ламаною, оскільки її завжди можна розбити на прямокутники і трикутники.

У разі криволінійної межі  $G$  на неї накладають прямокутну сітку. Ті комірки, усі точки яких належать  $G$ , назвемо *внутрішніми*, а ті, частина точок яких не належить  $G$ , — *межовими*.

Площею останніх вважатимемо площу, що потрапляє у  $G$ : цю площу обчислюють наближено, замінивши істинну межу на хорду. Підставивши ці площі до інтеграла (3.16), знайдемо значення інтеграла.

Оцінимо похибку (3.16) для *довільної межі*. Очевидно, похибка обчислення у внутрішній комірці має порядок  $O(h^2)$ , а



в межевій —  $O(h)$ , бо центр відповідної прямокутної комірки не збігається з центром ваги частини комірки, що входить до інтеграла. Проте оскільки межових комірок приблизно в  $N$  разів менше, ніж внутрішніх, то після додавання за всіма комірками загальна похибка все-таки буде  $O(h^2)$ , якщо  $f(x)$  двічі неперервно диференційована, а межа — кусково-гладка крива. Це означає другий порядок точності (3.16).

Оскільки знаходження площі межової комірки досить трудомістке, інтеграли по межових комірках обчислюють наближено або взагалі їх не включають до суми. Тоді похибка кубатурної формули буде  $O(h)$  і для одержання задовільної точності потрібна докладніша сітка.

Метод комірок легко поширити на більше число вимірювань. Зважаючи на труднощі за використання цього методу для довільних областей завжди бажано шляхом заміни змінних інтегрування перетворити область  $G$  на прямокутний паралелепіпед (прямокутник).

### 3.3.2. Послідовне інтегрування

Спочатку розглянемо інтеграл на прямокутнику, який розбито на комірки. Його можна обчислити послідовним інтегруванням:

$$I = \int_{a_2}^{b_2} \int_{a_1}^{b_1} f(x_1, x_2) dx_1 dx_2 = \int_{a_2}^{b_2} F(x_2) dx_2:$$

$$F(x_2) = \int_{a_1}^{b_1} f(x_1, x_2) dx_1.$$

Кожен із однократних інтегралів легко обчислити на заданій сітці за квадратурною формулою типу (3.3). Послідовне інтегрування за обома координатами приводить до кубатурних формул, які є прямий добуток одновимірних квадратурних формул:

$$I \approx \sum_j c_{2j} F(x_{2j}); \quad F(x_{2j}) \approx \sum_i c_{1i} f(x_{1i}, x_{2j}),$$

або

$$I \approx \sum_{ij} c_{ij} f(x_{1i}, x_{2j}) \quad (c_{ij} = c_{1i} c_{2j}). \quad (3.17)$$

Якщо, наприклад, за кожною координатою вибрана формула трапецій і сітка рівномірна, то ваги кубатурної формули дорівнюватимуть  $\frac{c_{ij}}{h_1 h_2} = 1, 1/2, 1/4$  відповідно для внутрішніх, межових і кутових вузлів сітки.

Як і для одновимірного випадку, для двічі неперервно диференційованих функцій  $f(x)$  ця формула має 2-й порядок точності і до неї можна застосовувати правило Рунге.

Узагалі за різними напрямками можна використовувати квадратурні формули різних порядків точності  $p$  і  $q$ , але для забезпечення можливості застосування правила Рунге слід величину  $h_1^p / h_2^q$  зберігати сталою.

Якщо підінтегральна функція має достатню кількість неперервних похідних, то за ваги і вузли кубатурної формули (3.17) можна взяти такі:

$$\begin{aligned} c_{ij} &= \frac{1}{4}(b_1 - a_1)(b_2 - a_2)\gamma_i \gamma_j; \\ x_{1i} &= \frac{1}{2}((a_1 + b_1) + (b_1 - a_1)\xi_i); \\ x_{2j} &= \frac{1}{2}((a_2 + b_2) + (b_2 - a_2)\xi_j) \quad (i = \overline{1, n}; j = \overline{1, m}), \end{aligned}$$

де  $\xi, \gamma$  — вузли і ваги відповідних квадратурної формули Гауса.

Методом послідовного інтегрування можна скористатись і для довільної області інтегрування  $G$ .

Наприклад, застосування послідовного інтегрування для обчислення інтеграла

$$I = \int_{a_1}^{b_1} \int_{\varphi(x_1)}^{\psi(x_1)} f(x_1, x_2) dx_1 dx_2$$

потребує таких виразів для ваг і вузлів:

$$c_{ij} = \frac{1}{4}(b_1 - a_1)(b_{2i} - a_{2i})\gamma_i \gamma_j;$$

$$\begin{aligned}
 x_{1i} &= \frac{1}{2}((a_1 + b_1) + (b_1 - a_1)\xi_i); \\
 x_{2ij} &= \frac{1}{2}((a_{2i} + b_{2i}) + (b_{2i} - a_{2i})\xi_j) \\
 [a_{2i} = \varphi(x_{1i}); b_{2i} = \psi(x_{1i}) \quad (i = \overline{1, n}; j = \overline{1, m})].
 \end{aligned}
 \tag{3.18}$$

### 3.3.3. Метод Монте-Карло

Метод Монте-Карло — це числовий метод розв'язування математичних задач їх моделюванням *випадковими величинами*, що набув значного поширення з появою ЕОМ. При цьому, як правило, для обчислень використовують *рівномірно розподілену на відрізку [0,1] випадкову величину  $\gamma$ , тобто таку, яка з однаковою ймовірністю набуває довільних значень на ньому*.

Метод Монте-Карло має дві важливі особливості. Перша — простота структури обчислювального алгоритму. Як правило, складають програму для здійснення одного випадкового випробування, яке потім повторюють  $N$  разів. Друга особливість — похибка обчислень, як правило, пропорційна величині  $\sqrt{\frac{C}{N}}$ , де  $C$  — деяка стала, а  $N$  — число випробувань.

Розглянемо один із варіантів застосування методу Монте-Карло для обчислення кратних інтегралів спочатку на прикладі одновимірного інтеграла

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

Нехай  $0 \leq f(x) \leq f_{\max}$ . Для застосування методу Монте-Карло перейдемо до змінних, які належать одиничному квадрату, тобто виконаємо такі заміни змінних:

$$x = a + (b - a)\xi; \quad \eta = \frac{f(x)}{f_{\max}} \quad (0 \leq \xi \leq 1; 0 \leq \eta \leq 1).$$

Відтак матимемо

$$F = \int_a^b f(x) dx = S\Phi,$$

де  $S = (b - a)f_{\max}$  — площа прямокутника, обмеженого віссю  $x$  та прямими  $x = a$ ,  $x = b$ ,  $y = f_{\max}$ ,

$$\Phi = \int_0^1 \varphi(\xi) d\xi = \int_0^1 \int_0^{\varphi(\xi)} d\eta d\xi \quad \text{—}$$

площа під кривою  $\eta = \varphi(\xi) \equiv f[a + (b - a)\xi]$  в одиничному квадраті  $[0 \leq \xi \leq 1, 0 \leq \eta \leq 1]$ .

Отже, для обчислення інтеграла

$$\Phi = \int_0^1 \varphi(\xi) d\xi$$

скористаємось згаданою рівномірно розподіленою випадковою величиною  $\gamma$ .

Розглядатимемо послідовні пари рівномірно розподілених випадкових чисел  $(\gamma^{(2j-1)}, \gamma^{(2j)})$  як координати точок  $M_j[(\xi^{(j)}, \eta^{(j)}) \quad (j = \overline{1, N})]$  в одиничному квадраті на площині  $\xi, \eta$ . Очевидно, ці точки будуть випадковими й рівномірно розподіленими в ньому. Позначатимемо через  $n$  ту частину точок з їх загального числа  $N$ , для яких виконується умова потрапляння точки під криву  $\gamma^{(2j)} < \varphi(\gamma^{(2j-1)})$ .

Геометрично очевидно, що площа під кривою  $\eta = \varphi(\xi)$  наближено дорівнює  $\frac{n}{N}$ , тобто

$$\Phi \approx \frac{n}{N}.$$

Отже, значення шуканого інтеграла визначатимемо за формулою

$$F = \int_a^b f(x) dx \approx (b - a) f_{\max} \frac{n}{N}.$$

Наприклад, для обчислення інтеграла  $\Phi = \int_0^1 \xi d\xi$  було зроблено 20 випробувань. З них для 11 умов потрапляння виконалась. Отже,

$$\Phi \approx \frac{11}{20} = 0,55 \text{ (точне значення } 0,5).$$

Тепер розглянемо обчислення методом Монте-Карло *K*-кратного інтеграла (3.14). Аналогічно одновимірному інтегралу розмістимо всю фігуру в одиничному  $K + 1$ -вимірному гіперкубі. Припустимо, що  $G$  —  $K$ -вимірний паралелепіпед, тобто  $a_i \leq x_i \leq b_i$  ( $i = \overline{1, K}$ ). Тоді для переходу до одиничного куба потрібно виконати такі заміни змінних інтегрування:

$$x_i = a_i + (b_i - a_i) \xi_i \quad (0 \leq \xi_i \leq 1; i = \overline{1, K})$$

та

$$\eta = \varphi(\xi_1, \xi_2, \dots, \xi_K) = \frac{f(x)}{f_{\max}} \quad (0 \leq \eta \leq 1).$$

Відтак можна записати:

$$I = \tilde{V} \bar{I},$$

де позначено

$$\bar{I} = \int_{G'} d\xi_1 d\xi_2 \dots d\xi_K d\eta; \quad \tilde{V} = f_{\max} V; \quad V = \prod_{i=1}^K (b_i - a_i).$$

Тут  $V$  — об'єм області інтегрування  $G$ , обмеженої площинами  $x = a_i$  та  $x = b_i$  ( $i = \overline{1, K}$ ), а  $G'$  — область інтегрування з об'ємом  $\tilde{V}$ , що знаходиться у  $K + 1$ -вимірному одиничному гіперкубі, яка обмежена площиною  $\eta = 0$  та поверхнею  $\eta = \varphi(x)$ .

Для обчислення інтеграла

$$\bar{I} = \int_{G'} d\xi_1 d\xi_2 \dots d\xi_K d\eta$$

візьмемо  $K + 1$  послідовність рівномірно розподілених на відріжку  $[0, 1]$  випадкових чисел  $(\gamma_1^{(j)}, \gamma_2^{(j)}, \dots, \gamma_K^{(j)}, \gamma_{K+1}^{(j)})$  і розглянемо їх як координати точок

$$M_j \left( \xi_1^{(j)}, \xi_2^{(j)}, \dots, \xi_K^{(j)}, \eta^{(j)} \right) \quad (j = \overline{1, N})$$

у  $K + 1$ -вимірному одиничному гіперкубі. Як і в однократному інтегралі, позначимо через  $n$  ту частину точок  $M_j$  ( $j = \overline{1, N}$ ) від їх загального числа  $N$ , яка потрапляє до  $G'$ , тобто для якої виконується умова

$$\gamma_{K+1}^{(j)} < \varphi \left( \gamma_1^{(j)}, \gamma_2^{(j)}, \dots, \gamma_K^{(j)} \right).$$

Тоді наближене значення цього інтеграла буде таким:

$$\tilde{I} \approx \frac{n}{N}.$$

Отже,

$$I = \int_{a_1}^{b_1} \int_{a_2}^{b_2} \dots \int_{a_K}^{b_K} f(x) dx_1 dx_2 \dots dx_K \approx \tilde{V} \tilde{I} = f_{\max} \prod_{i=1}^K (b_i - a_i) \frac{n}{N}. \quad (3.19)$$

**Приклад 3.3.** Обчислимо інтеграл

$$I = \int_G (x_1^2 + x_2^2) dx_1 dx_2, \quad (3.20)$$

де

$$G = [0, 5 \leq x_1 \leq 1; 0 \leq x_2 \leq 2x_1 - 1].$$

Значення (3.20) легко знайти послідовним застосуванням формули Ньютона — Лейбніца

$$I = \int_{0.5}^1 \int_0^{2x-1} (x_1^2 + x_2^2) dx_1 dx_2 = \frac{7}{32} = 0.2188.$$

1. Спочатку скористаємось *формулою комірок* (3.15), розглянувши всю область  $G$  як трикутник з координатами центру ваги  $\bar{x} = \left( \frac{5}{6}, \frac{1}{3} \right)$  та площею  $S = \frac{1}{4}$ . Тоді

$$I \approx \frac{1}{4} \left[ \left( \frac{5}{6} \right)^2 + \left( \frac{1}{3} \right)^2 \right] = \frac{29}{144} \approx 0.2014.$$

Далі лише для ілюстрації методу комірок накладемо на область  $G$  прямокутну сітку з кроками  $h_1 = \frac{1}{4}$  та  $h_2 = \frac{1}{2}$ , унаслідок чого отримаємо внутрішню прямокутну комірку з площею  $S_2 = \frac{1}{8}$  і координатами центра  $\bar{x}_2 = \left(\frac{7}{8}, \frac{1}{4}\right)$  та дві межові трикутні комірки з площами і координатами центрів відповідно

$$S_1 = S_3 = \frac{1}{16}; \quad \bar{x}_1 = \left(\frac{2}{3}, \frac{1}{6}\right); \quad \bar{x}_3 = \left(\frac{11}{12}, \frac{2}{3}\right).$$

Знехтувавши межовими комірками, згідно з формулою (3.15) отримаємо

$$I^B \approx \frac{1}{8} \left[ \left(\frac{7}{8}\right)^2 + \left(\frac{1}{4}\right)^2 \right] = \frac{53}{512} \approx 0,1035.$$

Доданок від межових комірок набуде такого вигляду:

$$I^M \approx \frac{1}{16} \left[ \left(\frac{2}{3}\right)^2 + \left(\frac{1}{6}\right)^2 \right] + \left[ \left(\frac{11}{12}\right)^2 + \left(\frac{2}{3}\right)^2 \right] = \frac{253}{2304} \approx 0,10981.$$

Остаточно матимемо

$$I \approx I^B + I^M = \frac{53}{512} + \frac{253}{2304} = \frac{983}{4608} \approx 0,2133.$$

Підрахунки зі сталими кроками за обома координатами  $h_1 = h_2 = h$  для  $h = 0,1$  та  $h = 0,05$  дали такі результати:

— без урахування межових комірок:  $I_{0,1} \approx 0,1300$ ;

$$I_{0,05} \approx 0,1926; \quad \tilde{I}_{0,05} \approx 0,1926 + \frac{0,1926 - 0,1300}{2^2 - 1} = 0,2135.$$

Тут  $\delta = |(I - I_{0,05}) / I| \cdot 100 = 2,4 \%$ ;

— з урахуванням межових комірок:

$$\tilde{I}_{0,05} \approx 0,2145; \delta = 1,94 \%$$

Реалізацію розрахунків за цим методом наведено у дод. 9 (без урахування межових комірок) — 10 (з урахуванням).

2. Обчислимо інтеграл (3.20) *методом послідовного інтегрування*, скориставшись для знаходження ваг і вузлів їхніми значеннями для квадратурної формули Гаусса за  $n = 2$ . Проміжні розрахунки наведено в табл. 3.2  $[f(x^{i,j}) = f(x_{1i}, x_{2j})]$ .

Таблиця 3.2

$i$			1	2
$c_{1i}$			0,25000	0,25000
$x_{1i}$			0,60566	0,60566
$c_{2ij}$	$j$	1	0,10566	0,39434
		2	0,10566	0,39434
$x_{1ij}$	$j$	1	0,04466	0,16667
		2	0,16667	0,62201
$c_{ij}$	$j$	1	0,02642	0,009858
		2	0,02642	0,009858
$c_{ij}f(x^{i,j})$	$j$	1	0,00974	0,08159
		2	0,01042	0,11699

Підставивши проміжні дані у формулу (3.17) з урахуванням виразу (3.18), отримуємо  $I \approx 0,21875$ , тобто результат практично збігається з точним значенням.

Реалізацію знаходження цих результатів наведено в дод. 11.

3. Для обчислення інтеграла (3.20) методом Монте-Карло перейдемо до трикратного інтеграла вигляду

$$I = \int_{0,5}^1 \int_0^{2x_1-1} \int_0^{x_1^2+x_2^2} dx_1 dx_2 dz. \tag{3.20}$$



Оскільки значення  $z \in [0,2]$ , тобто  $f_{\max} = 2$ , а отже, слід ввести нову змінну  $\eta = \frac{z}{2}$ . тоді цей інтеграл перейде в інтеграл  $I = \int_{G'} dx_1 dx_2 d\eta$ . де область  $G'$  обмежена площинами  $x_1 = 0$ ,  $x_1 = 1$ ,  $x_2 = 0$ ,  $x_2 = 2x_1 - 1$ ,  $\eta = 0$  та поверхнею  $\eta = \frac{(x_1^2 + x_2^2)}{2}$ , тобто належить одиничному кубу

$$x_i \in [0,1] \quad (i = 1,2), \quad \eta \in [0,1].$$

Тут  $S(V)$  — площа трикутника ( $S=1/4$ ), а  $\tilde{V} = f_{\max} S = 1$

Під час розрахунку методом Монте-Карло було взято 1000 трійок значень рівномірно розподіленої величини  $\gamma$ . З'ясувалось, що 205 з них потрапила в область  $G'$ . Отже, маємо

$$I \approx 0,2050.$$

Реалізацію обчислень за (3.20) наведено в дод. 12, а за (3.20) — у дод. 13, тобто тут бралися значення підінтегральної функції у 1000 випадкових точках області  $G$  і тоді  $I \approx 0,2126$ .

### 3.3.4. Вибір кубатурних формул

Розглянемо сіткові і статистичний методи обчислення кратних інтегралів. Який з них обрати? Зрозуміло, що метод Монте-Карло має невисоку точність і для обчислення однократного інтеграла він явно невигідний, у той час як для багатьох вимірювань ситуація різко змінюється.

Доведено, що для одержання заданої точності  $\varepsilon$  у разі  $K$ -кратного інтеграла сітковим (різницеvim) методом потрібно виконати близько  $\left(\frac{1}{\varepsilon}\right)^{K/p}$  обчислень підінтегральної функції, де  $p$  — порядок точності сіткової формули, а в методі Монте-

Карло — приблизно  $\left(\frac{1}{\varepsilon}\right)^2$  обчислень підінтегральної функції незалежно від вимірності інтеграла  $K$ .

Отже, якщо  $\frac{K}{p} < 2$ , то доцільніші сіткові методи; якщо ж  $\frac{K}{p} > 2$  — метод Монте-Карло. Так, за  $p = 2$  тривимірний інтеграл обчислюють сітковим методом, а за  $K = 5$  — метод Монте-Карло. Якщо ж  $p = 1$  (існує лише одна похідна), то метод Монте-Карло прийнятний навіть і за  $K = 3$ .



**Задачі для самостійного розв'язування**

1. Обчисліть інтеграл  $\int_1^9 \frac{dx}{x+1}$ , скориставшись формулами середніх, трапецій і парабол, уточнюючи їх значення за правилом Рунге ( $h_1 = 4, h_2 = 2$ ).

2. Знайдіть значення інтеграла  $\int_0^1 \frac{\sin x}{x} dx$  за квадратурною формулою Гаусса з двома вузлами.

3. Обчисліть інтеграл  $\int_0^\pi dx_1 \int_{x_1/2}^{2x_1} x_1 \sin x_2 dx_2$ , застосувавши формули комірок з чотирма відрізками та послідовного інтегрування. За вузли і ваги в разі послідовного інтегрування візьміть вирази (3.18).

4. За оцінкою (3.8) визначте, яким має бути крок, щоб похибка визначення інтеграла за квадратурною формулою парабол від функції  $f(x) = x^4$  була не більшою  $10^{-6}$  на відрізьку  $[0, 1]$ .

## Розділ 4 НЕЛІНІЙНІ АЛГЕБРИЧНІ РІВНЯННЯ

---

Досить часто у прикладних дослідженнях виникає потреба у розв'язанні нелінійних алгебричних рівнянь або їх систем, причому це може бути як самостійною задачею, так і допоміжною у складнішій задачі (наприклад, визначення вільних параметрів за нелінійної апроксимації, у разі розв'язування нелінійних диференціальних та інтегральних рівнянь).

### 4.1. Рівняння з одним невідомим

Розглянемо рівняння

$$f(x) = 0, \quad (4.1)$$

для якого потрібно знайти всі або лише деякі корені, де  $f(x)$  — неперервна функція.

Якщо  $f(x) = 0$ , то  $x$  називають *коренем рівняння* (4.1) або нулем функції  $f(x)$ . Здебільшого корені (4.1) можна знаходити лише наближено.

Процес знаходження коренів (4.1) включає такі етапи:

- 1) визначення кількості, характеру та розміщення коренів;
- 2) визначення наближених їхніх значень;
- 3) вибір потрібних коренів та їх обчислення із заданою точністю  $\epsilon$ .

Перші два етапи можна виконати аналітично або графічно. Наприклад відомо, що алгебричний поліном

$$P_n(x) = \sum_{k=0}^n a_k x^k$$

має  $n$  комплексних нулів, не обов'язково різних, причому всі вони належать колу

$$|x_p| \leq 1 + \frac{\max(|a_0|, |a_1|, \dots, |a_{n-1}|)}{|a_n|} \quad (p = \overline{1, n}),$$

що не є оптимальна оцінка.

Якщо потрібно відшукати лише *дійсні корені*, складають таблицю значень  $f(x)$ , і якщо у двох сусідніх вузлах значення різних знаків, то між ними лежить непарне число коренів рівняння (щонайменше один). Якщо, крім того, ці вузли знаходяться близько один від одного, його можна вважати одним. Зауважимо, що виявити парну кратність коренів за допомогою таблиці складно.

Для відокремлення дійсних коренів можна застосувати такий машинний алгоритм. Інтервал відрізка визначення функції  $f(x)$ , що нас цікавить, розбиваємо на підінтервали  $(x_{i-1}, x_i)$  ( $i = \overline{1, N}$ ). Зміна знака функції у двох сусідніх точках засвідчує наявність коренів на підінтервалі (*метод перебирання*).

За таблицею значень  $f(x)$  можна побудувати її графік. Тоді точки перетину графіка з віссю  $x$  і будуть наближеними значеннями коренів. До речі, за цим графіком часто можна виявити корені *парної кратності*.

Іноді рівняння (4.1) вдається замінити на тотожне йому рівняння  $\varphi(x) = \psi(x)$ , в якому функції  $y_1 = \varphi(x)$  та  $y_2 = \psi(x)$  мають прості графіки. Абсциси точок перетину цих графіків і будуть наближеними значеннями коренів вихідного рівняння.

Уважатимемо, що корені (4.1) *відокремлені*, якщо для кожного кореня існує окіл, де немає інших коренів цього рівняння.

Наближені значення коренів уточнюють різними ітераційними методами. Розглянемо деякі з них.

#### 4.1.1. Метод половинного ділення

Нехай знайдено такі  $x^{(0)}$  і  $x^{(1)}$ , що  $f(x^{(0)})f(x^{(1)}) \leq 0$ , тобто на відріжку  $[x^{(0)}, x^{(1)}]$  є хоча б один корінь.

Знайдемо середину відрізка  $x^{(2)} = \frac{x^{(0)} + x^{(1)}}{2}$  й обчислимо  $f(x^{(2)})$ . З двох одержаних відрізків  $[x^{(0)}, x^{(2)}]$ ,  $[x^{(2)}, x^{(1)}]$  виберемо той, для якого виконується умова  $f(x^{(2)})f(x^{(\text{гран})}) \leq 0$ .

Процес послідовного ділення відрізка навпіл продовжуватимемо до виконання умови

$$|x^{(s+1)} - x^{(s)}| \leq 2\varepsilon$$

або

$$f\left(\frac{x^{(s+1)} + x^{(s)}}{2}\right) = 0.$$

За корінь візьмемо

$$\underline{x} \approx \frac{x^{(s+1)} + x^{(s)}}{2}.$$

Метод половинного ділення є простий, надійний і стійкий до похибок округлення алгоритмом знаходження коренів (4.1), що забезпечує гарантовану точність і збіжність до простого кореня для неперервних функцій  $f(x)$ , у тім числі й для недиференційованих (тобто кусково-неперервних).

Головними недоліками методу половинного ділення є повільна збіжність (за одну ітерацію точність збільшується вдвічі, тобто уточнення трьох цифр потребує до 10 ітерацій) та неможливість його поширення на системи нелінійних рівнянь.

### 4.1.2. Вилучення коренів

Відомо, що загальним недоліком майже всіх ітераційних методів є збіжність ітерацій невідомо до якого кореня, якщо їх на відрізку декілька. Цей недолік усувають вилученням уже знайдених коренів.

Нехай  $\underline{x}_1$  — простий корінь рівняння (4.1), а  $f(x)$  — ліпшицова неперервна функція. Тоді допоміжна функція  $g(x) = \frac{f(x)}{x - \underline{x}_1}$  неперервна, причому всі нулі функцій  $g(x)$  і  $f(x)$

збігаються за винятком  $\underline{x}_1$ , бо  $g(\underline{x}_1) \neq 0$ . Якщо  $\underline{x}_1$  — кратний корінь рівняння (4.1), то він буде нулем  $g(x)$  кратності, на одиницю меншої; інші ж нулі обох функцій, як і раніше, будуть однаковими.

Якщо бути точним, то можна знайти лише наближене значення кореня  $\underline{x}_1 \approx \underline{x}_1$ . Тому функція  $g(x) = \frac{f(x)}{x - \underline{x}_1}$  має нуль у точці  $\underline{x}_1$  і полюс у близькій до неї точці  $\underline{x}_1$ , і лише на деякій відстані від цього кореня вона близька до  $g(x)$ . Щоб це не позначалось на результаті під час знаходження наступних коренів, потрібно обчислювати кожен корінь з високою точністю, особливо якщо він кратний або поруч нього знаходиться інший корінь рівняння.

Крім того, завершальні ітерації поблизу уточнюваного кореня слід виконувати за вихідною функцією  $f(x)$ , використовуючи як нульове наближення кореня його останню ітерацію, знайдену за функцією  $g(x)$ . Особливо це важливо, якщо знаходити багато коренів, тоді чим більше їх вилучено, тим менше корені  $G(x) = \frac{f(x)}{\prod_i (x - \underline{x}_i)}$  відповідають решті коренів  $f(x)$ .

З урахуванням цих застережень і того, що розрахунки проводять із 8–10 правильними цифрами, можна знайти до 20 коренів, про розміщення яких спочатку нічого невідомо (у тім числі коренів високої кратності ( $p \sim 5$ )).

### 4.1.3. Метод простих ітерацій

Метод простих ітерацій належить до найзагальніших методів розв'язування рівняння (4.1). Для його застосування рівняння  $f(x) = 0$  потрібно замінити на тотожне йому

$$x = \varphi(x). \quad (4.2)$$

Це можна зробити багатьма способами, поклавши, наприклад,

$$\varphi(x) = x + g(x)f(x),$$

де  $g(x)$  — довільна неперервна знакостала функція.

Виберемо нульове наближення  $x^{(0)}$  і будемо знаходити наступні наближення за формулою

$$x^{(s+1)} = \varphi(x^{(s)}) \quad (s = 0, 1, \dots). \quad (4.3)$$

Якщо  $x^{(s)}$  прямує до деякої межі  $\underline{x}$ , то ця межа й буде коренем вихідного рівняння (4.1).

Дослідимо умови збіжності методу простих ітерацій. Нехай функція  $\varphi(x)$  має неперервну похідну, тоді на підставі теореми Лагранжа можна записати таку формулу:

$$x^{(s+1)} - \underline{x} = \varphi(x^{(s)}) - \varphi(\underline{x}) = (x^{(s)} - \underline{x})\varphi'(\xi).$$

Тут точка  $\xi$  лежить між точками  $x^{(s)}$  і  $\underline{x}$ .

Збіжність до кореня визначається значенням похідної  $\varphi'(\xi)$ . Якщо всюди  $|\varphi'(x)| \leq q < 1$ , то відрізки  $|x^{(s)} - \underline{x}|$  зменшуються як і члени геометричної прогресії зі знаменником  $q < 1$ , і послідовність  $x^{(s)}$  збігається до кореня  $\underline{x}$  для довільного  $x^{(0)}$ .

Якщо  $|\varphi'(\underline{x})| > 1$ , то через неперервність  $|\varphi'(x)|$  більше за одиницю і в деякому околі кореня; тоді ітерації не збігаються. Якщо  $|\varphi'(\underline{x})| < 1$ , але віддалік від кореня  $|\varphi'(x)| > 1$ , то ітерації збігаються, якщо  $x^{(0)}$  вибране досить близько до кореня; у разі ж довільного  $x^{(0)}$  збіжності може і не бути.

Усе це стосується й ліпшицово неперервних функцій без змін.

Очевидно, що чим менший знаменник  $q$ , тим швидша збіжність. Поблизу кореня асимптотична збіжність визначається величиною  $|\varphi'(\underline{x})|$  і буде особливо швидкою, якщо  $|\varphi'(\underline{x})| = 0$ .

Отже, тепер зрозуміло, як важливо вдало вибрати вираз  $\varphi(x)$ .

На практиці ітерації припиняють за виконання умови

$$|x^{(s+1)} - x^{(s)}| \leq \varepsilon.$$

#### 4.1.4. Метод дотичних

Метод дотичних ще називають методом Ньютона або методом лінеаризації. Якщо функція  $f(x)$  двічі неперервно диференційована на досліджуваному проміжку, а  $x^{(s)}$  є деяке наближення до кореня  $\underline{x}$ , тоді можна, дещо перетворивши рівняння (4.1)

$$f(\underline{x}) = f[x^{(s)} + (\underline{x} - x^{(s)})] = 0,$$

розкласти його в ряд Тейлора й лінеаризувати:

$$f(x^{(s)}) + (\underline{x} - x^{(s)})f'(\xi) = 0.$$

Наближено замінюючи  $f'(\xi)$  на її значення у відомій точці  $x^{(s)}$ , а  $\underline{x}$  на  $x^{(s+1)}$ , одержимо такий ітераційний процес:

$$x^{(s+1)} = x^{(s)} - \frac{f(x^{(s)})}{f'(x^{(s)})} \quad (s = 0, 1, \dots). \quad (4.4)$$

Геометрично це є заміна графіка  $y = f(x)$  на кожній ітерації, дотичною до нього.

Загалом метод дотичних можна розглядати як окремий випадок методу простих ітерацій, якщо покласти

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

Тоді

$$\varphi'(x) = \frac{ff''}{f'^2}.$$

Нехай  $\underline{x}$  є  $p$ -кратний корінь рівняння (4.1), тоді поблизу нього можна записати функцію  $f(x)$  так:

$$f(x) \approx a(x - \underline{x})^p.$$

Звідси випливає, що

$$\varphi'(x) \approx \frac{p-1}{p},$$



тобто

$$0 \leq \varphi'(x) < 1.$$

Для простого кореня

$$p = 1 \text{ і } \varphi'(x) = 0.$$

Отже, якщо  $x^{(0)}$  обрано досить близьким до кореня, процес (4.4) збіжний.

У разі довільного  $x^{(0)}$  ітерації збігаються, якщо всюди

$$|ff''| < f'^2;$$

у протилежному випадку збіжність має місце не для будь-якого  $x^{(0)}$ , а лише в деякому околі кореня.

Поблизу простого кореня збіжність монотонна, але на відстані від нього можлива немонотонність ітерацій. Можна довести, що ітерації збігаються до кореня з того кінця відрізка, де виконується умова

$$f(x^{(\text{гран})})f''(x^{(\text{гран})}) \geq 0. \quad (4.5)$$

Оцінимо збіжність поблизу простого кореня. За визначенням методу простих ітерацій

$$\underline{x} - x^{(s+1)} = \varphi(\underline{x}) - \varphi(x^{(s)}).$$

Розклавши праву частину рівняння в ряд Тейлора і врахувавши, що  $\varphi(\underline{x}) = 0$ , одержимо

$$x^{(s+1)} - \underline{x} = \frac{(x^{(s)} - \underline{x})^2 \varphi''(\xi)}{2} \quad [\xi \in (x^{(s)}, \underline{x})],$$

тобто похибка кожного наступного наближення приблизно дорівнює квадрату похибки попереднього наближення, що ілюструє швидку збіжність поблизу кореня. Зазвичай на відстані від кореня подібні міркування можуть не справджуватись.

Якщо знаходять корінь високої кратності,  $f'(x)$  у знаменнику (4.4) може стати малим, тому  $\frac{f(x)}{f'(x)}$  потрібно обчислювати досить точно. Проти інших похибок метод дотичних стійкий.

Для знаходження нулів довільної диференційованої аналітично заданої функції  $f(x)$  найчастіше використовують метод дотичних, особливо якщо відомі відповідні нульові наближення.

#### 4.1.5. Метод січних

У разі числового задання рівняння (4.1) похідну у знаменнику (4.4) замінюють на першу поділену різницю й уточнення виконують за формулою

$$x^{(s+1)} = x^{(s)} - \frac{[x^{(s)} - x^{(s-1)}]f(x^{(s)})}{f(x^{(s)}) - f(x^{(s-1)})} \quad (s = 1, 2, \dots). \quad (4.6)$$

Геометрично це є заміна графіка  $y = f(x)$  на кожній ітерації січною.

Для початку процесу вибирають  $x^{(0)}$  і  $x^{(1)}$ . Цей процес називають *двокроковим*, бо потрібно знати два попередні наближення.

Такі заміни в методі дотичних істотно впливають на характер ітерацій і змінюють швидкість збіжності. Так, якщо за методом дотичних порядок швидкості збіжності дорівнює 2, то за методом січних — 1,62. Незважаючи на це методом січних користуватися доцільніше, оскільки за однакового обсягу обчислень цим методом можна зробити більше ітерацій, ніж методом дотичних, бо в останньому на кожній ітерації потрібно обчислювати як значення функції, так і її похідної.

Недоліком методу січних є наявність у знаменнику різниці функцій, яка може призвести до втрати значущих цифр поблизу кореня («розхитування» розрахунку), а це обмежує точність знаходження кореня. «Розхитування» можна уникнути за допомогою так званого *прийому Гарвіка*. Скажімо, узявши немале  $\varepsilon$ , слід вести розрахунки до виконання умови  $|x^{(s+1)} - x^{(s)}| \leq \varepsilon$ ; потім продовжувати вести розрахунки до зме-

ншення величини  $|x^{(s+1)} - x^{(s)}|$ . З початком збільшення різниці розрахунки слід припинити, останню ітерацію не використовувати, тобто за наближене значення кореня взяти  $x^{(s)}$ .

### 4.1.6. Метод парабол

Згідно з методом парабол функцію  $f(x)$  замінимо на інтерполяційний поліном Ньютона 2-го степеня, тобто

$$f(x) \approx P_2(x) = f(x^{(s)}) + (x - x^{(s)}) \left[ f(x^{(s)}, x^{(s-1)}) + (x - x^{(s-1)}) f(x^{(s)}, x^{(s-1)}, x^{(s-2)}) \right] \quad (s = 2, 3, \dots). \quad (4.7)$$

Прирівнявши праву частину до нуля, зведемо її до вигляду

$$az^2 + bz + c = 0,$$

де

$$c = f(x^{(s)}); \quad z = x - x^{(s)};$$

$$a = f(x^{(s)}, x^{(s-1)}, x^{(s-2)});$$

$$b = a(x^{(s)} - x^{(s-1)}) + f(x^{(s)}, x^{(s-1)}).$$

Менший за модулем із двох коренів цього квадратного рівняння й визначає наступне наближення, тобто

$$x^{(s+1)} = x^{(s)} + z.$$

Очевидно, для початку розрахунку потрібно знати  $x^{(0)}$ ,  $x^{(1)}$ ,  $x^{(2)}$ . Інколи беруть довільно три числа, які належать до відрізка відокремлення. Отже, метод парабол — *трикроковий*.

Зауважимо, що заміна похідних зумовлює зниження порядку швидкості збіжності до 1,84, тобто він менший, ніж у методі дотичних, але більший, ніж у методі січних і простих ітерацій. Застосовувати метод парабол вищого степеня недоцільно, бо швидкість збіжності за будь-яких умов нижча за квадратичну. У методі парабол «розхитування» поблизу кореня може

бути ще більшим і через використання другої поділеної різниці.

Зазначимо, що навіть за дійсного  $x^{(0)}$  метод парабол може привести до комплексного кореня. Особливо зручно ним користуватися для знаходження всіх нулів алгебричного поліному високого степеня.

**Приклад 4.1.** Розглянемо різні методи розв'язування рівняння

$$f(x) \equiv x - \operatorname{tg}x = 0 \quad \left[ x = L \sqrt{\frac{P_{\text{кр}}}{(EI)_{\text{min}}}} \right], \quad (4.8)$$

що визначає критичну силу  $P_{\text{кр}}$  пружного стержня, верхній кінець якого шарнірно рухомий, а нижній затиснутий. Тут  $L$  — довжина стержня, а  $(EI)_{\text{min}}$  — мінімальна жорсткість поперечного перетину стержня на згинання.

Для відокремлення коренів можна побудувати графіки функцій  $x$  та  $\operatorname{tg}x$  і тоді точки їх перетину будуть нульовими наближеннями для коренів рівняння (4.8). Оскільки, як правило, інтерес становить найменша критична сила  $P_{\text{кр}}$ , знайдемо перший ненульовий корінь у проміжку  $\left(\pi, \frac{3\pi}{2}\right)$  з похибкою  $\varepsilon = 0,001$ .

1. Спочатку для знаходження кореня скористаємось методом половинного ділення, взявши за відрізок відокремлення  $[x^{(0)}, x^{(1)}]$  відрізок  $[4,4; 4,5]$ . Оскільки  $f(4,4) \approx 1,304$ , а  $f(4,5) \approx -0,137$ , то, очевидно, на цьому відрізку є щонайменше один корінь.

Визначимо середину відрізка:

$$x^{(2)} = \frac{4,4 + 4,5}{2} = 4,45.$$

Відтак знайдемо значення функції :

$$f(x^{(2)}) = 4,45 - \operatorname{tg}4,45 \approx 0,727.$$

Звідси можна дійти висновку, що корінь належить відрізку  $[4,45; 4,5]$ . У середині цього відрізка ( $x^{(3)} = 4,475$ ) значення функції  $f(x^{(3)}) \approx 0,342$ .

Отже, наступний відрізок  $[4,475; 4,5]$  і  $x^{(4)} = 4,4875$ .  
Якщо за умову закінчення ітерацій взяти

$$|x^{(s+1)} - x^{(s)}| \leq 2\varepsilon,$$

то через 9 ітерацій матимемо таке наближене значення кореня  $\underline{x} \approx 4,4934$ .

Реалізацію розрахунків за методом половинного ділення наведено в дод. 14.

2. Зазначимо, що в разі застосування методу простих ітерацій не можна брати залежність (4.8) у вигляді

$$x = \operatorname{tg} x,$$

оскільки тоді в околі кореня

$$\varphi'(x) = \cos^{-2} x > 1.$$

Якщо ж формулу (4.8) записати у вигляді

$$x = \pi + \operatorname{arctg} x,$$

тоді в околі кореня похідна набуде вигляду  $\varphi'(x) = \frac{1}{1+x^2} < 1$ .

Якщо за нульове наближення взяти  $x^{(0)} = 4,5$ , то відповідно до (4.3) матимемо такий ітераційний процес:

$$x^{(1)} = \pi + \operatorname{arctg} 4,5000 = 4,4937;$$

$$x^{(2)} = \pi + \operatorname{arctg} 4,4937 = 4,4934.$$

Отже, через дві ітерації матимемо таке наближене значення кореня  $\underline{x} \approx 4,4934$ .

Реалізацію розрахунків за методом простих ітерацій наведено в дод. 15.

3. Оскільки за методом дотичних (4.4) нульове наближення слід вибирати за умови (4.5), знайдемо вираз для другої похідної від  $f(x)$ , тобто

$$f''(x) = -2 \frac{\sin x}{\cos^3 x}.$$

Відомо, що в третьому квадранті  $\sin x$  і  $\cos x$  від'ємні, а отже, друга похідна і функція у точці  $x = 4,5$  одного знака, а саме тут  $x^{(0)} = 4,5$ .

Звідси маємо такий ітераційний процес:

$$x^{(1)} = 4,5 - \frac{4,5 - \operatorname{tg} 4,5}{1 - \cos^{-2} 4,5} = 4,5019;$$

$$x^{(2)} = 4,5019 - \frac{4,5019 - \operatorname{tg} 4,5019}{1 - \cos^{-2} 4,5019} = 4,4937.$$

Реалізацію розрахунків за методом дотичних наведено в дод. 16.

4. За методом січних (4.6) потрібно мати два наближення до кореня, причому як і в методі дотичних, біля тієї самої межової точки, де виконується умова (4.5). Наприклад, якщо взяти  $x^{(0)} = 4,55$  і  $x^{(1)} = 4,5$ , матимемо такий ітераційний процес:

$$x^{(2)} = 4,5 - \frac{(4,5 - 4,55)(4,5 - \operatorname{tg} 4,5)}{(4,5 - \operatorname{tg} 4,5) - (4,55 - \operatorname{tg} 4,55)} = 4,4952.$$

Наступні два наближення до кореня  $x^{(2)} = 4,4935$  і  $x^{(3)} = 4,4934$  різняться лише на 0,0001.

Реалізацію розрахунків за методом січних наведено в дод. 17.

5. За методом парабол (4.7) потрібно мати три наближення до кореня, причому як і в методі дотичних, біля тієї самої межової точки, де виконується умова (4.5). Наприклад, якщо взяти  $x^{(0)} = 4,51$ ,  $x^{(1)} = 4,5$ , і  $x^{(2)} = 4,49$ , то через 2 ітерації  $x^{(4)} = 4,4934$ .

Реалізацію розрахунків за методом парабол наведено в дод. 18.

## 4.2. Системи нелінійних рівнянь

Розглянемо систему нелінійних рівнянь з  $n$  невідомими:

$$f_k(x_1, x_2, \dots, x_n) = 0 \quad (k = \overline{1, n}). \quad (4.9)$$

Увівши позначення  $x = (x_1, x_2, \dots, x_n)$  і  $f = (f_1, f_2, \dots, f_n)$ , запишемо цю систему у векторному вигляді:

$$f(x) = 0. \quad (4.9')$$

Розв'язання цієї системи є набагато складніша задача, ніж розв'язання одного рівняння. Такі системи, як правило, розв'язують лише ітераційними методами.

Якщо  $n = 2$ , то за нульове наближення  $x^{(0)}$  до розв'язку  $x$  візьмемо координати точок перетину графіків

$$f_1(x_1, x_2) = 0 \quad \text{і} \quad f_2(x_1, x_2) = 0.$$

Для трьох і більше змінних задовільних способів знаходження нульового наближення  $x^{(0)}$  немає, але іноді його можна вибрати, виходячи з фізичних міркувань або аналізу задачі.

До речі, якщо  $f(x) = 0$ , де  $x = x_1 + ix_2$ , замість такого рівняння розглянемо систему нелінійних рівнянь:

$$f_1(x_1, x_2) = 0; \quad f_2(x_1, x_2) = 0, \quad (4.10)$$

де

$$f_1(x_1, x_2) = \operatorname{Re} f(x_1 + ix_2); \quad f_2(x_1, x_2) = \operatorname{Im} f(x_1 + ix_2).$$

### 4.2.1. Метод простих ітерацій

У цьому методі, як і у разі одного невідомого, рівняння (4.9) зведемо до вигляду

$$x = \varphi(x) \quad \text{або} \quad x_k = \varphi_k(x) \quad (k = \overline{1, n}).$$

Вибравши нульове наближення  $x^{(0)}$ , наступні наближення знайдемо за формулами

$$x^{(s+1)} = \varphi(x^{(s)}) \text{ або } x_k^{(s+1)} = \varphi_k(x^{(s)}) \quad (k = \overline{1, n}; s = 0, 1, \dots), \quad (4.11)$$

причому якщо ітерації збіжні, то вони збігаються до розв'язку  $\underline{x}$  (звичайно, якщо він існує).

Збіжність досліджують так само, як і у випадку одного невідомого. Нехай  $\underline{x}_k$  — складові розв'язку, тоді

$$x^{(s+1)} - \underline{x} = \varphi(x^{(s)}) - \varphi(\underline{x}) = \left[ \frac{\partial \varphi(\xi)}{\partial x} \right] (x^{(s)} - \underline{x})$$

або

$$x_k^{(s+1)} - \underline{x}_k = \varphi_k(x^{(s)}) - \varphi_k(\underline{x}) = \sum_{i=1}^n \frac{\partial \varphi_k(\xi)}{\partial x_i} [x_i^{(s)} - \underline{x}_i] \\ (k = \overline{1, n}; s = 0, 1, \dots).$$

Тут  $\xi = (\xi_1, \xi_2, \dots, \xi_n)$  — точка на лінії, яка з'єднує точки  $x^{(s)}$  та  $\underline{x}$ . Це означає, що вектор похибки нового наближеного розв'язку дорівнює вектору похибки попередньої ітерації, помноженому на матрицю похідних  $\left[ \frac{\partial \varphi_k}{\partial x_i} \right]$ .

Якщо яка-небудь норма цієї матриці, погоджена з нормою вектора похибки, менша за одиницю, то норма похибки спадає від ітерації до ітерації.

На практиці як таку матрицю використовують матрицю з елементами

$$M_{ki} = \max \left[ \frac{\partial \varphi_k}{\partial x_i} \right].$$

Достатньою умовою збіжності є  $\|M_{ki}\| < 1$ . Доведено, що коли ітерації збіжні за однією з норм, то вони збігаються й за іншими.

Іноді  $\varphi(x)$  беруть у такому вигляді:

$$\varphi(x) = x + \Lambda f(x) \text{ або } \varphi_k(x) = x_k + \sum_{i=1}^n \lambda_{ki} f_i(x) \quad (k = \overline{1, n}),$$

де  $\Lambda$  — невироджена матриця; наприклад, за  $n = 2$  її елементи можна знайти як розв'язок таких СЛАР :



$$P \times \begin{pmatrix} \lambda_{11} \\ \lambda_{12} \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}; \quad P \times \begin{pmatrix} \lambda_{21} \\ \lambda_{22} \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}. \quad (4.12)$$

Тут позначено

$$P = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_2}{\partial x_1} \\ \frac{\partial f_1}{\partial x_2} & \frac{\partial f_2}{\partial x_2} \end{pmatrix}^{(0)}.$$

За такого визначення елементів матриці  $\Lambda$  умова збіжності ітераційного процесу (4.11) буде виконана, якщо тільки частинні похідні функцій  $f_1(x)$  та  $f_2(x)$  в околі розв'язку  $\underline{x}$  змінюватимуться не дуже швидко.

Якщо (4.11) записати у вигляді процесу Зейделя, тобто

$$\begin{aligned} x_k^{(s+1)} &= \varphi_k \left( x_1^{(s+1)}, x_2^{(s+1)}, \dots, x_{k-1}^{(s+1)}, x_k^{(s)}, \dots, x_n^{(s)} \right) \\ & \quad (k = \overline{1, n}; s = 0, 1, \dots), \end{aligned}$$

то збіжність буде дещо швидша, однак залишатиметься лінійною.

#### 4.2.2. Метод Ньютона

Нехай відомий якийсь наближений розв'язок  $x^{(s)}$  до  $\underline{x}$ . Як і в разі одного невідомого, запишемо систему (4.9')

$$f(\underline{x}) = 0$$

у вигляді

$$f(x^{(s)} + \Delta x^{(s)}) = 0, \quad (A)$$

де  $\Delta x^{(s)} = \underline{x} - x^{(s)}$  ( $s = 0, 1, \dots$ ).

Розклавши рівняння (A) в ряд Тейлора і обмежившись першим диференціалом, тобто лінеаризуючи його, одержимо

$$f(x^{(s)}) + \left[ \frac{\partial f(x^{(s)})}{\partial x} \right] \Delta x^{(s)} = 0.$$

Остаточно матимемо таку СЛАР відносно приростів  $\Delta x_i^{(s)}$ :

$$\sum_{i=1}^n \frac{\partial f_k(x^{(s)})}{\partial x_i} \Delta x_i^{(s)} = -f_k(x^{(s)}) \quad (4.13)$$

$$(k = \overline{1, n}; s = \overline{0, 1, \dots}).$$

Розв'язавши цю СЛАР методом Гаусса (див. п. 5.1.1), наступні наближення уточнюємо за формулою

$$x^{(s+1)} = x^{(s)} + \Delta x^{(s)} \quad (s = \overline{0, 1, \dots}). \quad (4.14)$$

Як і в разі одного невідомого, метод Ньютона можна звести до методу простих ітерацій і виконати відповідний аналіз збіжності. Проте перевірити достатню умову збіжності майже ніколи не вдається. Зазначимо лише очевидний результат: *у досить малому околі розв'язку  $x$  ітерації збігаються, якщо*

$\det \left[ \frac{\partial f}{\partial x} \right] \neq 0$ , причому збіжність квадратична. Отже, за вдалого вибору  $x^{(0)}$  метод Ньютона збігатиметься, і досить швидко (3 – 5 ітерацій).

На практиці ітерації припиняють за виконання умови

$$\sqrt{\frac{1}{n} \sum_{k=1}^n [x_k^{(s+1)} - x_k^{(s)}]^2} \leq \varepsilon. \quad (4.15)$$

Програмування методу Ньютона і розрахунки за ним складніші за методу простих ітерацій, зате збіжність квадратична.

**Приклад 4.2.** Для ілюстрації застосування викладених методів розглянемо знаходження розв'язку рівняння

$$shx + x = 0, \quad (4.16)$$

що має місце в теорії пружності під час визначення напруженого стану однорідної ізотропної скінченної смуги.

Оскільки ненульові розв'язки цього рівняння комплексні, а саме  $x = x_1 + ix_2$ , запишемо залежність (4.16) у вигляді системи рівнянь, тобто

$$\begin{aligned} shx_1 \cos x_2 + x_1 &= 0; \\ chx_1 \sin x_2 + x_2 &= 0. \end{aligned} \quad (4.17)$$

Можна довести, що розв'язки рівняння (4.16) асимптотично збігаються з розв'язками рівняння

$$\exp(\zeta) + 2\zeta = 0,$$

які мають такий вигляд:

$$\zeta_k \approx \pm \ln(2a_k) \pm ia_k$$

$$\left[ a_k = 2\pi \left( k - \frac{1}{4} \right); k = 1, 2, \dots \right].$$

Очевидно, їх можна взяти за нульові наближення  $x^{(0)}$  для розв'язання системи нелінійних рівнянь (4.17). Визначатимемо значення лише мінімального за модулем кореня, який знаходиться у першому квадранті, тобто візьмемо  $k = 1$ . Тоді нульові наближення будуть такі:

$$x_2^{(0)} = a_1 = 2\pi \left( 1 - \frac{1}{4} \right) \approx 4,5;$$

$$x_1^{(0)} = \ln(2a_1) \approx 2,2.$$

1. Розглянемо застосування методу простих ітерацій для знаходження розв'язку системи рівнянь (4.17). Оскільки безпосередньо для цієї системи умови збіжності методу простих ітерацій не виконуються, матрицю  $\Lambda$  знайдемо за допомогою виразу (4.12). Тут маємо:

$$\frac{\partial f_1}{\partial x_1} = chx_1 \cos x_2 + 1;$$

$$\frac{\partial f_1}{\partial x_2} = -shx_1 \sin x_2;$$

$$\frac{\partial f_2}{\partial x_1} = shx_1 \sin x_2;$$

$$\frac{\partial f_2}{\partial x_2} = chx_1 \cos x_2 + 1.$$

(4.18)

Після підстановки в ці рівняння  $x^{(0)}$  одержимо значення частинних похідних:

$$\frac{\partial f_1^{(0)}}{\partial x_1} = 0,0371; \quad \frac{\partial f_1^{(0)}}{\partial x_2} = 4,3570;$$

$$\frac{\partial f_2^{(0)}}{\partial x_1} = -4,3570; \quad \frac{\partial f_2^{(0)}}{\partial x_2} = 0,0371.$$

Отже, елементи матриці  $\Lambda$  визначимо з такої СЛАР:

$$\begin{pmatrix} 0,0371 & -4,3570 \\ 4,3570 & 0,0371 \end{pmatrix} \begin{pmatrix} \lambda_{11} \\ \lambda_{12} \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix};$$

$$\begin{pmatrix} 0,0371 & -4,3570 \\ 4,3570 & 0,0371 \end{pmatrix} \begin{pmatrix} \lambda_{21} \\ \lambda_{22} \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \end{pmatrix}.$$

Розв'язавши ці системи, одержимо такі значення елементів матриці  $\Lambda$ :

$$\lambda_{11} = \lambda_{22} = -0,00195;$$

$$\lambda_{12} = 0,22950;$$

$$\lambda_{21} = -0,22950.$$

Вектор правих частин системи (4.11) матиме вигляд

$$\begin{pmatrix} \varphi_1(x) \\ \varphi_2(x) \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} -0,00195 & 0,22950 \\ -0,22950 & -0,00195 \end{pmatrix} \times \begin{pmatrix} shx_1 \cos x_2 + x_1 \\ chx_1 \sin x_2 + x_2 \end{pmatrix}.$$

Поклавши  $x_1^{(0)} = 2,2$ ,  $x_2^{(0)} = 4,5$ , отримаємо такі значення складових розв'язку:

$$x_1^{(1)} = 2,2 + (-0,00195 \times 1,26046 + 0,22950 \times 0,03473) = 2,20551;$$

$$x_2^{(1)} = 4,5 + (-0,22950 \times 1,26046 - 0,00195 \times 0,03473) = 4,21065;$$

$$x_1^{(2)} = 2,20551 + (-0,00195 \times 0,0497 - 0,22950 \times 0,18416) = 2,24768;$$

$$x_2^{(2)} = 4,21065 + (-0,22950 \times 0,0497 - 0,00195 \times 0,18416) = 4,19888.$$

Продовжимо цей процес до виконання умови (4.15) за  $\varepsilon = 0,001$  й дістанемо таке:

$$x_1^{(3)} = 2,25477; \quad x_1^{(4)} = 2,25151; \quad x_1^{(5)} = 2,25042;$$

$$x_2^{(3)} = 4,21060; \quad x_2^{(4)} = 4,21354; \quad x_2^{(5)} = 4,21269.$$

Отже, можна покласти  $x \approx (2,25042; 4,21269)$ .

Реалізацію розрахунків за методом простих ітерацій наведено у дод. 19.

2. Знайдемо тепер розв'язок системи рівнянь (4.17) за допомогою методу Ньютона, тобто скориставшись формулами (4.13) і (4.14).

Оскільки вирази для  $\frac{\partial f_i(x)}{\partial x_j}$  ( $i, j = 1, 2$ ) уже знайдено [див. вираз (4.18)], можна записати систему нелінійних рівнянь (4.13) для (4.17) у такому вигляді:

$$\begin{pmatrix} chx_1 \cos x_2 + 1 & -shx_1 \sin x_2 \\ shx_1 \sin x_2 & chx_1 \cos x_2 + 1 \end{pmatrix}^{(s)} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix}^{(s)} = \\ = - \begin{pmatrix} shx_1 \cos x_2 + x_1 \\ chx_1 \sin x_2 + x_2 \end{pmatrix}^{(s)}.$$

Поклавши в цій системі  $x_1^{(0)} = 2,2$ ,  $x_2^{(0)} = 4,5$  одержимо таку систему:

$$\begin{pmatrix} 0,037104 & 4,35695 \\ -4,35695 & 0,037104 \end{pmatrix}^{(0)} \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \end{pmatrix}^{(0)} = - \begin{pmatrix} 1,26046 \\ 0,03473 \end{pmatrix}^{(0)}.$$

Розв'язавши цю систему, одержимо значення приростів  $\Delta x_1^{(0)} = 0,00551$ ,  $\Delta x_2^{(0)} = -0,28935$  й обчислимо наступне наближення за формулами (4.14) підставленням знайдених приростів у вираз  $x^{(1)} = x^{(0)} + \Delta x^{(0)}$ , унаслідок чого отримаємо перші наближення складових розв'язку:

$$x_1^{(1)} = 2,2 + 0,000551 = 2,20551;$$

$$x_2^{(1)} = 4,5 - 0,28935 = 4,21065.$$

Продовжимо цей процес до виконання умови (4.15) за  $\varepsilon = 0,001$  й дістанемо такі наступні наближення складових розв'язку:

$$x_1^{(2)} = 2,20551 + 0,04637 = 2,25188;$$

$$x_2^{(2)} = 4,21965 + 0,00161 = 4,21226;$$

$$x_1^{(3)} = 2,25188 - 0,00115 = 2,25073;$$

$$x_2^{(3)} = 4,21226 + 0,00013 = 4,21239.$$

Отже, можна покласти  $\underline{x} \approx (2,25073; 4,21239)$ .

Із порівняння результатів підрахунків за обома методами видно, що перші чотири цифри в обох методах збігаються, хоч за методом Ньютона проведено на дві ітерації менше.

Реалізацію розрахунків за методом Ньютона наведено в дод. 20.



### Задачі для самостійного розв'язування

1. Застосувавши наведені вище методи, знайдіть ненульовий корінь із похибкою 0,001 рівняння  $\sin x - \frac{x}{2} = 0$ .
2. Знайдіть розв'язки такої системи нелінійних рівнянь:

$$x_1^3 - x_2 = 0,$$

$$x_1 + 10e^{x_1} = 1$$

з точністю 0,001 обома розглянутими методами.

## Розділ 5

# ЗАДАЧІ ЛІНІЙНОЇ АЛГЕБРИ

---

Розв'язання багатьох задач математики, механіки й інженерної практики зводиться до розв'язування таких основних задач лінійної алгебри:

1) розв'язування систем лінійних алгебричних рівнянь

$$Ax = b \text{ або } \sum_{j=1}^n a_{ij}x_j = b \quad (i = \overline{1, n}); \quad (5.1)$$

2) обчислення визначника  $D = \det(A)$ :

3) відшукування елементів оберненої матриці

$$A^{-1} = [a_{ij}]_{i, j = \overline{1, n}};$$

4) визначення власних значень і відповідних їм власних векторів квадратної матриці  $Ax = \lambda x$  або  $(A - \lambda E)x = 0$ , тобто розв'язання однорідної СЛАР.

### 5.1. Розв'язання системи лінійних алгебричних рівнянь

---

Відомо, що за  $D = 0$  СЛАР (5.1) або не має дійсних розв'язків, або має їх нескінченно багато. Якщо ж  $D \neq 0$ , то розв'язок існує, й до того ж єдиний. Далі розглядатимемо лише такі СЛАР.

За  $n = 2$  має місце така геометрична інтерпретація вищесказаного на площині  $(x_1, x_2)$ . Якщо  $D \neq 0$ , то прямі перетинаються в точці, яка є розв'язок (5.1). Якщо  $D = 0$ , прямі, які відповідають рівнянням (5.1), є збіжні або паралельні.

На практиці, крім існування і єдиності, важливою ще є стійкість відносно малих змін вихідних даних  $b$  і  $A$ . Варіюванням розв'язку  $x = A^{-1}b$  знайдемо, що його приріст є

$\delta x = A^{-1}(\delta b - \delta A x)$ . Це означає, що малим  $\delta b$  і  $\delta A$  відповідають малі  $\delta x$ , тобто формально стійкість є. Однак якщо обернена матриця має великі елементи ( $D \approx 0$ ), то можна дібрати такі вихідні дані ( $A$  і  $b$ ), які суттєво змінять розв'язок  $x$ . У цьому разі СЛАР буде *погано обумовленою*.

У теоретичних дослідженнях обумовленість характеризують числом  $\chi = \|A\| \|A^{-1}\|$ . Число  $\chi$  завжди більше або дорівнює одиниці й чим воно більше, тим гірша обумовленість матриці. На практиці за умову поганої обумовленості беруть  $D \approx 0$ .

Якщо СЛАР має загальний вигляд і  $n < 200$ , доцільно користуватися простими й універсальними прямими (точними) методами (розв'язок знаходиться за скінченне число дій), якщо ж матриця  $A$  слабо заповнена і  $n$  досить велике — ітераційними методами.

### 5.1.1. Прямі (точні) методи

**Метод Гаусса.** Історично першим методом розв'язання СЛАР був метод Крамера, але починаючи з  $n = 4$ , він не є економічний, бо потребує до  $\frac{2n^4}{3}$  арифметичних дій. Найбільш поширеним і ефективним прямим методом є метод Гаусса, в якому приблизно в  $n$  разів менше дій, ніж у методі Крамера.

У методі Гаусса на першому етапі початкову матрицю  $A$  перетворимо на верхню трикутну.

На першому кроці, якщо  $a_{11} \neq 0$ , введемо множники

$$m_i = \frac{a_{i1}}{a_{11}} \quad (i = \overline{2, n}),$$

помножимо перше рівняння на  $m_i$  і віднімемо результат від кожного нижнього рівняння. Внаслідок цього, починаючи з другого, у всіх рівняннях доданки з  $x_1$  будуть відсутні.

Аналогічно за  $a_{22}^{(1)} \neq 0$  вилучимо  $x_2$ .

Узагалі, на  $k$ -му кроці ( $k = \overline{1, n-1}$ ) формули визначення перетворених елементів  $A$  та складових  $b$  мають такий вигляд (формули прямого ходу):



$$\begin{aligned}
 a_{ij}^{(k)} &= a_{ij}^{(k-1)} - m_i^{(k-1)} a_{kj}^{(k-1)} \quad (i = \overline{k+1, n}; j = \overline{k, n}); \\
 b_i^{(k)} &= b_i^{(k-1)} - m_i^{(k-1)} b_k^{(k-1)} \quad (i = \overline{k+1, n}); \\
 \left[ m_i^{(k-1)} &= \frac{a_{ik}^{(k-1)}}{a_{kk}^{(k-1)}} \quad (i = \overline{k+1, n}; a_{kk}^{(k-1)} \neq 0) \right].
 \end{aligned} \tag{5.2}$$

Значення невідомих знайдемо за формулами зворотного ходу:

$$\begin{aligned}
 x_n &= b_n^{(n-1)} / a_{nn}^{(n-1)}; \quad x_j = (b_j^{(j-1)} - \sum_{k=j+1}^n a_{jk}^{(j-1)} x_k) / a_{jj}^{(j-1)} \\
 (j &= \underline{n-1, 1}; a_{1k}^{(0)} = a_{1k}; b_1^{(0)} = b_1).
 \end{aligned} \tag{5.3}$$

Якщо  $a_{kk}^{(k-1)} = 0$ , то переставлянням місцями рядків на головній діагоналі можна розмістити ненульовий елемент (краще, щоб він був максимальним за абсолютною величиною у відповідному стовпчику; тоді цей метод називають методом Гаусса з вибором головного елемента) і продовжити процес вилучення невідомих.

Існують також інші прямі методи розв'язання СЛАР, але для сумісних СЛАР метод Гаусса є один із найефективніших.

**Приклад 5.1.** Розглянемо алгоритм розв'язування СЛАР методом Гаусса такої системи з трьох рівнянь:

$$\begin{pmatrix} 3 & 2 & 2 \\ 2 & -3 & 3 \\ 4 & 1 & -6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 12 \\ 14 \\ 9 \end{pmatrix}.$$

*Прямий хід.* Визначимо множники  $m_i = \frac{a_{i1}}{a_{11}} \quad (i = \overline{2, 3})$

на першому кроці ( $k = 1$ ):

$$m_2 = \frac{2}{3}; \quad m_3 = \frac{4}{3}.$$

Вилучимо невідому  $x_1$  з другого й третього рівнянь за формулами

$$a_{ij}^{(1)} = a_{ij} - m_i a_{kj}; \quad b_i^{(1)} = b_i - m_i b_k \quad (i = \overline{2,3}; j = \overline{1,3}),$$

тобто

$$a_{21}^{(1)} = 2 - \frac{2}{3} \cdot 3 = 0; \quad a_{22}^{(1)} = -3 - \frac{2}{3} \cdot 2 = -\frac{13}{3};$$

$$a_{23}^{(1)} = 3 - \frac{2}{3} \cdot 2 = \frac{5}{3}; \quad a_{31}^{(1)} = 4 - \frac{4}{3} \cdot 3 = 0;$$

$$a_{32}^{(1)} = 1 - \frac{4}{3} \cdot 2 = -\frac{5}{3}; \quad a_{33}^{(1)} = -6 - \frac{4}{3} \cdot 2 = -\frac{26}{3};$$

$$b_2^{(1)} = 14 - \frac{2}{3} \cdot 12 = 6; \quad b_3^{(1)} = 9 - \frac{4}{3} \cdot 12 = -7.$$

Як бачимо, після 1-го кроку СЛАР набуває такого вигляду:

$$\begin{pmatrix} 3 & 2 & 2 \\ 0 & -\frac{13}{3} & \frac{5}{3} \\ 0 & -\frac{5}{3} & -\frac{26}{3} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 12 \\ 6 \\ -7 \end{pmatrix}.$$

На 2-му кроці ( $k=2$ ) вилучимо  $x_2$  з 3-го рівняння:

$$a_{ij}^{(2)} = a_{ij}^{(1)} - m_i^{(1)} a_{kj}^{(1)}; \quad b_i^{(2)} = b_i^{(1)} - m_i^{(1)} b_k^{(1)} \quad (i = 3; j = \overline{2,3});$$

$$m_3^{(1)} = \frac{a_{32}^{(1)}}{a_{22}^{(1)}} = -\frac{5}{3} / \left(-\frac{13}{3}\right) = \frac{5}{13}; \quad b_3^{(2)} = -7 - \frac{5}{13} \cdot 6 = -\frac{121}{13};$$

$$a_{32}^{(2)} = -\frac{5}{3} - \frac{5}{13} \left(-\frac{13}{3}\right) = 0; \quad a_{33}^{(2)} = -\frac{26}{3} - \frac{5}{13} \cdot \left(-\frac{26}{3}\right) = -\frac{121}{13}.$$

У результаті одержали систему з верхньою трикутною матрицею:

$$\begin{pmatrix} 3 & 2 & 2 \\ 0 & -\frac{13}{3} & \frac{5}{3} \\ 0 & 0 & -\frac{121}{13} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 12 \\ 6 \\ -\frac{121}{13} \end{pmatrix}.$$

На цьому закінчується прямий хід методу Гаусса.

*Зворотний хід методу Гаусса* полягає у послідовному знаходженні  $x_3$ ,  $x_2$ ,  $x_1$  відповідно з третього, другого і першого рівнянь, тобто

$$x_3 = \left(-\frac{121}{13}\right) / \left(-\frac{121}{13}\right) = 1;$$

$$x_2 = \left(6 - \frac{5}{3} \cdot 1\right) / \left(-\frac{13}{3}\right) = -1;$$

$$x_1 = [12 - 2 \cdot (-1) - 2 \cdot 1] / 3 = 4.$$

Підстановкою одержаних значень  $x_1$ ,  $x_2$ ,  $x_3$  у вихідну СЛАР легко переконатись, що знайдений вектор  $(4, -1, 1)$  і є її розв'язок.

Реалізацію розрахунків методом Гаусса з вибором головного елемента наведено в дод. 21.

**Метод алгебричної прогонки.** Якщо матриця СЛАР (5.1) тридіагональна, то її запишемо у такому вигляді:

$$a_i x_{i-1} - b_i x_i + c_i x_{i+1} = d_i \quad (a_1 = c_n = 0; i = \overline{1, n}). \quad (5.4)$$

Після вилучення невідомих на нижній діагоналі одержимо формули *зворотного ходу*:

$$x_i = \xi_{i+1} x_{i+1} + \eta_{i+1} \quad (x_{n+1} = 0; i = \overline{n, 1}). \quad (5.5)$$

Коефіцієнти  $\xi_{i+1}$  і  $\eta_{i+1}$  визначимо за формулами *прямого ходу*:

$$\xi_{i+1} = \frac{c_i}{b_i - a_i \xi_i}; \quad \eta_{i+1} = \frac{a_i \eta_i - d_i}{b_i - a_i \xi_i} \quad (5.6)$$

$$(\xi_1 = \eta_1 = x_{n+1} = c_n = 0; i = \overline{1, n}).$$

Формули (5.6) можна одержати, якщо у виразі (5.5) індекс  $i$  зменшити на одиницю, підставити  $x_{i-1}$  у (5.4), а потім одержаний результат порівняти з (5.5). Оскільки  $\xi_1$  та  $\eta_1$  знаходяться біля  $a_1 = 0$ , а  $x_{n+1}$  — біля  $\xi_{n+1}$ , яке дорівнює нулю (воно пов'язане з  $c_n = 0$ ), можна вважати, що  $\xi_1 = \eta_1 = x_{n+1} = 0$ .

Узагалі якщо хоч для одного рівняння справджується умова  $|b_i| \geq |a_i| + |c_i|$ , розв'язок СЛАР існує, єдиний і стійкий до похибок округлення.

**Приклад 5.2.** Розглянемо алгоритм знаходження розв'язку СЛАР з тридіагональною матрицею для такої системи

$$\begin{pmatrix} 2 & 1 & 0 \\ 1 & -1 & 1 \\ 0 & -1 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \\ -8 \end{pmatrix}$$

або

$$\begin{aligned} -(-2)x_1 + x_2 &= 4; \\ x_1 - x_2 + x_3 &= 2; \\ -x_2 - 2x_3 &= -8. \end{aligned}$$

Ураховуючи, що  $c_1 = c_4 = \xi_1 = \eta_1 = x_4 = 0$ , знайдемо значення коефіцієнтів  $\xi_{i+1}$ ,  $\eta_{i+1}$  ( $i = \overline{1,3}$ ) за допомогою формул *прямого ходу* (5.6).

Для скорочення запису введемо такі позначення:

$$A_i = a_i \eta_i - d_i, \quad B_i = b_i - a_i \xi_i.$$

Обчислимо тепер  $\xi_{i+1}$ ,  $\eta_{i+1}$  за формулами

$$\xi_{i+1} = \frac{c_i}{B_i}; \quad \eta_{i+1} = \frac{A_i}{B_i}.$$

Для конкретних значень  $i$  маємо такі значення коефіцієнтів  $\xi_i$ ,  $\eta_i$ :

$$i = 1: A_1 = 0 \times 0 - 4 = -4; \quad B_1 = -2 - 0 \times 0 = -2;$$

$$\xi_2 = \frac{1}{-2} = -\frac{1}{2}; \quad \eta_2 = -\frac{4}{-2} = 2;$$

$$i = 2: A_2 = 1 \times 2 - 2 = 0; \quad B_2 = 1 - (-1) \times \frac{1}{2} = \frac{3}{2};$$

$$\xi_3 = \frac{1}{3/2} = \frac{2}{3}; \quad \eta_3 = \frac{0}{3/2} = 0;$$

$$i = 3: A_3 = -1 \times 0 - (-8) = 8; \quad B_3 = 2 - (-1) \times \frac{2}{3} = \frac{8}{3};$$

$$\xi_4 = \frac{0}{8/3} = 0; \quad \eta_4 = \frac{8}{8/3} = 3.$$

Невідомі знайдемо за формулами *зворотного ходу* (5.5), підставивши в них значення коефіцієнтів:

$$x_3 = 0 \times 0 + 3 = 3;$$

$$x_2 = \frac{2}{3} \times 3 = 2;$$

$$x_1 = \frac{-1}{2} \times 2 + 2 = 1.$$

Відтак підставимо значення  $x_1$ ,  $x_2$ ,  $x_3$  до вихідної СЛАР, і переконаємось, що вектор  $(1, 2, 3)$  і є її розв'язок (див. дод. 22).

### 5.1.2. Ітераційні методи

У разі великих  $n$  ефективнішими, ніж прямі методи розв'язування СЛАР, є ітераційні. Розглянемо один із найпоширеніших ітераційних методів, який відзначається простотою і легкістю програмування, — **метод Гаусса — Зейделя**.

За цим методом на головній діагоналі розмістимо максимальні за абсолютною величиною елементи *відповідних* стовпчиків вихідної матриці, а потім знаходимо невідомі  $x_1$ ,  $x_2$ ,  $x_3, \dots, x_n$  з першого, другого, третього, ...,  $n$ -го рівнянь СЛАР (5.1), тобто

$$x_k = (b_k - \sum_{j \neq k} a_{kj} x_j) / a_{kk} \quad (a_{kk} \neq 0; k = \overline{1, n}).$$

Скориставшись цим записом, побудуємо такий ітераційний процес:

$$x_k^{(s+1)} = \left( b_k - \sum_{j=1}^{k-1} a_{kj} x_j^{(s+1)} - \sum_{j=k+1}^n a_{kj} x_j^{(s)} \right) / a_{kk} \quad (a_{kk} \neq 0), \quad (5.7)$$

$$(k = \overline{1, n}; s = 0, 1, \dots).$$

Взявши за нульове наближення, наприклад,  $x_k^{(0)} = 0$  ( $k = \overline{1, n}$ ), наступні наближення знайдемо за формулами (5.7).

Якщо  $\left| \frac{a_{kj}}{a_{kk}} \right| < 1$ , то процес (5.7) має бути збіжним.

На практиці ітерації завершують після виконання умови

$$\max |x_k^{(s+1)} - x_k^{(s)}| \leq \varepsilon \quad (k = \overline{1, n})$$

чи

$$\max \left| \frac{x_k^{(s+1)} - x_k^{(s)}}{x_k^{(s)}} \right| \leq \varepsilon \quad (k = \overline{1, n}),$$

хоча для цього можна скористатись і формулою (4.15).

**Приклад 5.3.** Розв'яжемо за допомогою методом Гаусса — Зейделя таку СЛАР:

$$\begin{pmatrix} 4 & -1 & 1 \\ 2 & 6 & -1 \\ 1 & 2 & -3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 7 \\ 0 \end{pmatrix}.$$

Цілком очевидно, що розв'язком системи є  $\underline{x} = (1, 1, 1)$ .

Щоб застосувати метод Гаусса — Зейделя, запишемо СЛАР у такому вигляді:

$$x_1 = \frac{4 + x_2 - x_3}{4};$$

$$x_2 = \frac{7 - 2x_1 + x_3}{6};$$

$$x_3 = \frac{x_1 + 2x_2}{3}.$$

Взявши за нульове наближення  $x^{(0)} = (0, 0, 0)$ , знайдемо перше:

$$x_1^{(1)} = \frac{4 + 0 - 0}{4} = 1;$$

$$x_2^{(1)} = \frac{7 - 2 \times 1 + 0}{6} = \frac{5}{6} = 0,83333;$$

$$x_3^{(1)} = \frac{1 + 2 \times \frac{5}{6}}{3} = \frac{8}{9} = 0,8889.$$

Аналогічно знаходимо й наступні наближення:

$$x_1^{(2)} = \frac{4 + \frac{5}{6} - \frac{8}{9}}{4} = \frac{71}{72} = 0,9861;$$

$$x_2^{(2)} = \frac{7 - 2 \times \frac{71}{72} + \frac{8}{9}}{6} = \frac{71}{72} = 0,9861;$$

$$x_3^{(2)} = \frac{\frac{71}{72} + 2 \times \frac{71}{72}}{3} = \frac{71}{72} = 0,9861$$

Складові вектора наближеного розв'язку  $x^{(5)}$  близькі до точного розв'язку  $\underline{x}$  (див. дод. 23).

## 5.2. Обчислення визначника

Для обчислення визначника скористаємося виразом

$$D \equiv \det(A) = \pm \prod_{k=1}^n a_{kk}^{(k-1)}, \quad (5.8)$$

де  $a_{kk}^{(k-1)}$  знайдемо за формулами (5.2) ( $a_{11}^{(0)} = a_{11}$ ).

Якщо число переставлянь рядків парне, то у виразі (5.8) візьмемо знак «+», у протилежному випадку — знак «-».

Визначник тридіагональної матриці знайдемо за виразом

$$D \equiv \det(A) = \prod_{i=1}^n (a_i \xi_i - b_i). \quad (5.9)$$

**Приклад 5.4.** Знайдемо визначники матриці  $A$  з прикл. 5.1 за формулою (5.8), тобто

$$D = \prod_{k=1}^3 a_{kk}^{(k-1)} = 3 \times (-13/3) \times (-121/13) = 121,$$

і тридіагональної матриці з пр. 5.2 за формулою (5.9):

$$D = \prod_{i=1}^3 (a_i \xi_i - b_i) = \prod_{i=1}^3 (-B_i) = [-(-2)] \times \left(-\frac{3}{2}\right) \times \left(-\frac{8}{3}\right) = 8.$$

### 5.3. Відшукування елементів оберненої матриці

Елементи оберненої матриці  $A^{-1} = (\alpha_{ij})_{i,j=\overline{1,n}}$  знайдемо розв'язуванням таких СЛАР:

$$\sum_{k=1}^n a_{ik} \alpha_{kl} = \delta_{il} \equiv \begin{cases} 1 & (i=l) \\ 0 & (i \neq l) \end{cases} \quad (i, l = \overline{1, n}). \quad (5.10)$$

Вираз (5.10) у векторному записі має вигляд  $AA^{-1} = E$ .

### 5.4. Проблема власних значень і векторів матриць

Потреба у розв'язуванні однорідних СЛАР, єдиний розв'язок яких існує лише за певного значення параметра, що входить до них, досить часто виникає у багатьох прикладних задачах. Цей параметр називають *власним значенням*, а відповідний йому ненульовий розв'язок — *власним вектором*.

#### 5.4.1. Повна проблема власних значень

У загальному вигляді задачу на власні значення формують так: потрібно знайти  $n$  дійсних або комплексних чисел  $\lambda$  і відповідних кожному з них ненульових векторів  $x$ . тобто



$$Ax = \lambda x \text{ або } (A - \lambda E)x = 0 \quad (x \neq 0), \quad (5.11)$$

де  $A$  — квадратна матриця  $n$ -го порядку.

Відомо, що для існування ненульового розв'язку однорідної СЛАР (5.11) її визначник має перетворюватися на нуль, тобто

$$D(\lambda) \equiv \det(A - \lambda E) = 0. \quad (5.12)$$

Визначник  $D(\lambda)$  називають *характеристичним поліномом*, а алгебричне рівняння (5.12) — *характеристичним рівнянням*, корені якого слід знаходити методом парабол.

У загальному випадку  $D(\lambda)$  є алгебричний поліном  $n$ -го степеня відносно  $\lambda$  і характеристичне рівняння можна записати у такому вигляді:

$$D(\lambda) \equiv \sum_{i=0}^n (-1)^{i+n} p_i \lambda^{n-i} = 0. \quad (5.13)$$

Коефіцієнти цього рівняння визначимо за виразами

$$p_0 = 1; \quad p_1 = \sum_{i=1}^n a_{ii}; \quad p_2 = \sum_{i < j} \begin{vmatrix} a_{ii} & a_{ij} \\ a_{ji} & a_{jj} \end{vmatrix};$$

$$p_3 = \sum_{i < j < k} \begin{vmatrix} a_{ii} & a_{ij} & a_{ik} \\ a_{ji} & a_{jj} & a_{jk} \\ a_{ki} & a_{kj} & a_{kk} \end{vmatrix}, \dots, \quad p_n = \det(A).$$

Особливо просто всі власні значення можна знайти, якщо  $A$  трикутна або діагональна, бо для них характеристичне рівняння (5.13) набуває вигляду

$$D_{\Delta}(\lambda) \equiv \prod_{i=1}^n (a_{ii} - \lambda) = 0;$$

$$D_{\text{diag}}(\lambda) \equiv \prod_{i=1}^n (d_i - \lambda) = 0. \quad (5.12')$$

Отже, можна записати, що  $\lambda_i = a_{ii}$  або  $\lambda_i = d_i$  ( $i = \overline{1, n}$ ).

Рівняння (5.13) можна одержати методами безпосереднього розгортання визначника (методи Данилевського, Крилова, інтерполяції та ін.). Тут розглянуто найпростіший метод — *ме-*

тод інтерполяції, оскільки цей та й усі інші згадані методи доцільно застосовувати лише для невеликих  $n$ .

Якщо власні значення  $\lambda_i$  ( $i = \overline{1, n}$ ) знайдено точно, то їх можна підставити у систему (5.11) і, поклавши, наприклад,  $x_1^{(i)} = 1$ , розв'язати її як звичайну СЛАР, відкинувши одне з рівнянь. Це можна зробити, оскільки розв'язок однорідної СЛАР знаходиться з точністю до сталого множника.

Розглянемо, як саме задача знаходження власних частот у теорії лінійних коливань зводиться до задачі на власні значення.

**Приклад 5.5.** Власні частоти й відповідні їм власні форми коливань двомасової системи (рис. 5.1), визначимо за такою однорідною СЛАР:

$$\begin{pmatrix} 2c - mp^2 & -c \\ -c & c - mp^2 \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

де  $c$  — жорсткість пружин;  $m$  — маса;  $A_i$  — амплітуди періодичних коливань.

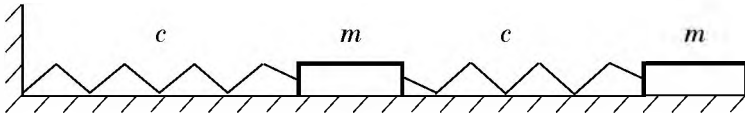


Рис. 5.1

Якщо кожне рівняння поділити на  $c$  ( $c \neq 0$ ) і ввести позначення

$$\lambda = \frac{mp^2}{c}; \quad x_1 = A_1; \quad x_2 = A_2,$$

то одержимо наступну однорідну СЛАР:

$$\begin{pmatrix} 2 - \lambda & -1 \\ -1 & 1 - \lambda \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (5.14)$$

або у векторному вигляді:

$$(A - \lambda E)x = 0,$$

де

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}; \quad (5.15)$$

$x = (x_1, x_2)$ , тобто вихідну задачу зведено до задачі (5.11).

Таким чином, задачу знаходження власних частот зведено до задачі визначення власних значень матриці  $A$  (5.15).

Знайшовши визначник системи (5.14), одержимо характеристичне рівняння  $\lambda^2 - 3\lambda + 1 = 0$ , яке має такі корені:

$$\lambda_1 = \frac{3 + \sqrt{5}}{2} = 2,618; \quad \lambda_2 = \frac{3 - \sqrt{5}}{2} = 0,382.$$

Для знаходження власних векторів підставимо  $\lambda_i$  до СЛАР (5.14), поклавши  $x_1^{(i)} = 1$ :

$$\begin{pmatrix} 2 - \lambda_i & -1 \\ -1 & 1 - \lambda_i \end{pmatrix} \begin{pmatrix} 1 \\ x_2^{(i)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Тоді, наприклад, з першого рівняння знайдемо другу складову власного вектора  $x_2^{(i)} = 2 - \lambda_i$ .

Отже, власні вектори для обох власних значень будуть такими:

$$x^{(1)} = (1; -0,618); \quad x^{(2)} = (1; 1,618).$$

Зазначимо, що далі алгоритми застосування розглянутих методів визначення власних значень і відповідних власних векторів для спрощення демонструватимемо на прикладі задачі (5.14).

**Метод інтерполяції.** Вибравши довільні  $\lambda^{(j)}$  ( $j = \overline{0, n}$ ), які належать до відрізка  $[-\|A\|, \|A\|]$  (природно, їх слід розмістити по відрізку рівномірно), знайдемо  $D(\lambda^{(j)})$  і побудуємо інтерполяційний поліном Ньютона:

$$D(\lambda) = D(\lambda^{(0)}) + \sum_{k=1}^n (\lambda - \lambda^{(0)})(\lambda - \lambda^{(1)}) \dots$$

$$\dots (\lambda - \lambda^{(k-1)}) D(\lambda^{(0)} \dots \lambda^{(k)}). \quad (5.16)$$

З єдиності алгебричного полінома випливає, що інтерполяційний поліном Ньютона (5.16) має бути характеристичним поліномом. Значення  $D(\lambda^{(j)})$  можна знайти за допомогою методу Гаусса. За норму  $\|A\|$  можна взяти

$$\|A\|_c = \max_i \sum_{j=1}^n |a_{ij}|. \quad (5.17)$$

Побудувавши характеристичний поліном, його нулі краще знаходити методом парабол (див. п. 4.1.6).

Метод інтерполяції доцільно застосовувати за  $n \leq 10$ . Якщо  $n > 10$ , то для симетричних матриць слід обирати *метод обер-тань*.

Якщо  $A$  тридіагональна, визначник знаходимо за рекурентною формулою

$$D_m(\lambda) = (a_{mm} - \lambda)D_{m-1}(\lambda) - a_{m,m-1}a_{m-1,m}D_{m-2}(\lambda) \quad (5.18)$$

$$(m = \overline{1, n}).$$

Для початку розрахунку покладемо

$$D_{-1}(\lambda) = 0; \quad D_0(\lambda) = 1.$$

Розрахунки за цією формулою потребують істотно менше часу, а втім варто мати на увазі, що кратні корені потрібно вилучати.

**Приклад 5.6.** Побудуємо характеристичне рівняння задачі (5.14) *методом інтерполяції*. З цією метою знайдемо норму матриці  $A$  (5.15) за формулою (5.17)

$$\|A\|_c = 2 + 1 = 3.$$

Оскільки вузли інтерполяції рекомендовано розміщувати рівномірно, покладемо

$$\lambda^{(0)} = -3; \quad \lambda^{(1)} = 0; \quad \lambda^{(2)} = 3,$$

обчислимо визначники  $D(\lambda^{(j)})$  і побудуємо таблицю поділених різниць (табл. 5.1).

Таблиця 5.1

$\lambda^{(j)}$	$D(\lambda^{(j)})$	1 ПР	2 ПР
-3	19	-6	1
0	1	0	
3	1		

Скориставшись нижнім скісним рядком табл. 5.1, побудуємо інтерполяційний поліном Ньютона, а отже, характеристичне рівняння:

$$D(\lambda) = 1 + (\lambda - 3) \times 0 + (\lambda - 3) \times \lambda \times 1 = \lambda^2 - 3\lambda + 1 = 0.$$

Це рівняння збігається із характеристичним рівнянням, знайденим у прикл. 5.5, чого і слід було очікувати.

Реалізацію розрахунків за методом інтерполяції див. в дод. 24.

До речі, у цьому разі характеристичне рівняння можна побудувати і за формулою (5.18). Тут

$$D_{-1}(\lambda) = 0, \quad D_0(\lambda) = 1, \quad D_1(\lambda) = 2 - \lambda.$$

Отже,

$$D_2(\lambda) = (1 - \lambda)(2 - \lambda)1 - (-1)(-1) = \lambda^2 - 3\lambda + 1.$$

#### 5.4.2. Часткова проблема власних значень

У прикладних задачах здебільшого інтерес становить лише невелика частина власних значень, причому інколи з фізичних міркувань можна знайти їхні нульові наближення. Методи розв'язання таких задач ітераційні.

**Метод лінеаризації.** Запишемо (5.11) у такому вигляді:

$$F(x, \lambda) = 0 \quad \text{або} \quad F_i(x, \lambda) \equiv \sum_{j=1}^n a_{ij} x_j - \lambda x_i = 0 \quad (i = \overline{1, n}).$$

Це система  $n$  алгебричних рівнянь відносно  $n+1$  невідомих  $\lambda, x_i$  ( $i = \overline{1, n}$ ), причому вона нелінійна, оскільки до неї входить добуток  $\lambda x_i$ .

Скористаємося для її розв'язання методом Ньютона. З цією метою надамо малі прирости невідомим  $x$  і  $\lambda$ , тобто перепишемо систему так:

$$F(x + \delta x, \lambda + \delta \lambda) = 0$$

або

$$\sum_{j=1}^n a_{ij}(x_j + \delta x_j) - (\lambda + \delta \lambda)(x_i + \delta x_i) = 0 \quad (i = \overline{1, n}).$$

Лінеаризуючи цю систему (тобто нехтуючи добутком  $\delta \lambda \delta x_i$ ), можна побудувати такий ітераційний процес:

$$\sum_{j=1}^n a_{ij} \delta x_j - \lambda^{(s)} \delta x_i - x_i^{(s)} \delta \lambda = -F_i(x^{(s)}, \lambda^{(s)}) \quad (5.19)$$

$$(i = \overline{1, n}; s = 0, 1, \dots),$$

де  $s$  — номер ітерації, причому його біля  $\delta \lambda$  та  $\delta x_i$  для скорочення опущено.

Оскільки власний вектор  $x$  завжди знаходиться з точністю до множника,  $\delta x_1$  (або  $\delta x_n$ ) можна прирівняти до нуля. Після цього в системі (5.19) кількість рівнянь і невідомих збігається, тобто система стає сумісною й може бути розв'язана методом Гаусса.

Після знаходження приростів  $\delta x_i, \delta \lambda$  невідомі уточнюють за такими формулами:

$$\lambda^{(s+1)} = \lambda^{(s)} + \delta \lambda; \quad x_j^{(s+1)} = x_j^{(s)} + \delta x_j;$$

$$(x_1^{(0)} = 1; \delta x_1 = 0; j = \overline{2, n}; s = 0, 1, \dots).$$

Оскільки процес Ньютона збігається квадратично, то за вдалого вибору нульового наближення шуканий розв'язок істотно уточнюється за 3 – 5 ітерацій; якщо ж за 10 ітерацій процес не збігається, потрібно замінити нульове наближення.

Цим методом успішно користуються для  $n \approx 100 \dots 1000$ . Він особливо зручний, якщо матриця тридіагональна.

**Приклад 5.7.** Розглянемо визначення найменшого власного значення  $\lambda_2$  у прикл. 5.5 і відповідного йому власного вектора  $x$  *методом лінеаризації*, поклавши

$$\lambda_2^{(0)} = \frac{1}{2}, \quad x_2^{(0)} = \frac{3}{2}, \quad x_1^{(0)} = 1, \quad \delta x_1 = 0,$$

тобто  $x_1^{(0)}$  не уточнюється під час ітерування.

Запишемо систему (5.19) для матриці  $A$  другого порядку загального вигляду, опустивши для спрощення верхній індекс  $s$  при  $x_1$ ,  $x_2$  і  $\lambda$ , а також нижній індекс 2 при  $\lambda$ :

$$a_{11}\delta x_1 + a_{12}\delta x_2 - \lambda\delta x_1 - x_1\delta\lambda = -[(a_{11} - \lambda)x_1 + a_{12}x_2];$$

$$a_{21}\delta x_1 + a_{22}\delta x_2 - \lambda\delta x_2 - x_2\delta\lambda = -[a_{21}x_1 + (a_{22} - \lambda)x_2].$$

Перепишемо цю систему, підставивши в неї значення  $a_{ij}$ , позначивши  $\delta x_2 = \delta x$  та врахувавши, що  $x_1^{(s)} = 1$ ,  $\delta x_1 = 0$ . Одержимо таку СЛАР відносно  $\delta x$ ,  $\delta\lambda$ :

$$\left. \begin{aligned} \delta x + \delta\lambda &= [(2 - \lambda) - x_2] \\ (1 - \lambda)\delta x - x_2\delta\lambda &= [1 - (1 - \lambda)x_2] \end{aligned} \right\} \quad (5.20)$$

Підставлення нульових наближень  $\lambda^{(0)}$ ,  $x_2^{(0)}$  у (5.20) приведе до такої СЛАР відносно приростів  $\delta\lambda$  і  $\delta x$  на першій ітерації:

$$\left. \begin{aligned} \delta x + \delta\lambda &= \left[ \left( 2 - \frac{1}{2} \right) - \frac{3}{2} \right] = 0 \\ \left( 1 - \frac{1}{2} \right) \delta x - \frac{3}{2} \delta\lambda &= \left[ 1 - \left( 1 - \frac{1}{2} \right) \frac{3}{2} \right] = \frac{1}{4} \end{aligned} \right\}$$

Звідси знайдемо значення приростів складових власного вектора та власного значення:

$$\delta x = \frac{1}{8}; \quad \delta\lambda = -\frac{1}{8}.$$

Отже,

$$\lambda^{(1)} = \frac{1}{2} - \frac{1}{8} = \frac{3}{8} = 0,375;$$

$$x_2^{(1)} = \frac{3}{2} + \frac{1}{8} = \frac{13}{8} = 1.625.$$

Аналогічно обчислимо наступне наближення:

$$\begin{aligned} \delta x + \delta \lambda &= \left[ \left( 2 - \frac{3}{8} \right) - \frac{13}{8} \right] \equiv 0 \\ \left( 1 - \frac{3}{8} \right) \delta x - \frac{13}{8} \delta \lambda &= \left[ 1 - \left( 1 - \frac{3}{8} \right) \frac{13}{8} \right] \equiv -\frac{1}{64}. \end{aligned}$$

Звідси

$$\delta x = -\frac{1}{144}; \quad \delta \lambda = \frac{1}{144}.$$

Отже,

$$\begin{aligned} \lambda^{(2)} &= \frac{3}{8} + \frac{1}{144} = \frac{55}{144} = 0.3819; \\ x_2^{(2)} &= \frac{13}{8} - \frac{1}{144} = \frac{233}{144} = 1.6180. \end{aligned}$$

Як видно з одержаного результату, друге наближення практично збігається з точним розв'язком характеристичного рівняння, що підтверджує квадратичну збіжність методу лінеаризації.

**Степеневий метод.** Якщо потрібно визначити найбільше за модулем власне значення  $\lambda_1$  ( $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots$ ), яке є простим коренем характеристичного рівняння (5.13), можна скористатися таким ітераційним процесом:

$$y^{(s+1)} = Ay^{(s)} \quad (s = 0, 1, \dots), \quad (5.21)$$

де  $y^{(0)}$  — довільний вектор.

Розкладемо  $y^{(0)}$  по власним векторам  $x^i$  матриці  $A$ , тобто запишемо

$$y^{(0)} = \sum_i \xi_i x^i.$$

Тоді

$$y^{(1)} = Ay^{(0)} = A \sum_i \xi_i x^i = \sum_i \xi_i Ax^i = \sum_i \lambda_i \xi_i x^i.$$



Звідси

$$y^{(s)} = \sum_i \lambda_i^s \xi_i x^i.$$

За досить великого  $s$  можна записати

$$y^{(s)} = \lambda_1^s \xi_1 (x^1 + \dots),$$

тобто вектор  $y^{(s)}$  збігається з власним вектором  $x^1$  за напрямком.

Отже, маємо

$$y_i^{(s+1)} \approx \lambda_1 y_i^{(s)} \quad (i = \overline{1, n})$$

або

$$|\lambda_1| \approx \frac{\|y^{(s+1)}\|}{\|y^{(s)}\|}. \quad (5.22)$$

У практичних розрахунках ітерації припиняють за умови

$$|y_i^{(s+1)} - y_i^{(s)}| \leq \varepsilon \quad (i = \overline{1, n})$$

або умови, аналогічної умові (4.15).

Швидкість збіжності залежить від того, наскільки вдало вибрано нульовий вектор  $y^{(0)}$ . Якщо він близький за напрямком до  $x^1$ , то ітерації швидко збігаються.

Для запобігання переповнення або втрати цифр під час розрахунків на ЕОМ потрібно у нормувати на кожній ітерації, тобто після обчислення  $\lambda_1$  кожену складову вектора  $y$  слід поділити на його норму:

$$\tilde{y}_j^{(s+1)} = \frac{y_j^{(s+1)}}{\|y^{(s+1)}\|}.$$

Замінивши у цьому ітераційному процесі матрицю  $A$  на  $A^{-1}$ , знайдемо найменше за модулем власне значення матриці  $A$  (воно є найбільше власне значення оберненої матриці

$A^{-1}$ ), бо власні значення матриць  $A$  та  $A^{-1}$  обернені стосовно одне одного.

**Приклад 5.8.** Знайдемо менше власне значення матриці  $A$  (5.15), оскільки в теорії коливань інтерес становить передусім найменша власна частота.

Для застосування степеневому методу потрібно знайти обернену матрицю  $A^{-1}$ , скориставшись, наприклад, формулою (5.10), для чого слід розв'язати дві такі СЛАР:

$$\begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} \alpha_{11} \\ \alpha_{21} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \quad \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} \alpha_{12} \\ \alpha_{22} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Розв'язавши ці системи, знайдемо обернену матрицю

$$B \equiv A^{-1} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}.$$

Візьмемо за нульове наближення вектор  $y^{(0)} = (1, 0)$  і одержуватимемо наступні за формулою (5.21), тобто

$$y^{(1)} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}; \quad y^{(2)} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \text{ і т.д.}$$

Значення складових вектора  $y^{(i)}$  та параметра  $\lambda_1^B$  для декількох ітерацій наведено в табл. 5.2.

Таблиця 5.2

$y^{(1)}$	$y^{(2)}$	$y^{(3)}$	$y^{(4)}$	$\lambda_1^B$	$y^{(5)}$	$\lambda_1^B$	$y^{(6)}$	$\lambda_1^B$	$x^1$
1	2	5	13	2,600	34	2,615	89	2,618	1,000
1	3	8	21	2,625	55	2,619	144	2,618	1,618

Якщо складові  $x^1$  пронормувати за Чебишовим, тобто поділити на абсолютну величину найбільшої його складової, то він стане таким:  $x^1 = (0,618; 1,000)$ .

Оскільки власні значення матриць  $A$  та  $A^{-1}$  обернені стосовно одне одного, найменше власне значення  $\lambda_2^A$  матриці  $A$  знайдемо так:

$$\lambda_2^A = \frac{1}{\lambda_1^B} = \frac{1}{2,618} = 0,382.$$

Реалізацію розрахунків за степеневим методом наведено у дод. 25.



### Задачі для самостійного розв'язування

1. Знайдіть розв'язок СЛАР та визначник її матриці методом Гаусса

$$\begin{pmatrix} 3 & -1 & 2 \\ 1 & 2 & 3 \\ 2 & -2 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 12 \\ 11 \\ 2 \end{pmatrix}.$$

2. Знайдіть обернену матрицю  $\begin{pmatrix} 1 & 1 & 3 \\ -1 & 0 & 3 \\ 2 & 1 & 1 \end{pmatrix}$  методом Гаусса.

3. Skorиставшись методом Гаусса — Зейделя, розв'яжіть СЛАР

$$\begin{pmatrix} 8 & 1 & -1 \\ 2 & 1 & 9 \\ 1 & -1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 8 \\ 12 \\ 2 \end{pmatrix}.$$

4. Визначте власні значення та їхні власні вектори для матриці  $\begin{pmatrix} 2 & -1 & 3 \\ -2 & 4 & 5 \\ 3 & 2 & -1 \end{pmatrix}$ , у тім числі найбільше власне значення та відповідний йому власний вектор описаними ітераційними методами.

## Розділ 6 ЗВИЧАЙНІ ДИФЕРЕНЦІАЛЬНІ РІВНЯННЯ

---

Математичними моделями прикладних задач безпосередньо або після відокремлення змінних є задачі для звичайних диференціальних рівнянь. Причому це можуть бути звичайні диференціальні рівняння будь-якого порядку або їх системи.

Зазначимо, що звичайне диференціальне рівняння  $p$ -го порядку

$$u^{(p)}(x) = f(x, u, u', \dots, u^{(p-1)})$$

заміною  $u_k(x) = u^{(k)}(x)$  зводиться до тотожної йому системи  $p$  звичайних диференціальних рівнянь 1-го порядку:

$$u'_k(x) = u_{k+1}(x) \quad (k = \overline{0, p-2});$$

$$u'_{p-1}(x) = f(x, u_0, u_1, \dots, u_{p-1}) \quad [u_0 = u(x)].$$

Аналогічно можна замінити систему  $p$  звичайних диференціальних рівнянь 1-го порядку на одне рівняння  $p$ -го порядку, але при цьому не завжди виконується еквівалентність. Далі, розглядаючи той чи інший метод розв'язання звичайних диференціальних рівнянь, використовуватимемо найзручнішу форму запису звичайних диференціальних рівнянь для його викладення.

Систему  $p$  звичайних диференціальних рівнянь 1-го порядку запишемо у такому вигляді:

$$u'_k(x) = f_k(x, u_1, u_2, \dots, u_p) \quad (k = \overline{1, p}) \quad \text{або} \quad u'(x) = f(x, u), \quad (6.1)$$

де  $u = (u_1, u_2, \dots, u_p)$ ;  $f = (f_1, f_2, \dots, f_p)$ .

З теорії звичайних диференціальних рівнянь відомо, що розв'язок  $u(x)$  звичайних диференціальних рівнянь  $p$ -го порядку в загальному випадку залежить від  $p$  довільних сталих  $C = (C_1, C_2, \dots, C_p)$ , тобто  $u = u(x; C)$ . Для визначення  $C$  (виді-

лення єдиного або потрібного нам розв'язку) на розв'язок слід накласти  $p$  додаткових умов.

У звичайних диференціальних рівняннях залежно від вигляду додаткових умов розглядають такі три основні типи задач: задача Коші, крайова задача та задача на власні значення.

### 6.1. Задача Коші

Тут усі додаткові умови, тобто значення всіх функцій  $u_k(x)$ , задаватимемо в одній точці  $x = \xi$ , яку називають *початковою*:

$$u_k(\xi) = \eta_k \quad (k = \overline{1, p})$$

або

$$u(\xi) = \eta \quad [\eta = (\eta_1, \eta_2, \dots, \eta_p)]. \quad (6.2)$$

Самі ж додаткові умови називатимемо *початковими умовами*. Їх можна вважати координатами початкової точки  $(\xi, \eta_1, \eta_2, \dots, \eta_p)$  інтегральної кривої  $u(x)$  у  $p + 1$ -вимірному просторі  $(x, u_1, u_2, \dots, u_p)$ .

Розв'язок у цьому разі визначатимемо на деякому відрізку  $[\xi, X]$  або  $[X, \xi]$ . Тому точку  $\xi$  можна вважати початковою точкою цього відрізка.

Якщо права частина (6.1)  $f(x, u)$  задовольняє умову Ліпшиця за змінними  $u_k$ , то розв'язок задачі Коші (6.1), (6.2) єдиний і неперервно залежить від координат початкової точки, тобто задача *коректно поставлена*. Якщо, крім того, права частина має неперервні похідні за всіма аргументами до  $q$ -го порядку включно, то розв'язок  $u(x)$  має  $q + 1$  похідну за змінною  $x$ .

Умовно методи розв'язання задачі Коші поділятимемо на точні, наближені та числові. У *точних методах* розв'язок можна виразити через елементарні функції або через квадратури від них. Зрозуміло, що знайдений точний розв'язок легко застосовувати та проводити його якісний аналіз, але звичайні диференціальні рівняння, для яких розроблені точні методи розв'язання, охоплюють лише малу частину задач, які трапляються на практиці. До *наближених методів* належать такі, в яких розв'язок  $u(x)$  одержують як межу деякої послідовності функцій  $y_n(x)$ , причому  $y_n(x)$  виражають через елементарні функції або через квадратури від них. Об-

межуючись скінченною кількістю членів, знайдемо наближений вираз для  $u(x)$ . Відомим прикладом цих методів є розкладання розв'язку в ряд Тейлора або Маклорена. Ці методи зручні, якщо більшу частину перетворень можна виконати аналітично. В основному їх застосовують для розв'язування лінійних задач, що зазвичай дуже звужує межі їх використання.

Значно універсальнішими методами розв'язування задачі Коші є *числові*, які дають змогу визначити наближені значення розв'язку  $u(x)$  на довільній сітці значень аргументу  $x_n$  ( $n = \overline{0, N}$ ) відрізка  $[\xi, X]$ , тобто  $\xi = x_0 < x_1 < \dots < x_N = X$ .

Зрозуміло, що вони не дають змоги знайти загальний розв'язок задачі Коші, а лише якийсь окремих розв'язок, однак їх можна використовувати для розв'язування істотно ширшого кола звичайних диференціальних рівнянь, ніж у разі точних чи наближених методів. При цьому задачі Коші має бути *коректно поставленою та добре обумовленою*, тобто такою, щоб мала зміна початкових умов приводила до малих змін розв'язку.

### 6.1.1. Наближені методи

**Метод Пікара.** Розглянемо таку задачу Коші:

$$u'(x) = f(x, u(x)) \quad [\xi \leq x \leq X]; \quad (6.3)$$

$$u(\xi) = \eta. \quad (6.4)$$

Інтегруючи диференціальне рівняння, задачу Коші замінимо на тотожну їй задачу для інтегрального рівняння Вольterra II роду:

$$u(x) = \eta + \int_{\xi}^x f(t, u(t)) dt.$$

Його можна розв'язати за допомогою такого ітераційного процесу:

$$y^{(0)}(x) = \eta, \quad y^{(s+1)}(x) = \eta + \int_{\xi}^x f(t, y^{(s)}(t)) dt \quad (s = 0, 1, \dots), \quad (6.5)$$

де  $y^{(s)}(x)$  — наближений розв'язок, причому за нульове наближення можна взяти будь-яку відому функцію, але, як правило, беруть його початкове значення  $\eta$ .

На кожному кроці інтегрування виконують точними або числовими методами. Можна довести, що *наближений розв'язок рівномірно збігається до точного в області  $G(x,u)$ , якщо права частина рівняння  $f(x,u)$  неперервна і задовольняє умову Ліпшиця за змінною  $u$ .*

Знайдені наближення збігаються до розв'язку в інтервалі  $(\xi - \delta, \xi + \delta)$ , де  $\delta$  — найменше з чисел  $A$  і  $\frac{B}{M}$ , причому  $A$  і  $B$  — довільні додатні числа, для яких  $\xi - A \leq x \leq \xi + A$ ,  $\eta - B \leq u \leq \eta + B$ ,  $M$  — стала, що обмежує значення функції  $f(x,u)$  у цій області, тобто  $|f(x,u)| \leq M$ .

Розглянутий метод доцільно застосовувати, якщо квадратури можна знайти точно або виразити через елементарні функції. Якщо ж інтеграли знаходити числово, то він стає не зовсім зручним.

Метод Пікара легко поширити на системи звичайних диференціальних рівнянь.

**Метод малого параметра (асимптотичний, збурень)** запропонував у 1892 р. А. Пуанкаре. Нехай права частина звичайного диференціального рівняння (6.3) залежить від малого параметра  $\varepsilon$  або ж його вводять до правої частини штучно так, щоб за  $\varepsilon = 0$  допоміжне звичайне диференціальне рівняння істотно спрощувалось, а за  $\varepsilon = 1$  перетворювалося на вихідне.

Розглянемо задачу Коші такого вигляду:

$$u'(x) = f(x, u; \varepsilon); \quad u(\xi) = \eta \quad [\xi \leq x \leq X]. \quad (6.6)$$

Шукатимемо її розв'язок у вигляді ряду по  $\varepsilon$ :

$$u(x) = \sum_{m=0}^{\infty} \varepsilon^m z_m(x). \quad (6.7)$$

Підставивши формально цей ряд у (6.6), отримаємо

$$\sum_{m=0}^{\infty} \varepsilon^m z'_m(x) = f\left(x, \sum_{m=0}^{\infty} \varepsilon^m z_m(x)\right), \quad u(\xi) = \eta.$$

Щоб знайти невідомі функції  $z_m(x)$ , потрібно розв'язати ланцюг звичайних диференціальних рівнянь, застосувавши

метод невизначених коефіцієнтів, тобто прирівнявши вирази лівої та правої частин за однакових степенів  $\varepsilon$ .

При цьому початковою умовою  $u(\xi) = \eta$  скористаємося лише для знаходження  $z_0(x)$ , тобто тут  $z_0(\xi) = \eta$ . В усіх інших випадках слід брати  $z_m(\xi) = 0$ , щоб не змінити початкові умови вихідної задачі.

Метод малого параметра, як і метод Пікара, неважко поширити на системи звичайних диференціальних рівнянь, але відтак усі викладки стануть істотно громіздкішими.

**Приклад 6.1.** Розглянемо застосування наведених наближених методів до розв'язування такої задачі Коші:

$$u'(x) = x^2 + u^2; \quad u(0) = 0. \quad (A)$$

1. У разі застосування *методу Пікара* квадратури в (6.5) можна знайти точно, тобто одержати таку послідовність наближених розв'язків :

$$y^{(0)}(x) = 0; \quad y^{(1)}(x) = \frac{x^3}{3}; \quad y^{(2)}(x) = \left(1 + \frac{x^4}{21}\right) \frac{x^3}{3};$$

$$y^{(3)}(x) = \left(1 + \frac{x^4}{21} + 2 \frac{x^8}{693} + \frac{x^{12}}{19\,845}\right) \frac{x^3}{3}, \dots$$

Видно, що для  $x \leq 1$  ці наближення швидко збігаються, тому наближений розв'язок можна обчислити з великою точністю.

2. Розглянемо тепер застосування *методу малого параметра* до розв'язування задачі Коші (A). З цієї метою введемо  $\varepsilon$  у праву частину, тобто будемо розв'язувати таку задачу Коші:

$$u'(x) = x^2 + \varepsilon u^2; \quad u(0) = 0. \quad (B)$$

Якщо покласти  $\varepsilon = 0$ , матимемо дуже просту задачу Коші:

$$u'_0(x) = x^2; \quad u_0(0) = 0, \quad \text{яка має розв'язок } u_0(x) = \frac{x^3}{3}.$$

Для одержання наближеного розв'язку задачі Коші (B) підставимо в нього ряд (6.7):



$$\sum_{m=0}^{\infty} \varepsilon^m z'_m(x) = \varepsilon \left[ \sum_{m=0}^{\infty} \varepsilon^m z_m(x) \right]^2 + x^2; \quad u(0) = 0.$$

Далі, прирівнявши доданки лівої і правої частин за однакових степенів  $\varepsilon$ , отримуємо такий ланцюг задач Коші для знаходження  $z_m(x)$ :

$$\begin{aligned} \varepsilon^0 \mid z'_0(x) &= x^2; & z_0(0) &= 0. \rightarrow z_0(x) = \frac{x^3}{3}; \\ \varepsilon^1 \mid z'_1(x) &= z_0^2(x) \equiv \frac{x^6}{9}; & z_1(0) &= 0. \rightarrow z_1(x) = \frac{x^7}{63}; \\ \varepsilon^2 \mid z'_2(x) &= 2z_0(x)z_1(x) \equiv \frac{2x^{10}}{189}; & z_2(0) &= 0. \rightarrow z_2(x) = \frac{2x^{11}}{2079}; \\ & \dots & & \dots \end{aligned}$$

Отже, якщо покласти  $\varepsilon = 1$  та обмежитись одержаними наближеннями, наближений розв'язок задачі Коші (A) набуде такого вигляду:

$$y(x) = \sum_{m=0}^{\infty} z_m(x) = \left( 1 + \frac{x^4}{21} + 2 \frac{x^8}{693} \right) \frac{x^3}{3},$$

тобто наближений розв'язок збігається з трьома першими доданками наближеного розв'язку за методом Пікара.

**Приклад 6.2.** Знайдемо методом Пікара розв'язок задачі Коші для такої системи звичайних диференціальних рівнянь:

$$\begin{aligned} \frac{du_1}{dx} &= 5u_1(x) + 4u_2(x), & u_1(0) &= 1; \\ \frac{du_2}{dx} &= 4u_1(x) + 5u_2(x), & u_2(0) &= -1. \end{aligned}$$

Поширивши дії (6.5) на систему звичайних диференціальних рівнянь, отримаємо

$$\begin{cases} u_1(x) = 1 + \int_0^x (5u_1(t) + 4u_2(t)) dt; \\ u_2(x) = -1 + \int_0^x (4u_1(t) + 5u_2(t)) dt; \end{cases}$$

$$\left\{ \begin{aligned} y_1^{(0)}(x) = 1, \quad y_1^{(1)}(x) = 1 + \int_0^x 1 dt = 1 + x; \\ y_2^{(0)}(x) = -1, \quad y_2^{(1)}(x) = -1 + \int_0^x (-1) dt = -(1 + x); \\ y_1^{(2)}(x) = 1 + \int_0^x (5(1+t) - 4(1+t)) dt = 1 + x + \frac{x^2}{2!}; \\ y_2^{(2)}(x) = -1 + \int_0^x (4(1+t) - 5(1+t)) dt = -\left(1 + x + \frac{x^2}{2!}\right); \\ \dots \\ y_1^{(s+1)}(x) = \left(1 + x + \frac{x^2}{2!} + \dots + \frac{x^s}{s!}\right) \xrightarrow{s \rightarrow \infty} e^x; \\ y_2^{(s+1)}(x) = -\left(1 + x + \frac{x^2}{2!} + \dots + \frac{x^s}{s!}\right) \xrightarrow{s \rightarrow \infty} -e^x. \end{aligned} \right.$$

Отже, шуканим розв'язком розглянутої задачі є сукупність функцій  $u_1(x) = e^x$  і  $u_2(x) = -e^x$ , причому він єдиний за теоремою Коші існування і єдиності розв'язку.

### 6.1.2. Числові методи

Розглянемо застосування числових методів на прикладі задачі Коші (6.3), (6.4), для чого виберемо на відрізку  $[\xi, X]$  сітку  $x_n$  ( $n = \overline{0, N}$ ) значень аргументу ( $\xi = x_0 < x_1 < \dots < x_N = X$ ). Сітка може бути нерівномірною.

Якщо розв'язок  $u(x)$  розкласти у ряд Тейлора на інтервалі  $x_n < x < x_{n+1}$  й позначити  $u(x_n) = u_n$ , отримаємо

$$u_{n+1} = u_n + h_n u'_n + h_n^2 \frac{u''_n}{2!} + \dots \quad (h_n = x_{n+1} - x_n). \quad (6.8)$$

Похідні в правій частині можна легко одержати, продиференціювавши вираз (6.3) потрібне число разів:

$$\begin{aligned} u' = f(x, u); \quad (u')' = \frac{d}{dx}[f(x, u(x))] = f_x + f u \\ \left( f_x = \frac{\partial f}{\partial x}; \quad f_u = \frac{\partial f}{\partial u} \right). \end{aligned} \quad (6.9)$$

Загалом, якщо  $f(x, u)$  має  $q$  неперервних похідних за аргументами  $x$  та  $u$ , тоді в (6.8) можна утримувати члени до  $O(h^{q+1})$  включно. Проте застосовувати для розрахунків формулу (6.8) з великим числом членів незручно навіть для порівняно простих правих частин, бо вирази для похідних можуть бути громіздкими, а якщо праві частини відомі лише наближено, то знаходження їх похідних із застосуванням поділених різниць узагалі небажане.

**Метод ламаних.** Це найпростіший числовий метод, який на практиці для обчислень застосовують досить рідко через невелику точність, але за його допомогою легко пояснити способи побудови й дослідження числових методів.

Залишивши у виразі (6.8) лише перші два доданки, одержимо розрахункову формулу методу ламаних (Ейлера) розв'язання задачі Коші для звичайного диференціального рівняння 1-го порядку:

$$y_{n+1} = y_n + h_n f(x_n, y_n); \quad y_0 = \eta \quad (6.10)$$

$$\left[ n = \overline{0, N-1}; u(x_n) \approx y_n \right].$$

Як і раніше, позначатимемо наближені значення розв'язку  $u_n$  через  $y_n$ .

Для числових розрахунків за методом ламаних досить знати початкове значення  $y_0 = \eta$  (*самостартовність*), бо всі інші значення  $y_1, y_2, y_3, \dots, y_N$  знаходять за формулою (6.10).

Геометрично вираз (6.10) означає заміну руху по інтегральній кривій на рух по дотичній до неї.

Отже, на кожному кроці траєкторія руху буде відрізком прямої, а на всьому відрізку інтегрування вона матиме вигляд ламаної (звідси і назва метод ламаних).

Можна довести, що за  $h \rightarrow 0$  числовий розв'язок збігається на обмеженому відрізку  $|x - x_0| < a$  до точного рівномірно з 1-м порядком точності, тобто з точністю до  $O(h)$ , а в разі нерівномірного кроку — до  $O(h_{\max})$ . Слід лише мати на увазі, що ця оцінка дуже завищена.

Якщо вихідна задача *погано обумовлена*, то числовий розв'язок може дуже різнитися від точного, оскільки відомо, що розв'язок звичайних диференціальних рівнянь зі сталими коефіцієнтами має експоненціальний характер.

Для системи звичайних диференціальних рівнянь (6.1) формула (6.10) набуває вигляду

$$y_k^{n+1} = y_k^n + h_n f_k(x_n, y_1^n, \dots, y_p^n); \quad y_k^0 = \eta_k \quad (6.11)$$

$$\left[ n = \overline{0, N-1}; u_k(x_n) \approx y_k^n; k = \overline{1, p} \right].$$

Тут і далі  $y_k^n$  — значення  $k$ -ї складової числового розв'язку в точці  $x_n$ .

**Методи Рунге — Кутти.** Ефективнішими для розрахунків є методи Рунге — Кутти, які уможливають одержання розрахункових формул з різними порядками точності. Найпростішою з них є формула 2-го порядку точності, вираз якої можна одержати, якщо у формулі (6.8) утримати члени порядку  $O(h^2)$ . Щоб запобігти диференціюванню  $f(x, u)$ , замінимо другу похідну  $u''$  на першу поділену різницю від неї, тобто

$$u'' = \frac{d}{dx} f(x, u) \approx (f(\tilde{x}, \tilde{u}) - f(x, u)) / \Delta x.$$

відповідним чином добираючи  $\tilde{x}$ ,  $\tilde{u}$ ,  $\Delta x$ . Тоді після підстановки цього виразу для  $u''$  у (6.8) і зведення подібних дістанемо такий вираз:

$$y_{n+1} = y_n + h_n [\beta f(x_n, y_n) + \alpha f(x_n + \gamma h_n, y_n + \delta h_n)], \quad (6.12)$$

де  $\alpha, \beta, \gamma, \delta$  — невідомі коефіцієнти, які потрібно визначити.

З цією метою функцію  $f(\cdot)$  перед  $\alpha$  розкладемо у ряд Тейлора за степенями  $h$  (індекс  $n$  при ньому далі упустимо), унаслідок чого одержимо

$$f(\cdot) = f(x_n, y_n) + h(\gamma f_x + \delta f_u) + \dots$$

Тоді формулу (6.12) можна записати так:

$$y_{n+1} = y_n + h(\alpha + \beta)f(x_n, y_n) + \alpha h^2(\gamma f_x + \delta f_u)_n. \quad (6.13)$$

Виберемо параметри  $\alpha, \beta, \gamma, \delta$  так, щоб вираз (6.13) якомога краще передавав (6.8). З їх порівняння випливає, що

$$\alpha + \beta = 1; \quad \alpha\gamma = \frac{1}{2}; \quad \alpha\delta = \frac{f_n}{2} \quad (f_n = f(x_n, y_n)).$$

Отже, для визначення цих чотирьох параметрів маємо лише три рівняння, а це означає, що один із них залишається вільним. Якщо всі параметри в (6.12) виразити через  $\alpha$ , ма-

тимемо однопараметричну сім'ю двочленних різницевих схем методу Рунге — Кутти ( $0 < \alpha \leq 1$ ):

$$y_{n+1} = y_n + h \left[ (1-\alpha)f_n + \alpha f \left( x_n + \frac{h}{2\alpha}, y_n + \frac{h}{2\alpha} f_n \right) \right]. \quad (6.12')$$

Можна довести, що за  $h \rightarrow 0$  числовий розв'язок у збігається до точного розв'язку  $u(x)$  рівномірно з 2-м порядком точності [у разі нерівномірного кроку  $O(h_{\max}^2)$ ], якщо  $f(x,u)$  неперервна й обмежена разом зі своїми двома похідними за обома аргументами.

Ця схема має непогану точність, її здебільшого застосовують у числових розрахунках, беручи  $\alpha = 1$  або  $\alpha = \frac{1}{2}$ :

$$y_{n+1} = y_n + hf \left( x_n + \frac{h}{2}, y_n + \frac{hf_n}{2} \right) \quad (\alpha = 1); \quad (6.14)$$

$$y_{n+1} = y_n + \frac{h}{2} [f_n + f(x_n + h, y_n + hf_n)] \quad \left( \alpha = \frac{1}{2} \right). \quad (6.15)$$

Зрозуміло, що початкова умова в обох випадках така:

$$y_0 = \eta \quad [n = \overline{0, N-1}; u(x_n) \approx y_n].$$

Посилаючись на цей метод, зазначатимемо значення  $\alpha$ .

Формули методу Рунге — Кутти легко поширити на розв'язання систем звичайних диференціальних рівнянь.

Розрахункові формули методу Рунге — Кутти 2-го порядку ( $\alpha = 1$ ) для системи звичайних диференціальних рівнянь 2-го порядку мають такий вигляд:

$$\begin{aligned} y_1^{n+1} &= y_1^n + hf_1 \left( x_n + \frac{h}{2}, y_1^n + h \frac{f_1^n}{2}, y_2^n + h \frac{f_2^n}{2} \right); \\ y_2^{n+1} &= y_2^n + hf_2 \left( x_n + \frac{h}{2}, y_1^n + h \frac{f_1^n}{2}, y_2^n + h \frac{f_2^n}{2} \right), \end{aligned} \quad (6.16)$$

де

$$y_k^0 = \eta_k \quad [n = \overline{0, N-1}; u_k(x_n) \approx y_k^n; f_k^n = f_k(x_n, y_1^n, y_2^n); k = \overline{1, 2}].$$

Як уже зазначалось, методи Рунге — Кутти можуть бути різних порядків. У більшості вбудованих функцій для

розв'язування звичайних диференціальних рівнянь на ЕОМ реалізовано схему 4-го порядку точності:

$$y_{n+1} = y_n + (q_1 + 2q_2 + 2q_3 + q_4)h/6, \quad (6.17)$$

де

$$\begin{aligned} q_1 &= f(x_n, y_n); & q_2 &= f\left(x_{n+1/2}, y_n + \frac{h}{2}q_1\right); \\ q_3 &= f\left(x_{n+1/2}, y_n + \frac{h}{2}q_2\right); & q_4 &= f(x_{n+1}, y_n + hq_3). \end{aligned}$$

Основні переваги методу Рунге — Кутти: досить висока точність (крім метода ламаних), явність ( $y_{n+1}$  знаходять за обмежене число дій), можливість застосування змінного кроку та самостартовність.

Деяке уявлення про похибки різницевих схем методів Рунге — Кутти можна одержати, коли права частина має вигляд  $f(x)$ , тобто в ній відсутня невідома функція  $u$ : тоді формули методу Рунге — Кутти 2-го порядку за (6.14) і (6.15), тобто  $\alpha = 1$  і  $\alpha = \frac{1}{2}$ , переходять відповідно у квадратурні формули середніх та трапецій із кроком  $h$ , а формула методу Рунге — Кутти 4-го порядку (6.17) — у формулу парабол із кроком  $\frac{h}{2}$ . Мажорантні оцінки похибок цих формул на рівномірній сітці із зазначеними кроками відповідно дорівнюють:

$$\begin{aligned} R^{(2)}\Big|_{\alpha=1} &\leq (b-a)h^2 \max|f''|/24; \\ R^{(2)}\Big|_{\alpha=1/2} &\leq - (b-a)h^2 \max|f''|/12; & (6.18) \\ R^{(4)} &\leq - (b-a)h^4 \max|f^{IV}|/2880. \end{aligned}$$

З огляду на невеликі значення числових коефіцієнтів у цих формулах впливає, що методи Рунге — Кутти можуть забезпечити досить високу точність розрахунків.

**Метод Адамса.** Цей метод є багатокроковий, оскільки для визначення  $y_{n+1}$  використовують значення на декількох попередніх кроках.

Розглядатимемо праву частину рівняння (6.3) не на всій площині зміни її аргументів  $x, u$ , а лише на визначеній інтегральній кривій  $u(x)$ , що відповідає шуканому розв'язку. Тоді

вона буде функцією лише аргументу  $x$ : позначимо її через  $F(x) = f(x, u(x))$ .

Якщо значення числового розв'язку відомі в декількох вузлах сітки  $y_n, y_{n-1}, \dots, y_{n-m}$  то, очевидно, у цих точках також відомі  $F(x_k) = f(x_k, y_k)$  ( $k = \underline{n, n-m}$ ). В околі цих вузлів  $F(x)$  замінюють на інтерполяційний поліном Ньютона, який у разі нерівномірної сітки матиме вигляд

$$\begin{aligned}
 F(x) = & F(x_n) + (x - x_n)F(x_n, x_{n-1}) + \\
 & + (x - x_n)(x - x_{n-1})F(x_n, x_{n-1}, x_{n-2}) + \\
 & + (x - x_n)(x - x_{n-1})(x - x_{n-2})F(x_n, x_{n-1}, x_{n-3}) + \dots \quad (6.19)
 \end{aligned}$$

Обмежившись лише записаними членами, можна одержати різницеву формулу Адамса 4-го порядку звичайних диференціальних рівнянь точності. Для обчислення розв'язку в наступній точці  $x_{n+1}$  звичайних диференціальних рівнянь запишемо в інтегральній формі:

$$u_{n+1} = u_n + \int_{x_n}^{x_{n+1}} f(x, u(x))dx = u_n + \int_{x_n}^{x_{n+1}} F(x)dx.$$

Якщо в цю формулу підставити інтерполяційний поліном Ньютона (6.19), провести спрощувальні перетворення та проінтегрувати, матимемо формули Адамса 4-го порядку точності на нерівномірній сітці:

$$\begin{aligned}
 y_{n+1} = & y_n + h_n F(x_n) + h_n^2 F(x_n, x_{n-1})/2 + \\
 & + h_n^2 (2h_n + 3h_{n-1})F(x_n, x_{n-1}, x_{n-2})/6 + \\
 & + h_n^2 [3h_n^2 + 4h_n(2h_{n-1} + h_{n-2}) + \\
 & + 6h_{n-1}(h_{n-1} + h_{n-2})] F(x_n, x_{n-1}, x_{n-2}, x_{n-3})/12 \\
 & (h_n = x_{n+1} - x_n). \quad (6.20)
 \end{aligned}$$

Знехтувавши в цій формулі останнім доданком, отримаємо формулу 3-го порядку точності. Аналогічно можна одержати формули нижчих порядків. Формула 1-го порядку збігається з формулою ламаних.

Залежність (6.20) істотно спрощується, якщо сітка рівномірна, тобто  $h = \text{const}$ . Увівши замість поділених різниць скінченні різниці, одержимо таку формулу:

$$y_{n+1} = y_n + hF_n + \frac{h^2 \Delta^1 F_n}{2} + \frac{5h^3 \Delta^2 F_n}{12} + \frac{3h^4 \Delta^3 F_n}{8}, \quad (6.20')$$

залишковий член якої дорівнює  $\frac{251}{750}h^5 F^{(iv)}(x)$ .

Цей метод без будь-яких змін переноситься на системи звичайних диференціальних рівнянь 1-го порядку.

Оскільки для початку розрахунків за формулою (6.20) потрібно знати розв'язок у чотирьох точках  $x_0, x_1, x_2, x_3$ , то за допомогою будь-якого іншого методу слід обчислити відповідні  $y_n$ , що вдвічі збільшує обсяг програми. Крім того, лише для сталого кроку сітки формула Адамса проста.

На перший погляд здається, що метод Адамса вигідніший, ніж метод Рунге — Кутти такого самого порядку, оскільки на кожному кроці в ньому значення  $f(x, u)$  обчислюють один раз, а в методі Рунге — Кутти 4-го порядку — чотири рази. Проте зважаючи, що числовий коефіцієнт похибки формули методу Рунге — Кутти 4-го порядку в  $\sqrt[4]{960} \approx 6$  разів менший за похибку формули Адамса, зазначимо, що метод Рунге — Кутти уможливує знайти числовий розв'язок за менше число арифметичних дій, оскільки в ньому крок можна збільшити майже в 6 разів порівняно з кроком за методом Адамса.

Тому метод Адамса, як і подібні йому багатокрокові методи, нині досить рідко застосовують для практичних розрахунків.

**Спеціальні методи.** Якщо існує достатня кількість обмежених похідних від  $f(x, u)$ , найкращим методом розв'язання звичайних диференціальних рівнянь є метод Рунге — Кутти. Коли ж розв'язок швидко- або знакозмінний, цей метод потребує дуже малого кроку інтегрування для одержання прийнятної точності числового розв'язку.

Для таких задач розроблено спеціальні методи. У більшості з них для *вихідної задачі*

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

треба знайти таку *допоміжну*

$$v'(x) = g(x, v),$$

щоб розв'язок останньої якнайпростіше був виражений через елементарні функції і при цьому на значному проміжку зміни аргументу виконувалась умова

$$u(x) \approx v(x).$$



Розв'язок  $v(x)$  можна знайти наближеними методами (Пікара, асимптотичним та ін.) або ж розв'язанням вихідного звичайне диференціальне рівняння, спростивши його праву частину.

У разі *знакозмінних* розв'язків (наприклад, швидкоколивних) уведемо функцію

$$w(x) = u(x) - v(x)$$

і перейдемо до розв'язання такого рівняння:

$$w'(x) = f(x, v(x) + w(x)) - g(x, v(x)),$$

де  $v(x)$  — відома функція.

Якщо  $v(x)$  є дійсно добре наближення до  $u(x)$ , то  $w(x)$  невелика і її легко знайти за допомогою методу Рунге — Кутти.

Якщо ж розв'язок  $u(x)$  *швидкозмінний* (наприклад, росте за експоненціальним чи степеневим законом), уведемо допоміжну функцію

$$w(x) = \frac{u(x)}{v(x)}.$$

Тоді

$$w'(x) = (f(x, v(x)w(x)) - w(x)g(x, v(x)))/v(x),$$

де  $v(x)$  — відома функція.

Функцію  $w(x)$  знайдемо за допомогою методу Рунге — Кутти.

Зазвичай є й інші спеціальні методи, але всі вони можуть бути застосовні лише для відповідного вузького класу задач.

**Особливі точки.** Якщо  $f(x, u(x))$  або будь-яка її похідна прямує до нескінченності в будь-якій точці відрізка інтегрування, цю точку називають *особливою*. Для розв'язання таких задач можна вихідне рівняння за допомогою заміни змінної інтегрування перетворити на рівняння без особливих точок або побудувати спеціальну різницеву схему чи, знайшовши за допомогою методу Пікара розв'язок в околі цієї точки, далі послугуються стандартними методами, наприклад методом Рунге — Кутти.

**Про згущення сітки. Правило Рунге.** Основним способом оцінки точності числового розв'язку задачі Коші для звичайних диференціальних рівнянь є застосування правила Рунге.

Якщо сітка рівномірна  $h(x) = h = \text{const}$  і залишковий член схеми  $p$ -го порядку пропорційний  $h^p$ , то для уточнення розв'язку можна застосувати правило Рунге.

Якщо ж є числові розв'язки на двох сітках  $y(x;h)$  і  $y(x;gh)$  ( $g > 1$ ), то похибку розв'язку на сітці з меншим кроком можна знайти за такою формулою:

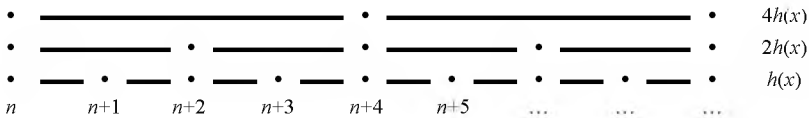
$$\Delta y(x;h) \approx \frac{y(x;h) - y(x;gh)}{g^p - 1}. \quad (6.21)$$

Замість оцінки точності знайдену похибку можна додати до числового розв'язку  $y(x;h)$ , тим самим уточнивши його:

$$\tilde{y}(x;h) \approx y(x;h) + \Delta y(x;h), \quad (6.22)$$

але тоді залишиться відкритим питання про оцінку точності  $\tilde{y}$ . Ці міркування справедливі й тоді, коли сітки з різною кількістю вузлів нерівномірні, але їх можна описати функціями  $h(x)$ , де при цьому  $\frac{h_I(x)}{h_{II}(x)} = g = \text{const}$  (наприклад, у разі квазі-рівномірної сітки).

Правило Рунге справедливе для будь-якого  $g$ , але найчастіше сітку згущують удвічі:



У вузлах, спільних для декількох сіток,  $y(x)$  можна уточнити за формулою (6.22). Так, у вузлі  $n$  точність можна підвищити на «дві одиниці», у вузлі  $n+2$  — на «одиницю», у вузлі  $n+1$  цього не можна зробити, тобто в цьому разі оцінку похибки за формулою (6.21) можна здійснити лише для парних вузлів.

Дещо ускладнивши розрахунки,  $u(x)$  можна уточнити в усіх вузлах найдокладнішої сітки. Наприклад, похибку на двох нижніх сітках (див. схему) можна обчислити за формулами

$$\Delta_m \approx \frac{y(x_m;h) - y(x_m;gh)}{g^p - 1} \quad (m = n, n+2, n+4, \dots);$$

$$\Delta_m \approx \frac{\Delta_{m-1} + \Delta_{m+1}}{2} \quad (m = n+1, n+3, n+5, \dots).$$

Після цього розв'язок в усіх вузлах можна уточнювати за такою формулою:

$$\tilde{y}(x_m; h) \approx y(x_m; h) + \Delta_m \quad (m = n, n+1, n+2, \dots). \quad (6.23)$$

Отже, застосування правила Рунге дає змогу одержувати прийнятні результати навіть за формулами схем невисокого порядку точності.

**Приклад 6.3.** Безрозмірний атмосферний тиск визначають за таким диференціальним рівнянням:

$$u'(x) = -\frac{0.9u}{1+2x}, \quad u(0) = 1,$$

де  $u(x) = \frac{p(x)}{p_0}$ ;  $p_0$  — атмосферний тиск на поверхні Землі, тобто за  $x=0$ ;  $x$  — безрозмірна відстань від неї ( $x = \frac{z}{R}$ , де  $R$  — радіус Землі).

Точний розв'язок цього звичайного диференціального рівняння зі змінними коефіцієнтами такий:

$$u(x) = (1+2x)^{-0.45}.$$

Значення числових розв'язків цього рівняння, одержані за допомогою різних методів, наведено в табл. 6.1.

Таблиця 6.1

x	Ейлера	МРК2		МРК4	Точний розв'язок
		$\alpha = 1$	$\alpha = \frac{1}{2}$		
0.0	1.00000	1.00000	1.00000	1.00000	1.00000
0.2	0.82000	0.86350	0.85729	0.85948	0.85949
0.4	0.71457	0.77260	0.76482	0.76757	0.76759
0.6	0.64311	0.70654	0.69842	0.70129	0.70131
0.8	0.59050	0.65572	0.64765	0.65051	0.65052
1.0	0.54962	0.61503	0.60715	0.60994	0.60995

З цієї таблиці випливає, що числовий розв'язок, одержаний за формулами методу Рунге-Кутти 4-го порядку, різниться від точного лише на одну одиницю п'ятої значущої цифри (див. дод. 26).

**Приклад 6.4.** Знайдемо числовий розв'язок такого безрозмірного рівняння коливань математичного маятника в порожнині:

$$U''(x) + \sin U(x) = 0, \quad U(0) = 2, \quad U'(0) = 0. \quad (6.24)$$

У разі малих коливань рівняння можна лінеаризувати, тобто записати у вигляді

$$u''(x) + u(x) = 0, \quad u(0) = 2, \quad u'(0) = 0. \quad (6.25)$$

Точний розв'язок останньої задачі  $u(x) = 2\cos x$ .

Щоб застосувати розглянуті методи, перейдемо до системи звичайних диференціальних рівнянь 1-го порядку. Наприклад, задачу Коші (6.24) запишемо так:

$$U'(x) = U_1(x) \quad U(0) = 2;$$

$$U_1' = -\sin U(x) \quad U_1(0) = 0.$$

У табл. 6.2 наведено результати інтегрування лінеаризованої (6.25) і вихідної (6.24) задач Коші різними методами  $\left(h = \frac{1}{4}\right)$ .

Таблиця 6.2

x	y <sup>n</sup> для лінійної задачі				Y <sup>n</sup> для нелінійної задачі		
	МЛ	МРК2 α = 1	МРК4	u(x) = = 2 cos x	МЛ	МРК2 α = 1	МРК4
0.00	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
0.25	2.0000	1.9375	1.9378	1.9378	2.0000	1.9716	1.9715
0.50	1.8750	1.7520	1.7552	1.7552	1.9432	1.8860	1.8854
0.75	1.6250	1.4550	1.4634	1.4635	1.8295	1.7412	1.7398
1.00	1.2578	1.0654	1.0807	1.0806	1.6576	1.5349	1.5327
1.25	0.7891	0.6078	0.6307	0.6306	1.4253	1.2659	1.2635
1.50	0.2417	0.1112	0.1416	0.1414	1.1308	0.9366	0.9352
1.75	-0.3550	-0.3930	-0.3564	-0.3564	0.7744	0.5558	0.5574
2.00	-0.9668	-0.8727	-0.8322	-0.8322	0.3614	0.1407	0.1470

*Примітка:* У таблицях 6.1 і 6.2 МРК це є метод Рунге-Кутти, при цьому цифра 2 або 4 означає порядок МРК, а МЛ це є метод ламаних.

Дані табл. 6.2 засвідчують, що розв'язок лінійної задачі (6.25) близький до розв'язку нелінійної (6.24) на невеликому відрізку інтегрування, а результати розрахунків за формулами методу Рунге-Кутти 2-го досить близькі до результатів розрахунків за формулами методу Рунге-Кутти 4-го порядку (див. дод. 27).

## 6.2. Крайова задача

Якщо додаткові умови для звичайних диференціальних рівнянь задано не в одній точці відрізка інтегрування, то задачу називають *крайовою*. Зрозуміло, що її можна поставити лише для звичайних диференціальних рівнянь, у яких  $p \geq 2$ . У цьому разі додаткові умови називають *крайовими*. Вони можуть бути дуже різноманітними. Наведемо один із поширених типів крайових умов:

$$\Phi[u_1(\xi_k), u_2(\xi_k), \dots, u_p(\xi_k)] = \eta$$

або

$$\begin{aligned} \varphi_k[u_1(\xi_k), u_2(\xi_k), \dots, u_p(\xi_k)] &= \eta_k \\ \{\xi_k \in [a, b]; k = \overline{1, p}\}. \end{aligned} \quad (6.26)$$

Незважаючи на різноманітність крайових умов, крайові задачі для звичайних диференціальних рівнянь розв'язують в основному одними й тими самими методами.

*Точні методи* дають змогу виразити розв'язок через елементарні функції чи квадратури від них. Наприклад, крайова задача

$$u''(x) = x, \quad u(0) = u(1) = 0$$

має такий точний розв'язок:

$$u(x) = \frac{x(x^2 - 1)}{6}.$$

До *наближених методів* розв'язування крайових задач належать методи колокації, найменших квадратів, Бубнова — Гальборкіна, Рітца, малого параметра та ін.

Проте, як і для задачі Коші, найуніверсальнішими методами розв'язування крайових задач є *числові*. Насамперед це

балістичний (зведення крайової задачі до задач Коші, стрільби) і різницевої методи. В останньому вихідну задачу зводять до системи алгебричних рівнянь відносно значень шуканого розв'язку у вузлах сітки на відріжку  $[a, b]$ .

### 6.2.1. Числові методи

**Балістичний метод.** *Нелінійні системи 2-го порядку.* Проілюструємо ідею балістичного методу на прикладі розв'язування нелінійної крайової задачі для системи двох звичайних диференціальних рівнянь 1-го порядку:

$$u_1'(x) = f_1(x, u_1, u_2), \quad u_2'(x) = f_2(x, u_1, u_2) \quad \{x \in [a, b]\}; \quad (6.27)$$

$$\varphi_1(u_1(a), u_2(a)) = 0, \quad \varphi_2(u_1(b), u_2(b)) = 0. \quad (6.28)$$

Вибравши довільне  $u_1(a) = \eta$ , розглянемо ліву крайову умову як алгебричне рівняння  $\varphi_1[\eta, u_2(a)] = 0$ . розв'язавши яке знайдемо, що  $u_2(a) = \zeta(\eta)$ .

Тепер, узявши  $u_1(a) = \eta$ ,  $u_2(a) = \zeta$  за початкові умови для звичайних диференціальних рівнянь (6.27), проінтегруємо цю задачу Коші числовим методом, унаслідок чого одержимо числовий розв'язок

$$y_1(x; \eta); \quad y_2(x; \eta),$$

який залежить від  $\eta$  як від параметра.

Підставивши цей розв'язок у праву крайову умову, дістанемо

$$\Psi(\eta) \equiv \varphi_2[y_1(b; \eta), y_2(b; \eta)] \neq 0. \quad (6.29)$$

Зазвичай  $\Psi(\eta) \neq 0$ , але коли  $\eta$  змінювати, то загалом можна знайти таке  $\eta^*$ , що перетворить  $\Psi(\eta)$  на нуль.

Отже, для розв'язання поставленої крайової задачі знайдемо корінь алгебричного рівняння

$$\Psi(\eta) = 0. \quad (6.29')$$

Найпростіший спосіб його розв'язання — метод половинного ділення. Спочатку зробимо пробні «постріли» — вестимемо розрахунки з довільно вибраними  $\eta^{(i)}$  доти, доки серед них не

знайдуться такі  $\eta^{(i)}$  і  $\eta^{(i+1)}$ , для яких  $\Psi(\eta)$  матиме різні знаки. Далі цей відрізок поділимо навпіл до досягнення потрібної точності.

Ефективнішим методом визначення  $\eta^*$  є *метод січних*, згідно з яким наближення для  $\eta^*$  знайдемо за формулою

$$\eta^{(s+1)} = \eta^{(s)} - \frac{(\eta^{(s)} - \eta^{(s-1)})\Psi(\eta^{(s)})}{\Psi(\eta^{(s)}) - \Psi(\eta^{(s-1)})} \quad (s = 1, 2, \dots), \quad (6.30)$$

слід лише вдало підібрати нульові наближення  $\eta^{(0)}$ ,  $\eta^{(1)}$ , тобто вони мають бути близькими до  $\eta^*$ .

Маючи значення  $\eta^*$ , ще раз розв'яжемо задачу Коші, взявши  $\eta = \eta^*$ . У результаті одержимо розв'язок вихідної нелінійної крайової задачі  $u_1(x) \approx y_1(x; \eta^*)$ ,  $u_2(x) \approx y_2(x; \eta^*)$ .

*Лінійні системи 2-го порядку.* У цьому разі балістичний метод особливо просто застосовувати. Нехай відповідна система має вигляд

$$u'_i(x) = \sum_{j=1}^2 p_{ij}(x)u_j(x) + f_i(x) \quad \{i = 1, 2; x \in [a, b]\}; \quad (6.31)$$

$$\sum_{j=1}^2 \alpha_{ij}u_j(\xi_i) = \gamma_i \quad (i = 1, 2; \xi_1 = a, \xi_2 = b) \quad (6.32)$$

$$(|\alpha_{i1}| + |\alpha_{i2}| \neq 0, i = 1, 2).$$

Якщо ж  $\alpha_{12} \neq 0$ , то початкові умови такі:

$$u_1(a) = \eta; \quad u_2(a) = \zeta(\eta) \equiv \frac{\gamma_1 - \alpha_{11}\eta}{\alpha_{12}}. \quad (6.33)$$

Якщо  $\alpha_{11} \neq 0$ , то початкові умови задачі Коші будуть такими:

$$u_2(a) = \eta; \quad u_1(a) = \zeta(\eta) \equiv \frac{\gamma_1 - \alpha_{12}\eta}{\alpha_{11}}. \quad (6.33')$$

Розв'язавши одним із числових методів задачу Коші для звичайних диференціальних рівнянь (6.31) з початковими умовами (6.33) чи (6.33'), одержимо таблицю значень  $y_1(x; \eta)$ ,  $y_2(x; \eta)$ . Зрозуміло, що вони лінійно залежатимуть від  $\eta$ , а тому після підставлення цього розв'язку у праву крайову умову

одержимо лінійну функцію  $\Psi(\eta)$ , яка визначається своїми двома значеннями. Отже, взявши два довільні значення  $\eta^{(0)}$  і  $\eta^{(1)}$  і підставивши їх у (6.30), знайдемо потрібне значення  $\eta^*$ . Тепер розв'яжемо задачу Коші, у початковій умові якої  $\eta = \eta^{(2)}$ , унаслідок чого одержимо розв'язок вихідної лінійної крайової задачі.

*Зауваження.* Обсяг обчислень можна зменшити, скориставшись тим, що загальний розв'язок лінійної крайової задачі дорівнює сумі її *частинного* розв'язку та *загального* розв'язку відповідної однорідної системи:

$$u_i(x) = u_{i0}(x) + C u_{i1}(x).$$

Покладемо, що  $\eta^{(0)} = 0$ . Відтак вирази (6.33) і (6.33') набудуть відповідно такого вигляду:

$$u_1(a) = 0; \quad u_2(a) = \frac{\gamma_1}{\alpha_{12}} \quad (\alpha_{12} \neq 0) \quad (6.34)$$

або

$$u_2(a) = 0; \quad u_1(a) = \frac{\gamma_1}{\alpha_{11}} \quad (\alpha_{11} \neq 0). \quad (6.34')$$

Розв'язавши задачу Коші для вихідної системи з цими початковими умовами, знайдемо необхідний частинний розв'язок вихідної задачі:

$$y_{10}(x), y_{20}(x).$$

Потім розв'яжемо однорідну задачу Коші:

$$u'_i(x) = \sum_{j=1}^2 p_{ij}(x) u_j(x) \quad \{i = 1, 2; x \in [a, b]\}; \quad (6.35)$$

$$\sum_{j=1}^2 \alpha_{ij} u_j(\xi_i) = 0 \quad (i = \overline{1, 2})$$

з такою початковою умовою ( $\eta^{(1)} = 1$ ):

$$u_1(a) = 1; \quad u_2(a) = \frac{-\alpha_{11}}{\alpha_{12}} \quad (\alpha_{12} \neq 0) \quad (6.36)$$



або

$$u_2(a) = 1; \quad u_1(a) = \frac{-\alpha_{12}}{\alpha_{11}} \quad (\alpha_{11} \neq 0). \quad (6.36')$$

Розв'язавши задачу Коші для цієї системи з такими початковими умовами, знайдемо

$$y_{11}(x), y_{21}(x).$$

Отже, загальний розв'язок неоднорідної задачі Коші (завдяки такому вибору початкової умови), який задовольняє лівій крайовій умові, має вигляд

$$y_i(x) = y_{i0}(x) + C y_{i1}(x) \quad (i = 1, 2). \quad (6.37)$$

Якщо його підставити у праву крайову умову, одержимо такий вираз для визначення довільної сталої  $C$ :

$$C = \frac{\gamma_2 - [\alpha_{21} y_{10}(b) + \alpha_{22} y_{20}(b)]}{\alpha_{21} y_{11}(b) + \alpha_{22} y_{21}(b)}.$$

Підставивши  $C$  у (6.37), отримаємо шуканий розв'язок.

З викладеного видно, що балістичний метод простий і застосовний як для лінійних, так і для нелінійних крайових задач з використанням методу Рунге-Кутти.

Він непридатний у разі, коли крайова задача обумовлена добре, а відповідна задача Коші — погано. Тоді числове інтегрування задачі Коші для визначення  $\Psi(\eta)$  виконується з великими похибками. Тому слід починати розрахунки від точки  $x = b$ , що іноді їх поліпшує; якщо ж ні — то вдаються до різницевого методу.

*Нелінійні системи високих порядків.* Розглянемо системи звичайних диференціальних рівнянь  $p$ -го порядку. Тут крайові умови різноманітніші й тому розв'язувати їх значно складніше. Зупинимось на системі звичайних диференціальних рівнянь, для якої додаткові задано лише на кінцях відрізка:

$$u'_k(x) = f_k(x, u_1, u_2, \dots, u_p) \quad \{k = \overline{1, p}; x \in [a, b]\}; \quad (6.38)$$

$$\varphi_k[u_1(a), u_2(a), \dots, u_p(a)] = 0 \quad (k = \overline{1, m}); \quad (6.39)$$

$$\varphi_k[u_1(b), u_2(b), \dots, u_p(b)] = 0 \quad (k = \overline{m+1, p}). \quad (6.40)$$

Нехай у точці  $x = a$  задана більша частина крайових умов  $(m \geq \frac{p}{2})$ , тому візьмемо її за початкову. За “пристрілочні” параметри виберемо якісь  $p-m$  функцій  $u_q(x)$ , тобто покладемо

$$u_q(a) = \eta_q \quad (q = \overline{1, p-m}). \quad (6.41)$$

Підставивши ці значення у ліві крайові умови (6.39) і розв'язавши їх відносно інших  $m$  функцій  $u_q(x)$ , знайдемо інші початкові значення шуканих функцій

$$u_q(a) = \zeta_q(\eta) \quad (q = \overline{p-m+1, p}). \quad (6.42)$$

Тут позначено  $\eta = (\eta_1, \eta_2, \dots, \eta_{p-m})$ .

Тепер, узявши за початкові умови (6.41) та (6.42), розв'яжемо задачу Коші для (6.38) і знайдемо  $y(x; \eta)$ .

Підставимо цей розв'язок у праві крайові умови (6.40), отримаємо значення таких функцій:

$$\Psi_k(\eta) \equiv \varphi_{k+m}[y(b; \eta)] \quad (k = \overline{1, p-m}).$$

Природно, що вони не обертаються на нуль

$$\Psi_k(\eta) \neq 0 \quad (k = \overline{1, p-m}).$$

Отже, щоб розв'язати поставлену крайову задачу, слід знайти розв'язок такої системи алгебричних рівнянь:

$$\Psi_k(\eta) = 0 \quad (k = \overline{1, p-m}). \quad (6.43)$$

Зазначимо, що в загальному випадку зробити це досить складно, бо для кожного  $\eta$  потрібно числово інтегрувати систему звичайних диференціальних рівнянь.

Оскільки явний вигляд  $\Psi_k(\eta)$  невідомий, то систему (6.43) не можна звести до вигляду

$$\eta_k = \Phi_k(\eta) \quad (k = \overline{1, p-m}).$$

тому застосування тут метода простих ітерацій практично неможливе.

Якщо ж спробувати скористатись методом Ньютона (січних), то потрібно додатково проінтегрувати диференціальні рівняння для знаходження  $\left(\frac{\partial \Psi_k}{\partial \eta_q}\right)$ . Тому для розв'язування нелінійних задач балістичний метод застосовувати лише тоді, коли  $p-m=1$ .

Маючи розв'язок

$$\eta^* = (\eta_1^*, \eta_2^*, \dots, \eta_{p-m}^*)$$

системи (6.43), ще раз розв'яжемо задачу Коші для звичайних диференціальних рівнянь (6.38) з початковими умовами (6.41) та (6.42), але вже з використанням  $\eta^*$ , унаслідок чого одержимо розв'язок вихідної нелінійної крайової задачі  $u(x) \approx y(x; \eta^*)$ .

*Лінійні системи високих порядків.* У цьому разі балістичний метод дуже спрощується, що дає змогу легко розв'язувати задачі з довільною різницею  $p-m$ , бо система алгебричних рівнянь: (6.43) будуть лінійними і тому вони однозначно визначаються своїми  $p-m+1$  точками  $\eta^{(s)}$  ( $s = \overline{1, p-m+1}$ ). Тобто, виконавши  $p-m+1$  інтегрування такої задачі Коші з різними наборами  $\eta$ , можна знайти шуканий набір  $\eta^*$ . Тоді  $(p-m+2)$ -е інтегрування задачі Коші забезпечить розв'язок вихідної крайової задачі.

Потрібні обчислення краще виконувати за таким алгоритмом. Візьмемо довільний набір параметрів  $\eta^0$ , розв'яжемо відповідну задачу Коші і позначимо одержані функції так:

$$\Psi_k^0 = \Psi_k(\eta^0) \quad (k = \overline{1, p-m}).$$

Потім змінимо перший параметр  $\eta_1^0$  на величину  $\Delta\eta=1$ , а всі інші залишимо без змін, тобто візьмемо набір

$$\eta^1 = (\eta_1^0 + 1, \eta_2^0, \dots, \eta_{p-m}^0)$$

і знайдемо  $\Psi_k^1$ .

Далі візьмемо набір  $\eta^2 = (\eta_1^0, \eta_2^0 + 1, \dots, \eta_{p-m}^0)$

і знайдемо  $\Psi_k^2$ .

Виконавши повний цикл обчислень, кожену функцію  $\Psi_k(\eta)$  запишемо у вигляді багатовимірного інтерполяційного поліному Ньютона 1-го степеня:

$$\Psi_k(\eta) = \Psi_k^0 + \sum_{q=1}^{p-m} (\Psi_k^q - \Psi_k^0)(\eta_q - \eta_q^0) \quad (k = \overline{1, p-m}).$$

Прирівнявши ці функції до нуля, одержимо СЛАР для визначення шуканих параметрів  $\eta_q^*$ :

$$\sum_{q=1}^{p-m} (\Psi_k^q - \Psi_k^0) \eta_q^* = \sum_{q=1}^{p-m} (\Psi_k^q - \Psi_k^0) \eta_q^0 - \Psi_k^0 \quad (k = \overline{1, p-m}). \quad (6.44)$$

Зазначимо, що тут також можна зменшити кількість інтегрувань задачі Коші на одиницю, але для великих значень  $p-m$  це неістотно зменшує обсяг обчислень, значно ускладнюючи при цьому програмну реалізацію.

Якщо  $p-m=1$  (тобто параметр  $\eta$  один), його значення можна знайти за формулою

$$\eta^* = \eta^0 - \frac{\Psi^0}{\Psi^1 - \Psi^0},$$

що випливає з формули (6.44).

**Різницевий метод. Лінійні задачі.** Розглянемо застосування різницевого методу на прикладі розв'язування крайової задачі для лінійного звичайного диференціального рівняння 2-го порядку зі змінними коефіцієнтами

$$u''(x) + p(x)u'(x) + q(x)u(x) = f(x) \quad (x \in [a, b]) \quad (6.45)$$

і крайові умови загального вигляду

$$\alpha_i u(\xi_i) + \beta_i u'(\xi_i) = \gamma_i \quad (i = 1, 2; \xi_1 = a, \xi_2 = b), \quad (6.46)$$

де  $p(x)$ ,  $q(x)$ ,  $f(x)$  — задані обмежені та неперервні зі своїми другими похідними функції;  $\alpha_i$ ,  $\beta_i$ ,  $\gamma_i$  ( $i = 1, 2$ ) — задані дійсні числа, причому  $|\alpha_i| + |\beta_i| \neq 0$ .

Припустимо, що ця крайова задача має єдиний розв'язок, і введемо на  $[a, b]$  сітку  $a = x_0 < x_1 < x_2 < \dots < x_N = b$ , причому для спрощення рівномірну, тобто  $x_{n+1} - x_n = h \equiv \frac{b-a}{N}$ .

Замінивши похідні в звичайному диференціальному рівнянні симетричними різницевиими співвідношеннями (2.5), які мають похибку  $O(h^2)$ , а в крайових умовах — однобічними співвідношеннями

$$u'(a) \approx \frac{y_1 - y_0}{h} \quad \text{і} \quad u'(b) \approx \frac{y_N - y_{N-1}}{h}$$

з похибкою  $O(h)$ , дійдемо такої різницевої схеми:

$$\frac{y_{n-1} - 2y_n + y_{n+1}}{h^2} + p_n \frac{(y_{n+1} - y_{n-1})}{2h} + q_n y_n = f_n$$

$$\left[ p_n = p(x_n), q_n = q(x_n), f_n = f(x_n); n = \overline{1, N-1} \right]: \quad (6.47)$$

$$\alpha_1 y_0 + \beta_1 \frac{y_1 - y_0}{h} = \gamma_1, \quad \alpha_2 y_N + \beta_2 \frac{y_N - y_{N-1}}{h} = \gamma_2.$$

Одержану СЛАР перепишемо так:

$$(\alpha_1 h - \beta_1) y_0 + \beta_1 y_1 = \gamma_1 h;$$

$$\left( 1 - \frac{hp_n}{2} \right) y_{n-1} - (2 - q_n h^2) y_n + \left( 1 + \frac{hp_n}{2} \right) y_{n+1} = h^2 f_n \quad (6.48)$$

$$(n = \overline{1, N-1});$$

$$-\beta_2 y_{N-1} + (\alpha_2 h + \beta_2) y_N = \gamma_2 h.$$

Оскільки матриця системи (6.48) тридіагональна, то цю СЛАР краще розв'язувати *методом алгебричної прогонки*, для якого показано, що коли хоча б для одного рядка елемент на головній діагоналі за абсолютною величиною більший або дорівнює сумі абсолютних величин елементів, які лежать на бічних діагоналях цього рядка, СЛАР має єдиний розв'язок.

Крім того, можна довести, що числовий розв'язок вихідної задачі, одержаний як наслідок розв'язування цієї СЛАР, має порядок  $O(h)$ , стійкий і прямує до точного з тим самим порядком точності.

Виключити втрату точності можна за допомогою розрахунків на трьох і більше сітках з різними кроками. Якщо всі одержані різницеві розв'язки близькі між собою, це засвідчує їх стійкість. Очевидно, для поліпшення розв'язку можна скористатися правилом Рунге.

*Нелінійні задачі.* Ці задачі особливо складні, бо, як правило, для їх розв'язування потрібно застосовувати ітераційні методи.

Розглянемо таку задачу:

$$u''(x) = f[x, u(x), u'(x)] \quad (x \in [a, b]), \quad u(a) = \alpha, \quad u(b) = \beta. \quad (6.49)$$

Якщо існує неперервна й обмежена похідна  $u^{IV}(x)$ , то, замінивши похідні в рівнянні на різницеві співвідношення (2.5), одержимо таку систему алгебричних рівнянь відносно невідомих значень різницевого розв'язку  $y$  внутрішніх вузлових точках:

$$\begin{aligned} y_{n-1} - 2y_n + y_{n+1} &= h^2 f\left(x_n, y_n, \frac{y_{n+1} - y_{n-1}}{2h}\right); \\ y_0 &= \alpha, \quad y_N = \beta \quad (n = \overline{1, N-1}). \end{aligned} \quad (6.50)$$

Можна довести, що різницевий розв'язок рівномірно прямує до точного з 2-м порядком точності.

Для знаходження розв'язку (6.50) застосуємо метод простих ітерацій:

$$\begin{aligned} y_{n-1}^{(s+1)} - 2y_n^{(s+1)} + y_{n+1}^{(s+1)} &= h^2 f\left(x_n, y_n^{(s)}, \frac{y_{n+1}^{(s)} - y_{n-1}^{(s)}}{2h}\right); \\ y_0 &= \alpha, \quad y_N = \beta \quad (n = \overline{1, N-1}; s = 0, 1, \dots). \end{aligned} \quad (6.51)$$

Зрозуміло, що  $y_0$  і  $y_N$  не будемо уточнювати. На кожній ітерації система (6.51) розв'язується алгебричною прогонкою.

Значно ефективнішим є *метод Ньютона*. Наприклад, якщо розв'язку  $u(x)$  у крайовій задачі

$$u''(x) = f[x, u(x)] \quad (x \in [a, b]), \quad u(a) = \alpha, \quad u(b) = \beta$$

дати приріст  $\Delta_n$  і лінеаризувати праву частину, одержимо такі вирази:

$$y_n^{(s+1)} = y_n^{(s)} + \Delta_n^{(s)} \quad (n = \overline{0, N});$$

$$\begin{aligned} & \left(y_{n-1}^{(s)} + \Delta_{n-1}^{(s)}\right) - 2\left(y_n^{(s)} + \Delta_n^{(s)}\right) + \left(y_{n+1}^{(s)} + \Delta_{n+1}^{(s)}\right) = \\ & = h^2 \left(f_n + \Delta_n f_u|_n\right)^{(s)} \quad (n = \overline{1, N-1}), \end{aligned}$$

а з неї — СЛАР для визначення приростів  $\Delta_n$ :

$$\begin{aligned} & \Delta_{n-1}^{(s)} - \left(2 + h^2 f_u\right)_n^{(s)} \Delta_n^{(s)} + \Delta_{n+1}^{(s)} = \\ & = h^2 f_n^{(s)} - \left(y_{n-1} - 2y_n + y_{n+1}\right)^{(s)} \quad (n = \overline{1, N-1}); \quad (6.52) \end{aligned}$$

$$\Delta_0^{(s)} = \Delta_N^{(s)} = 0 \quad (s = 0, 1, 2, \dots).$$

Оскільки система (6.52) тридіагональна, то її можна розв'язувати алгебричною прогонкою. Якщо  $f_u > 0$ , то ітерації збігаються квадратично і метод Ньютона корисніший за метод простих ітерацій.

Якщо ітерації (6.51) і (6.52) збігаються, то внаслідок неперервності та обмеженості  $f(x, u)$  вони збігаються до розв'язку  $u(x)$ . Тим самим встановлено існування різницевого розв'язку  $y(x)$ .

За нульове наближення на першій сітці беруть або нулі, або значення розв'язку, знайдені наближеними методами. Для уточнення розв'язку можна скористатись правилом Рунге.

**Приклад 6.5.** Розглянемо поперечне згинання шарнірно опертого прямого бруса змінної жорсткості  $EI(z)$ , на який діє поздовжня стискувальна сила  $P$  та розподілене поперечне навантаження  $p(z)$ .

Якщо поздовжню координату  $z$  відлічують від середини бруса ( $-l \leq z \leq l$ ), то згинальний момент  $M(z)$  визначають із такої крайової задачі:

$$\frac{d^2 M(z)}{dz^2} + \frac{P}{EI(z)} M(z) = -p(z); \quad M(-l) = M(l) = 0.$$

Покладемо

$$EI(z) = \frac{EI_0}{1 + \left(\frac{z}{l}\right)^2}; \quad P = \frac{EI_0}{l^2}; \quad p(z) = p_0 = \text{const.}$$

Якщо тепер ввести безрозмірні змінні  $u = \frac{M}{\rho_0 l^2}$ ,  $x = \frac{z}{l}$ ,

одержимо таку крайову задачу:

$$u''(x) + (1 + x^2)u(x) = -1; \quad u(-1) = u(1) = 0. \quad (6.53)$$

1. Для розв'язання цієї задачі балістичним методом перейдемо до крайової задачі для такої системи звичайних диференціальних рівнянь:

$$u'(x) = u_1(x), \quad u_1'(x) = -[(1 + x^2)u(x) + 1]; \quad (6.54)$$

$$u(-1) = u(1) = 0. \quad (6.55)$$

Розв'яжемо дві задачі Коші для (6.54) з такими початковими умовами:

$$u(-1) = 0, \quad u_1(-1) = \eta^{(0)} = 1;$$

$$u(-1) = 0, \quad u_1(-1) = \eta^{(1)} = 2,$$

тобто взявши два довільні значення для другої невідомої функції  $u_1$  у точці  $x = -1$ .

Урахувавши лінійну залежність невідомих функцій  $y(x; \eta)$ ,  $y_1(x; \eta)$  від  $\eta$ , знайдемо  $\eta^*$  за формулою січних:

$$\eta^* = \eta^{(1)} - \frac{(\eta^{(1)} - \eta^{(0)})\Psi(\eta^{(1)})}{\Psi(\eta^{(1)}) - \Psi(\eta^{(0)})},$$

де  $\Psi(\eta) \equiv y(1; \eta)$ .

Результати розв'язання цієї задачі балістичним методом з використанням методу Рунге-Кутти 4-го порядку за  $h = 0,25$  наведено в табл. 6.3 (див. дод. 28):

Таблиця 6.3

$x$	$\eta = 1$		$\eta = 2$		$\eta^* = 1,7362$	
	$y$	$y_1$	$y$	$y_1$	$y$	$y_1$
-1,00	0,0000	1,0000	0,0000	2,0000	0,0000	1,7362
-0,50	0,3465	0,3521	0,8140	1,1749	0,6906	0,9578
0,00	0,3450	-0,3504	1,1415	0,1260	0,9314	0,0003
0,50	0,0114	-0,9565	0,9340	-0,9571	0,6906	-0,9569
1,00	-0,5587	-1,2313	0,2002	-1,9134	0,0000	-1,7335



2. У разі розв'язування крайової задачі (6.53) різнице-вим методом СЛАР (6.48) матиме такий вигляд:

$$y_{n-1} - [2 - (1 + x_n^2)h^2]y_n + y_{n+1} = -h^2;$$

$$y_0 = y_N = 0 \quad \left(h = \frac{2}{N}; n = \overline{1, N-1}\right).$$

Якщо взяти  $N = 8$  ( $h = 0,25$ ), одержимо такий числовий розв'язок (див. дод. 29):

$$y(-0,5) = y(0,5) = 0,6992; \quad y(0,0) = 0,9415.$$

**Приклад 6.6.** Скінченну деформацію пружної струни під дією поперечного навантаження знайдемо як розв'язок такої нелінійної крайової задачі ( $a^2 = 0,49$ ):

$$u'' = -a^2(u')^2 - 1; \quad u(0) = u(1) = 0 \quad \{x \in [0,1]\}.$$

Точний її розв'язок має вигляд

$$u(x) = \ln[\cos a(x - 1/2) / \cos(a/2)] / a^2.$$

1. Щоб розв'язати цю крайову задачу балістичним методом, перейдемо до крайової задачі для системи звичайних диференціальних рівнянь:

$$u'(x) = u_1(x), \quad u_1'(x) = -a^2(u_1)^2 - 1; \quad (6.57)$$

$$u(0) = 0, \quad u(1) = 0 \quad (6.58)$$

і застосуємо для визначення  $\eta^*$  формулу січних (6.30). Для початку розрахунків покладемо

$$\eta^{(0)} = 1,0 \quad \text{та} \quad \eta^{(1)} = 0,9,$$

тобто для знаходження  $\eta^{(2)}$  розв'яжемо дві задачі Коші для (6.57) за таких початкових умов:

$$u(0) = 0, \quad u_1(0) = \eta^{(0)} = 1,0; \quad u(0) = 0, \quad u_1(0) = \eta^{(1)} = 0,9.$$

Застосувавши формулу (6.30), одержимо  $\eta^{(2)} = 0,4830$ .

Результати розв'язання цієї задачі методом Рунге-Кутти 4-го порядку з кроком  $h = 0,125$  за виконання умови (4.15) для двох послідовних розв'язків  $y$  у вузлах сітки, де  $\varepsilon = 10^{-4}$ , наведено в табл. 6.4.

Таблиця 6.4

$x$	Значення $y(x)$ для різних значень $\eta$					$u(x)$
	1,0000	0,9000	0,4830	0,5247	0,5217	
0,000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
0,125	0,1138	0,1020	0,0518	0,0569	0,0565	0,0565
0,250	0,2067	0,1841	0,0871	0,0970	0,0962	0,0962
0,375	0,2806	0,2480	0,1063	0,1209	0,1198	0,1198
0,500	0,3368	0,2948	0,1099	0,1291	0,1276	0,1276

Оскільки розв'язок симетричний відносно середини відрізка інтегрування, дані наведено лише для відрізка  $[0; 0,5]$  (див. дод. 30).

2. Різницеві розв'язки задачі за ітераційною схемою (6.51) із кроком  $h = 0,125$  наведено в табл. 6.5. За нульове наближення взято  $y^{(0)}(x) = 0$ , а умова закінчення ітерацій така сама, як і в попередньому пункті прикладу.

Таблиця 6.5

$x$	$y^{(1)}$	$y^{(2)}$	$y^{(3)}$	$u(x)$
0,000	0,0000	0,0000	0,0000	0,0000
0,125	0,0547	0,0564	0,0564	0,0565
0,250	0,0938	0,0960	0,0961	0,0962
0,375	0,1172	0,1196	0,1197	0,1198
0,500	0,1250	0,1274	0,1275	0,1276

Як видно з даних таблиці, вже друге наближення практично збігається з точним розв'язком (див. дод. 31).

### 6.2.2. Наближені методи

Розглянемо застосування наближених методів до розв'язання крайової задачі для довільного звичайного диференціального рівняння  $p$ -го порядку

$$F(x, u, u', \dots, u^{(p)}) = 0 \quad \{x \in [a, b]\} \quad (6.59)$$

з довільними крайовими умовами

$$\begin{aligned} \Phi_k \left[ u(\xi_1), \dots, u^{(t_{1k})}(\xi_1), \dots, u(\xi_l), \dots, u^{(t_{lk})}(\xi_l) \right] &= \eta_k \\ (k = \overline{1, p}; \xi_j \in [a, b]; t_{jk} \leq p-1; j = \overline{1, l} \geq 2). \end{aligned} \quad (6.60)$$

де  $\xi_j$  — система точок, в яких задані додаткові умови.

Наближений розв'язок шукатимемо в такому вигляді:

$$y_n(x) = y(x; C) \quad (6.61)$$

де  $C = (C_1, C_2, \dots, C_n)$  — вектор вільних параметрів.

Наближений розв'язок  $y_n(x)$  має задовольняти крайові умови (6.60) за довільного вибору  $C$ .

Підставивши вираз (6.61) у формулу (6.59), одержимо відхилення

$$R(x; C) \equiv F(x, y, \dots, y^{(p)}) \neq 0. \quad (6.62)$$

Значення вектора  $C$  знайдемо з умови близькості відхилення  $R(x; C)$  до нуля.

**Метод колокації.** Виберемо систему точок  $\xi_m$  ( $m = \overline{1, n}$ ) (краще з урахуванням особливостей рівняння (6.59)) і прагнутимемо точного виконання умови (6.62), тобто щоб відхилення в них дорівнювало нулю. Ця вимога приводить до системи алгебричних рівнянь:

$$R(\xi_m; C) = 0 \quad (m = \overline{1, n}), \quad (6.63)$$

розв'язанням якої і визначатимемо  $C$ .

**Метод найменших квадратів.** За цим методом вектор  $C$  визначають з умови перетворення на мінімум величин (*інтегральний метод найменших квадратів*):

$$I = \int_a^b R^2(x; C) dx \quad (6.64)$$

або (точковий метод найменших квадратів)

$$I = \sum_{m=1}^M R^2(\xi_m; C) \quad (M \gg n), \quad (6.65)$$

що в обох випадках приводить до системи алгебричних рівнянь:

$$\frac{\partial I}{\partial C_j} = 0 \quad (j = \overline{1, n}). \quad (6.66)$$

**Метод Бубнова — Гальоркіна.** Якщо крайові умови (6.60) лінійні, то наближений розв'язок можна взяти у такому вигляді:

$$y_n(x; C) = \varphi_0(x) + \sum_{k=1}^n C_k \varphi_k(x), \quad (6.67)$$

де  $\{\varphi_k(x)\}$  — система лінійно незалежних функцій (базисні функції), причому  $\varphi_0(x)$  задовольняє неоднорідні крайові умови, а  $\varphi_k(x)$  — аналогічні однорідні умови.

Тоді  $C$  знайдемо розв'язанням СЛАР:

$$\int_a^b R(x; C) \varphi_k(x) dx = 0 \quad (k = \overline{1, n}), \quad (6.68)$$

що впливає з умови ортогональності відхилення  $R(x; C)$  до базисних функцій  $\varphi_k(x)$ .

У разі *лінійних крайових задач* наближений розв'язок завжди беруть у вигляді (6.67) і тоді системи алгебричних рівнянь (6.63), (6.66), (6.68) для визначення  $C$  стають *лінійними*.

Головні переваги наближених методів — компактність запису та достатня універсальність.

**Приклад 6.7.** Проілюструємо застосування наведених методів на прикладі розв'язування крайової задачі (6.53), тобто

$$u''(x) + (1+x^2)u(x) = -1; \quad u(-1) = u(1) = 0,$$

де  $\varphi_0(x) \equiv 0$ .

За систему лінійно незалежних функцій  $\{\varphi_k(x)\}$  візьмемо поліноми

$$\varphi_k(x) = x^{2(k-1)}\psi(x) \quad \left[ \psi(x) = (1-x^2); k = \overline{1, n} \right],$$

які дійсно задовольняють однорідні крайові умови  $\varphi_k(\pm 1) = 0$ . Для відрізка  $[a, b]$  матимемо такий множник:

$$\psi(x) = (a-x)(b-x)$$

Обмежившись двома базисними функціями, матимемо такий наближений розв'язок:

$$y_2(x) = (1-x^2)C_1 + x^2(1-x^2)C_2. \quad (6.69)$$

Підставивши формулу (6.69) у диференціальне рівняння (6.53), одержимо загальний вираз для відхилення

$$R(x; C) = 1 - (1+x^4)C_1 - (x^6 + 11x^2 - 2)C_2. \quad (6.70)$$

1. Визначимо вільні параметри  $C$  спочатку *методом колокації*, взявши за вузли колокації точки  $\xi_1 = 0$ ,  $\xi_{\pm 2} = 1/2$ .

Після підставлення вузлів колокації у (6.63) отримаємо співвідношення

$$R(\xi_m; C) = 0 \quad (m = \overline{1, 2}),$$

які приводять до такої СЛАР:

$$C_1 - 2C_2 = 1; \quad 68C_1 + 49C_2 = 64.$$

Розв'язок цієї системи

$$C_1 = \frac{177}{185} = 0.9567; \quad C_2 = -\frac{4}{185} = -0.0216.$$

Отже, наближений розв'язок крайової задачі (6.53) має вигляд

$$y_2^{(K)}(x) = (1-x^2)(0.9567 - 0.0216x^2).$$

2. У разі застосування *інтегрального методу найменших квадратів* за основний досить взяти відрізок  $[0, 1]$ , оскільки вихідна задача симетрична.

Відповідно до виразу (6.64) отримаємо вираз для величини  $I$ , яку мінімізуватимемо для визначення  $C$

$$I = \int_0^1 R^2(x; C) dx = \int_0^1 [1 - (1 + x^4)C_1 - (x^6 + 11x^2 - 2)C_2]^2 dx.$$

Тут залежність (6.66) набуває вигляду

$$\int_0^1 (1 + x^4 [1 - (1 + x^4)C_1 - (x^6 + 11x^2 - 2)C_2]) dx = 0;$$

$$\int_0^1 (x^6 + 11x^2 - 2) [1 - (1 + x^4)C_1 - (x^6 + 11x^2 - 2)C_2] dx = 0$$

або після обчислення інтегралів

$$\frac{68}{45}C_1 + \frac{3548}{1155}C_2 = \frac{6}{5}; \quad \frac{3548}{1155}C_1 + \frac{63\,404}{4095}C_2 = \frac{38}{21}.$$

Звідси  $C_1 = 0,9327$ ,  $C_2 = -0,0681$ .

Отже, маємо такий наближений розв'язок:

$$y_2^{(I)}(x) = (1 - x^2)(0,9327 - 0,0681x^2).$$

3. Для визначення  $C$  точковим методом найменших квадратів візьмемо за вузли (тут теж обмежимося відрізком  $[0, 1]$ )

$$\xi_1 = 0, \quad \xi_{\pm 2} = \frac{1}{4}, \quad \xi_{\pm 3} = \frac{1}{2}, \quad \xi_{\pm 4} = \frac{3}{4} \quad (M = 4).$$

Для скорочення запису перепишемо відхилення (6.62) у такому вигляді:

$$R(x; C) = 1 - \alpha(x)C_1 - \beta(x)C_2,$$

де  $\alpha(x) = (1 + x^4)$ ;  $\beta(x) = (x^6 + 11x^2 - 2)$ .

Тоді залежність (6.65) набуде вигляду

$$I = \sum_{m=1}^M [1 - \alpha(\xi_m)C_1 - \beta(\xi_m)C_2]^2,$$

а СЛАР (6.66) —

$$C_1 \sum_{m=1}^M \alpha^2(\xi_m) + C_2 \sum_{m=1}^M \alpha(\xi_m)\beta(\xi_m) = \sum_{m=1}^M \alpha(\xi_m);$$

$$C_1 \sum_{m=1}^M \alpha(\xi_m) \beta(\xi_m) + C_2 \sum_{m=1}^M \beta^2(\xi_m) = \sum_{m=1}^M \beta(\xi_m)$$

або після обчислень сум —

$$4,8697C_1 + 3,2428C_2 = 4,3828;$$

$$3,2428C_1 + 25,3656C_2 = 1,8188.$$

Звідси  $C_1 = 0,9316$ ;  $C_2 = -0,0474$ .

Отже, наближений розв'язок буде такий:

$$y_2^{(T)}(x) = (1 - x^2)(0,9316 - 0,0474x^2).$$

4. Для визначення вектора вільних параметрів  $C$  методом Бубнова-Гальоркіна потрібно розв'язати СЛАР, аналогічну (6.68), яку у загальному вигляді можна записати так:

$$\int_0^1 (1 - x^2) [1 - (1 + x^4)C_1 - (x^6 + 11x^2 - 2)C_2] dx = 0;$$

$$\int_0^1 (x^2 - x^4) [1 - (1 + x^4)C_1 - (x^6 + 11x^2 - 2)C_2] dx = 0$$

або після обчислення інтегралів —

$$0,7238C_1 + 0,1651C_2 = 0,6667;$$

$$0,1651C_1 + 0,3821C_2 = 0,1333.$$

Звідси

$$C_1 = 0,9335, \quad C_2 = -0,0544.$$

Отже, маємо такий наближений розв'язок:

$$y_2^{(БГ)}(x) = (1 - x^2)(0,9335 - 0,0544x^2).$$

**Приклад 6.8.** Розв'яжемо нелінійну крайову задачу вигляду

$$u''(x) + a^2[u'(x)]^2 = -1; \quad u(0) = u(1) = 0$$

методом колокації, взявши за базисні функції

$$\varphi_0(x) \equiv 0; \quad \varphi_k(x) = [x(1-x)]^k \quad (k = \overline{1, n}),$$

які задовольняють однорідні крайові умови

$$\varphi_k(0) = \varphi_k(1) = 0.$$

Якщо обмежитися двома базисними функціями, тобто взяти наближений розв'язок таким:

$$y_2(x) = C_1\varphi_1(x) + C_2\varphi_2(x)$$

і підставити в диференціальне рівняння, одержимо відхилення

$$R(x; C) = C_1\varphi_1''(x) + C_2\varphi_2''(x) + \alpha^2[C_1\varphi_1'(x) + C_2\varphi_2'(x)]^2 + 1,$$

де

$$\varphi_1'(x) = 1 - 2x; \quad \varphi_2'(x) = 2x(1 - 3x + 2x^2);$$

$$\varphi_1''(x) = -2; \quad \varphi_2''(x) = 2 - 12x + 12x^2.$$

За вузли колокації візьмемо точки  $\xi_1 = \frac{1}{4}$ ,  $\xi_2 = \frac{1}{2}$  і запишемо систему алгебричних рівнянь для цієї крайової задачі:

$$\frac{R(\xi_m; C)}{2} \equiv -C_1 + (1 - 6\xi_m + 6\xi_m^2)C_2 + \frac{\alpha^2 D_m^2 + 1}{2} = 0 \quad (m = \overline{1, 2}),$$

де позначено

$$D_m = (1 - 2\xi_m)C_1 + 2\xi_m(1 - 3\xi_m + 2\xi_m^2)C_2.$$

Розв'яжемо одержану систему алгебричних рівнянь методом Ньютона (4.13), (4.14), для чого знайдемо похідні від функції

$$f|_m(C) \equiv \frac{R(\xi_m; C)}{2}$$

по  $C_1$  та  $C_2$ :

$$\frac{\partial f|_m}{\partial C_1} = -1 + \alpha^2 D_m(1 - 2\xi_m);$$

$$\frac{\partial f|_m}{\partial C_2} = 1 - 6\xi_m + 6\xi_m^2 + \alpha^2 D_m 2\xi_m(1 - 3\xi_m + 2\xi_m^2).$$



Якщо за нульове наближення взяти  $C = (0,5; 0,04)$ , то після трьох ітерацій дістанемо

$$C_1 = 0,520844, \quad C_2 = -0,041689.$$

Отже, тут наближений розв'язок має вигляд

$$y_2(x) = x(1-x)[0,520844 - 0,041689x(1-x)].$$

Після обчислень матимемо такі значення наближеного розв'язку в окремих точках відрізка інтегрування

$$y_2\left(\frac{1}{6}\right) = 0,07154, \quad y_2\left(\frac{1}{3}\right) = 0,11369,$$

а значення точного розв'язку в них такі:

$$u\left(\frac{1}{6}\right) = 0,07157, \quad u\left(\frac{1}{3}\right) = 0,11372.$$

Таким чином, наближений розв'язок

$$y(x) = x(1-x)[0,520844 - 0,041689x(1-x)]$$

забезпечує знаходження розв'язку  $u(x)$  з точністю до декількох одиниць четвертої значущої цифри.

### 6.3. Задачі на власні значення

Задачі на власні значення — це крайові задачі для системи звичайних диференціальних рівнянь 1-го порядку такого вигляду:

$$u'(x) = f(x, u; \lambda) \quad (6.71)$$

$\{x \in [a, b], f = (f_1, f_2, \dots, f_p); u = (u_1, u_2, \dots, u_p); \lambda = (\lambda_1, \lambda_2, \dots, \lambda_t)\}$ , в якій праві частини залежать від  $t$  числових параметрів  $\lambda$ , значення яких невідомі і їх слід визначати із самої задачі.

Отже, кількість додаткових умов у цій задачі має дорівнювати  $p+t$ . До таких задач зводиться багато задач механіки.

Значення  $\lambda$ , за яких крайова задача має ненульовий розв'язок, називають власними значеннями, а відповідні їм розв'язки  $u_k(x)$  ( $k = \overline{1, p}$ ) — власними функціями крайової задачі.

Прикладом задачі на власні значення для звичайного диференціального рівнянь є класична задача Штурма-Ліувілля для рівняння 2-го порядку за нульових крайових умов:

$$u''(x) + p(x)u'(x) + [\lambda + q(x)]u(x) = 0; \quad u(a) = u(b) = 0. \quad (6.72)$$

Це рівняння має один параметр  $\lambda$ .

Як бачимо, задача потребує трьох додаткових умов, але внаслідок *лінійності та однорідності рівняння* його розв'язок можна визначити з точністю до сталого множника. Це і є задання третьої додаткової умови. Наприклад, за додаткову умову можна взяти умову  $u'(a) = 1$  або якусь іншу.

### 6.3.1. Числові методи

**Балістичний метод.** У задачах на власні значення природним “пристрілочним” параметром є  $\lambda$ . Тому під час розв'язування задачі

$$u'(x) = f(x, u; \lambda), \quad u(a) = \alpha, \quad u(b) = \beta$$

можна відкинути праву крайову умову  $\beta$ , вибравши довільне значення параметра  $\lambda$ , розв'язати задачу Коші для цього звичайного диференціального рівняння з початковою умовою

$$u(a) = \alpha.$$

Унаслідок цього знайдемо розв'язок  $u(x; \lambda)$ . Зрозуміло, що

$$u(b; \lambda) \neq \beta.$$

Перебираючи  $\lambda$ , можна знайти таке його значення, щоб права крайова умова задовольняла умову

$$u(b) \approx \beta,$$

тобто розв'язати алгебричне рівняння

$$f(\lambda) \equiv u(b; \lambda) - \beta = 0.$$

Як правило, балістичний метод застосовують, коли параметр  $\lambda$  один.

**Різницьвий метод.** Як і раніше, до різницевого методу звертаються, якщо  $t > 1$  або коли задача Коші, до якої зведена крайова задача, погано обумовлена.

Уведемо на відрізьку  $[a, b]$  сітку  $x_n$  ( $n = \overline{0, N}$ ), причому для спрощення рівномірно, і замінимо у вихідній задачі похідні на

різницеви співвідношення. Тоді замість звичайного диференціального рівняння та крайових умов одержимо систему алгебричних рівнянь

$$F_k(x_0, x_1, \dots, x_N, y_0, y_1, \dots, y_N; \lambda) = 0 \quad (k = \overline{0, N+1}). \quad (6.73)$$

Ця система алгебричних рівнянь складається з  $N+2$  рівнянь і стільки ж невідомих  $y_0, y_1, \dots, y_N, \lambda$  (для спрощення вважаємо, що власне значення  $\lambda$  одне). Тут, як і в крайовій задачі, виникають запитання: чи має система рівнянь (6.73) дійсний розв'язок, як його знайти та чи близький він до точного?

### 6.3.2. Наближені методи

Більшість розглянутих раніше наближених методів розв'язання крайової задачі можна застосувати для розв'язання задач на власні значення для звичайних диференціальних рівнянь. До найпростіших перетворень приводить *метод Бубнова-Гальоркіна*, але крайові умови при цьому обов'язково мають бути лінійними.

Нехай рівняння має вигляд

$$A(u(x); \lambda) = f(x),$$

а крайові умови — лінійні.

Наближений розв'язок візьмемо у вигляді (6.67). Нагадаємо, що задані неоднорідні крайові умови задовольняє функція  $\varphi_0(x)$ , а сталі  $C$  знайдемо з умов

$$\int_a^b R(x; \lambda) \varphi_k(x) dx = 0 \quad (k = \overline{1, n}), \quad (6.74)$$

де  $R(x; \lambda) = A(y_n(x); \lambda) - f(x)$ .

Система алгебричних рівнянь (6.74) має  $n$  рівнянь з  $n+1$  невідомими  $C_1, C_2, \dots, C_n, \lambda$ .

Отже, одного рівняння в (6.74) не вистачає, але його можна одержати з однієї з крайові умов або покласти  $u'(a) = 1$ .

Зазначимо, що коли за базисні функції брати ортогональні системи функцій  $\{\varphi_k(x)\}$ , то задача істотно спрощується, особливо якщо вона лінійна.

**Приклад 6.9.** Розглянемо окремий випадок задачі (6.72) — задачу Ейлера про визначення критичної сили  $P$  в стійкості пружних стержнів

$$EI_2 u''(z) + Pu(z) = 0, \quad u(0) = u(l) = 0$$

або в безрозмірній формі —

$$u''(x) + \lambda u(x) = 0, \quad u(0) = u(1) = 0 \quad \left( \lambda = \frac{l^2 P}{EI}; \quad x = \frac{z}{l} \right). \quad (6.75)$$

Ця задача має такий точний розв'язок:

$$u_m(x) = C_m \sin m\pi x, \quad \lambda_m = m^2 \pi^2 \quad (m = 1, 2, \dots) \\ (\lambda_1 = \pi^2 \approx 9,8696), \quad (6.76)$$

на ній просто проілюструвати наведені методи розв'язування задач на власні значення: балістичний, різницевий та метод Бубнова-Гальоркіна.

1. Задачу (6.75) з використанням балістичного методу розв'язуватимемо задачу Коші з початковими умовами

$$u(0) = 0, \quad u'(0) = 1,$$

ведучи “пристрілку” за параметром  $\lambda$  до виконання умови  $u(1) = 0$ . Хоча крайова задача лінійна, це не спрощує розв'язування, бо розв'язок  $u(x; \lambda)$  залежить від  $\lambda$  *нелінійно*.

У табл. 6.6 наведено значення найменшого власного значення  $\lambda_1$ , визначеного методами ламаних, методами Рунге-Кутти 2-го порядку ( $\alpha = 1$ ) та 4-го в разі використання балістичного методу у сукупності з методом січних.

В усіх методах за початкові значення брали  $\lambda_1^{(0)} = 10$  та  $\lambda_1^{(1)} = 11$ . Ітерації завершували, коли різниця між сусідніми значеннями  $\lambda$  ставала меншою за  $10^{-3}$  (див. дод. 32).

Таблиця 6.6

$h$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{32}$	$\frac{1}{64}$
МЛ	16.0000	10.1289	9.9334	9.8855	9,8736
МРК2 ( $\alpha = 1$ )	8.5744	9.4234	9.7470	9.8382	9,8617
МРК4	9.9197	9.8733	9.8700	9.8696	9,8696

*Примітка:* У таблиці МРК це є метод Рунге-Кутти, при цьому цифра 2 або 4 означає порядок МРК, а МЛ це є метод ламаних.

2. Застосування різницевого методу для розв'язання задачі (6.75) приводить до такої СЛАР:

$$y_{n-1} - By_n + y_{n+1} = 0, \quad y_0 = y_N = 0$$

$$\left[ B = 2 - \lambda h^2; n = \overline{1, N-1} \right]. \quad (6.77)$$

Отже, одержали задачу на власні значення для матриці. Тут буде  $N-1$  рівнянь з невідомими  $y_1, \dots, y_{N-1}, \lambda$ . Ця задача має спектр власних значень, який складається з  $(N-1)$  чисел (згідно з порядком матриці). Перші власні значення є наближення до перших  $\lambda_m$  власних значень з дискретного спектра розв'язуваної задачі. різницева схема (6.77) така, що її матриця симетрична. отже, наближені власні значення дійсні.

У цьому разі власні значення знаходитимемо так, як і раніше для матриць. Матриця СЛАР тридіагональна, тому знаходження власних значень особливо просте. За невеликої кількості інтервалів після побудови характеристичного полінома за допомогою (5.18) його корені (власні значення) знаходитимемо методом парабол.

Знайдемо власні значення  $\lambda$  за деяких значень  $N$ .

Нехай  $N = 2$ , тоді  $h = 1/2$  і СЛАР буде такою:

$$y_0 - (2 - \lambda(1/2)^2)y_1 + y_2 = 0, \quad y_0 = y_2 = 0.$$

Розв'язок цієї системи  $\lambda_1^{(2)} = 8$ .

У разі  $N = 3$  ( $h = 1/3$ ) СЛАР (6.77) набуває вигляду

$$\begin{pmatrix} -B & 1 \\ 1 & -B \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Отже, характеристичний поліном такий:  $B^2 - 1 = 0$

або  $Z^2 - 4Z + 3 = 0 \left( Z = \frac{\lambda}{9} \right)$ ,

а власні значення  $\lambda_1^{(3)} = 9$ ,  $\lambda_2^{(3)} = 27$ .

Якщо  $N = 4$  ( $h = 1/4$ ), то характеристичний поліном такий:  $B(B^2 - 2) = 0$  і власні значення  $\lambda_1^{(4)} = 9,3726$ ,  $\lambda_2^{(4)} = 32$ ,  $\lambda_3^{(4)} = 54,6274$ .

Після застосування правила Рунге до останніх двох наближень, матимемо  $\lambda_1^{(4)}(1/4) \approx 9.8517$

3. У разі застосування методу Бубнова-Гальоркіна до цієї задачі на власні значення оператор  $A$  має вигляд

$$A(\cdot) \equiv (\cdot)^n + \lambda(\cdot).$$

Наближений розв'язок візьмемо у такому вигляді:

$$y_n(x; C) = \sum_{k=1}^n C_k x^k (1-x),$$

Видно, що наближений розв'язок істотно різниться від  $\sin \pi x$ , але задані крайові умови задовольняє.

Третьою необхідною додатковою умовою може бути умова нормування його розв'язку. Для цього можна покласти  $u'(0) = 1$ , що дає  $C_1 = 1$ .

Нехай  $n = 1$ , тоді відхилення  $R$  має вигляд

$$R(x) \equiv -2 + \lambda x(1-x).$$

Підставивши його у формулу (6.74), одержимо рівняння для визначення  $\lambda$ :

$$\int_0^1 [-2 + \lambda x(1-x)] x(1-x) dx = 0.$$

З цього рівняння випливає, що

$$\lambda^I = 10, \quad y^I(x) = x(1-x).$$

Якщо покласти  $n = 2$ , одержимо такі власні значення та функції:

$$\lambda^I = 10, \quad y^I(x) = x(1-x);$$

$$\lambda^{II} = 42, \quad y^{II}(x) = x(1-x)(1-11x/13).$$

Звідси видно, що перше власне значення близьке до  $\pi^2$ , а друге — різниться від точного  $4\pi^2$ .

Отже, метод Бубнова-Гальборкіна дає змогу досить добре визначити найменші власні значення, а власні функції — дещо гірше.



### Задачі для самостійного розв'язування

1. Застосовуючи наближені методи, знайдіть перші три наближення такої задачі Коші на відрізку  $[0; 1]$ :

$$u' = 4u(1+x), \quad u(0) = 1.$$

2. Розв'яжіть задачу Коші на відрізку  $[0; 0,5]$  методом лама-них:

$$u' = 0,1u^2 + x, \quad u(0) = 1 \quad (h = 0,1).$$

3. Розв'яжіть задачу Коші на відрізку  $[0; 1]$  ( $h = 0,2$ ) методами Рунге — Кутти 2-го порядку:

$$u'' = xu, \quad u(0) = 1, \quad u'(0) = 1.$$

4. Знайдіть розв'язок лінійної крайової задачі 2-го порядку ( $h = 0,1$ ) числовими методами:

$$u'' + xu' - \frac{u}{2x} = 1; \quad u(2) + 2u'(2) = 1; \quad u(2,3) = 2,15.$$

5. Розв'яжіть нелінійну крайову задачу різницеvim методом:

$$u'' = 2 - \frac{4u^2}{\sin^2 x}; \quad u'(1) = 0,9093; \quad u(2) = 0,8263.$$

6. Знайдіть розв'язок крайової задачі ( $h = 0,2$ ) балістичним методом:  $u'' - u'u = e^x, \quad u(0) = 1, \quad u(1) = -1.$

7. Розв'яжіть такі крайові задачі наближеними методами:

$$u'' = 2x, \quad u(0) = u(1) = 0;$$

$$u'' + u = -1, \quad u(-1) = u(1) = 0;$$

$$u'' + u = -x, \quad u(0) = u(1) = 0.$$

8. Розв'яжіть задачу на власні значення описаними методами:  $u'' + \lambda x^2 u = 0, \quad u(0) = u(1) = 0.$

## Розділ 7

# РІВНЯННЯ В ЧАСТИННИХ ПОХІДНИХ

---

До розв'язання диференціальних рівнянь у частинних похідних приводять задачі механіки суцільного середовища.

1. Поширення теплоти в неоднорідному ізотропному суцільному тілі (*задача теплопровідності*) описує рівняння

$$\operatorname{div}(\alpha \nabla u) - \gamma \rho u_t = -\phi \left( u_t = \frac{\partial u}{\partial t} \right), \quad (7.1)$$

де  $u$  — температура тіла;  $\rho$  — його густина;  $\gamma$  — питома теплоємність;  $\phi$  — інтенсивність джерел теплоти, тобто кількість теплоти, яку виділяє одиниця об'єму за одиницю часу;  $\alpha$  — коефіцієнт теплопровідності.

У разі однорідного тіла це рівняння можна записати так:

$$u_t = k \Delta u + f, \quad (7.1')$$

де  $k = \alpha / (\gamma \rho)$ ;  $f = \phi / (\gamma \rho)$ .

2. Рух рідини (суцільного середовища) визначають за рівнянням

$$\rho_t + \operatorname{div}(\rho \vec{v}) = \phi, \quad (7.2)$$

де  $\vec{v}$  — вектор швидкості руху частинок рідини;  $\rho$  — її густина;  $\phi$  — інтенсивність джерел, тобто кількість рідини, яку виділяє одиниця об'єму за одиницю часу.

Якщо рідина нестислива, це рівняння набуває вигляду

$$\operatorname{div} \vec{v} = \phi / \rho. \quad (7.2')$$

За потенційного потоку рідини  $\vec{v} = -\nabla u$ , і тоді з рівняння нерозривності (7.2') одержимо рівняння потенційного потоку нестислової рідини:

$$\Delta u = -f, \quad (7.2'')$$

де  $f = \phi / \rho$ .



3. Малі поперечні коливання струни описують рівнянням

$$Pu_{xx} - \rho u_{tt} = -\rho f \left( u_{xx} = \frac{\partial^2 u}{\partial x^2} \right), \quad (7.3)$$

де  $u$  — відхилення точок струни від стану рівноваги;  $P$  — натяг струни;  $\rho$  — її лінійна густина;  $f$  — масова сила, тобто сила, яка діє на одиницю маси струни, перпендикулярна до неї.

Для однорідної струни, тобто за  $\rho = \text{const}$ , це рівняння можна записати у такому вигляді:

$$u_{tt} = c^2 u_{xx} + f, \quad (7.3')$$

де  $c^2 = P / \rho$ .

Останнє рівняння збігається з хвильовим рівнянням для випадку однієї просторової координати, його називають *рівнянням коливань струни*.

4. Відхилення точок мембрани від стану рівноваги під час її малих поперечних коливань є розв'язок рівняння

$$P(u_{,11} + u_{,22}) - \rho u_{tt} = -\rho f \left( u_{,11} = \frac{\partial^2 u}{\partial x_1^2} \right), \quad (7.4)$$

де  $P$  — натяг мембрани, тобто сила, яка діє на одиницю довжини перетину мембрани, перпендикулярна до цього перетину, лежить у дотичній площині до мембрани;  $\rho$  — поверхнева густина мембрани;  $f$  — масова сила, тобто сила, яка діє на одиницю маси мембрани, перпендикулярна до мембрани.

Якщо мембрана однорідна, тобто  $\rho = \text{const}$ , дістанемо рівняння, відоме під назвою *рівняння коливань мембрани*:

$$u_{tt} = c^2 \Delta u + f, \quad (7.4')$$

де  $c^2 = P / \rho$ .

Це є хвильове рівняння з двома просторовими координатами.

5. Функцію напружень  $u$ , що пов'язана із функцією кручення  $\psi$  співвідношенням  $u = \psi - \frac{x_1^2 + x_2^2}{2}$ , можна визначити розв'язуванням такої крайової задачі:

$$\Delta u = -2, \quad u|_{\Gamma} = 0. \quad (7.5)$$

## 7.1. ОСНОВНІ ПОНЯТТЯ

### 7.1.1. Вступні зауваження

Запишемо рівняння з частинними похідними з однією невідомою функцією  $u$ , яка залежить від часу  $t$ , координати  $r$  та інших змінних, у такому вигляді:

$$F(t, r, u, u_t, u_r, \dots) = 0.$$

Шукатимемо його розв'язок в області

$$G(t, r, \dots) = g(r, \dots) \times [t_0, T],$$

тобто в деякій просторовій зоні  $g(r, \dots)$  на відрізку часу  $[t_0, T]$ . У цьому разі додаткові умови, задані за  $t = t_0$ , називають *початковими умовами*, а додаткові умови на границі  $\gamma(r)$  області  $g(r)$  — *крайовими умовами*.

*Задачу Коші* (задано лише початкові умови) можна побудувати для рівняння теплопровідності в необмеженому просторі

$$u_t(r, t) = k\Delta u(r, t) + f(r, t), \quad u(r, t_0) = \mu(r). \quad (7.6)$$

Задачу з початковими і крайовими умовами називають *мішаною крайовою задачею* або *нестационарною крайовою задачею*. Додаткові умови такої задачі мають вигляд

$$u(r, t_0) = \mu(r) \quad [r \in g(r)]; \quad (7.7)$$

$$u(r, t)|_{\gamma} = \mu_1(r, t) \quad [t_0 \leq t \leq T].$$

Далі розглянемо рівняння з частинними похідними 1- та 2-го порядку, причому лише *коректно поставлені задачі*, тобто задачі, розв'язок яких існує для деякого класу початкових і крайових умов, єдиний та неперервно залежить від цих додаткових умов і коефіцієнтів рівнянь.

### 7.1.2. Класифікація рівнянь у частинних похідних 2-го порядку

Загальний вигляд рівняння 2-го порядку з двома змінними такий:

$$Au_{,11} + 2Bu_{,12} + Cu_{,22} + Du_{,1} + Eu_{,2} + F = 0. \quad (7.8)$$

У загальному випадку коефіцієнти рівняння можуть залежати від  $x_1$ ,  $x_2$  та шуканої функції  $u$ . З огляду на це рівняння (7.8) може бути: зі сталими коефіцієнтами (якщо вони не залежать від змінних); лінійним зі змінними коефіцієнтами, коли  $F$  лінійно залежить від  $u$ , а інші від нього не залежать; квазілінійним, коли всі коефіцієнти залежать від  $u$  (лінійним відносно похідних).

Якщо лише коефіцієнти  $D$  і  $E$  не дорівнюють нулю, це є рівняння *перенесення*. Якщо у ньому однією зі змінних є час  $t$ , це рівняння називають *еволюційним*.

Якщо один з коефіцієнтів  $A$ ,  $B$  або  $C$  відрізняється від нуля, воно залежно від знака дискримінанта  $d = B^2 - AC$  може належати до одного з трьох типів: *еліптичного* ( $d < 0$ ), *параболічного* ( $d = 0$ ) або *гіперболічного* ( $d > 0$ ).

Зазначимо, що рівняння зі змінними коефіцієнтами може мати різний тип у різних точках області  $G$ , що істотно ускладнює формулювання повної постановки задачі й обґрунтування її коректності.

### 7.1.3. Різницевий метод розв'язування рівнянь у частинних похідних

Відомо, що розв'язування лише найпростіших рівнянь з частинними похідними може бути проведено аналітичними або наближеними методами, але основними є *числові*. До найпоширеніших належить *різницевий метод*. Для його застосування в області  $G(r, t)$  або  $g(r)$  уведемо сітку, в основному прямокутну (для просторових областей сітка складається з прямокутних паралелепіпедів).

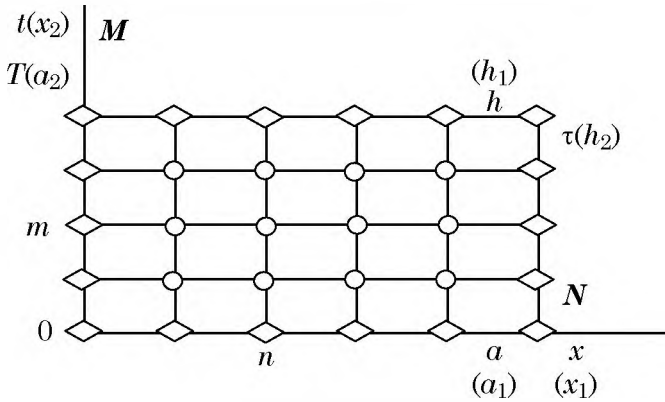


Рис. 7.1

На рис. 7.1 зображено сітки для двох прямокутних областей

$$G = [0 \leq x \leq a, 0 \leq t \leq T] \text{ та } g = [0 \leq x_1 \leq a_1, 0 \leq x_2 \leq a_2].$$

Для їх побудови сторони прямокутника поділимо на елементарні підвідрізки точками

$$x_n = nh \left( n = \overline{0, N}; h = \frac{a}{N} \right); \quad t = m\tau \left( m = \overline{0, M}; \tau = \frac{T}{M} \right) \quad (7.9)$$

або

$$x_{1n} = nh_1 \left( n = \overline{0, N}; h_1 = \frac{a_1}{N} \right); \quad x_{2m} = mh_2 \left( m = \overline{0, M}; h_2 = \frac{a_2}{M} \right). \quad (7.10)$$

Через ці точки проведемо дві сім'ї прямих  $x = \text{const}$  та  $t = \text{const}$  або  $x_1 = \text{const}$  та  $x_2 = \text{const}$ , які і утворюють сітки з прямокутними комірками. Довільний вузол цих сіток, номер якого  $(n, m)$ , визначимо координатами  $(x_n, t_m)$  або  $(x_{1n}, x_{2m})$ .

Якщо у вихідній диференціальній задачі, записаній разом з додатковими умовами в операторному вигляді

$$A(u(x, t)) = f(x, t) \quad (x, t) \in G, \quad (7.11)$$

значення похідних передати значеннями шуканої функції у вузлах сітки за допомогою різницьових співвідношень згідно з

вибраним *шаблоном* (тобто розміщенням вузлів, які використовують у разі апроксимації похідних), то одержимо різницеву задачу

$$A_h(y) = \varphi \quad (x_n, t_m) \in G_h. \quad (7.12)$$

Як наслідок, отримаємо систему алгебричних рівнянь відносно наближених значень  $u$  у вузлах сітки (*різницева схема*), розв'язавши яку визначимо числовий розв'язок  $u$ . Для спрощення спочатку вважатимемо, що сітка залежить від одного параметра  $h$ , тобто  $\tau = rh$  ( $r = \text{const}$ ).

Значення шуканої функції  $u_n^m = u(x_n, t_m)$  у вузлах сітки  $(x_n, t_m) \in G$  наближено замінимо на значення сіткової функції  $y_n^m$  в тих самих вузлах з похибками  $\delta y_n^m = y_n^m - u_n^m$ . Позначимо максимальну за модулем похибку через  $\delta y = \max |\delta y_n^m|$ . Тоді різницева схема буде *збіжною* за виконання такої умови:

$$\lim_{h \rightarrow 0} \delta y = 0.$$

Якщо при цьому  $\delta y \leq Mh^k$ , де  $M = \text{const} > 0$ , то РС (7.12) має  $k$ -й порядок точності, тобто вона збігається із швидкістю  $O(h^k)$ .

Порядок точності різницевої схеми за наявності декількох незалежних змінних також можна оцінювати за значеннями кроків. Якщо виконується умова

$$\delta y \leq M(h^p + \tau^q),$$

то різницева схема збігається зі швидкістю  $O(h^p + \tau^q)$  і має  $p$ -й порядок точності за змінною  $x$  і  $q$ -й порядок — за  $t$ .

Нехай відхилення (*похибка апроксимації*) різницевої схеми

$$R = \max_{(x,t) \in G_h} |\varphi - A_h(u)|$$

дорівнює  $O(h^k)$  або  $O(h^p + \tau^q)$ , коли  $h$  та  $\tau$  незалежні. Тоді різницева схема (7.12) апроксимує вихідну задачу (7.11), якщо в разі зменшення кроку сітки відхилення  $R$  прямує до нуля, тобто якщо

$$\lim_{h \rightarrow 0} R = 0 \quad \text{або} \quad \lim_{\substack{h \rightarrow 0, \\ \tau \rightarrow 0}} R = 0.$$

Апроксимацію такого типу, коли відхилення  $R$  прямує до нуля в разі прямування до нуля  $h$  та  $\tau$  за довільним законом (без будь-яких умов), називають *безумовною* або *абсолютною апроксимацією*.

За *умовної* апроксимації накладаються деякі умови на розміри кроків за просторовими змінними та часом.

Різницева схема (7.12) *стійка*, якщо її розв'язок неперервно залежить від вихідних даних, тобто їх малим змінам відповідають малі зміни розв'язку. Стійкість є характеристика чутливості різницевої схеми до різного роду похибок, тобто вона — внутрішня властивість різницевої задачі і ця властивість не пов'язана з вихідною диференціальною задачею (*на відміну від збіжності й апроксимації*).

Зауважимо, що *стійкість* також буває *умовною* і *безумовною* залежно від того, накладаються чи ні обмеження на співвідношення між кроками за різними незалежними змінними.

Слід мати на увазі, що для впевненого розв'язування конкретних задач різницевим методом потрібні обчислювальні експерименти, які полягають у розрахунках з різними кроками за різних вихідних даних, особливо в задачах, які мають точний розв'язок або числовий, знайдений іншим числовим методом. Тут різницевий розв'язок також можна уточнювати за правилом Рунге, якщо за всіма змінними порядок швидкості збіжності однаковий.

Для спрощення далі розглядатимемо в основному прямокутні області, вкриті прямокутною сіткою зі сталими кроками.

### 7.1.4. Рівняння 1-го порядку

Прикладом рівняння з частинними похідними 1-го порядку є *рівняння перенесення*, яким описують процеси перенесення частинок у середовищах, поширення збурень у них та ін.

Обмежимося розглядом лінійного рівняння, припустивши, що шукана функція  $u$  залежить від часу  $t$  та однієї просторової координати  $x$ . Тоді лінійне рівняння перенесення матиме такий вигляд:

$$u_t(x,t) + c u_x(x,t) = f(x,t), \quad (7.13)$$

де  $c$  — швидкість перенесення, яку вважатимемо сталою.

Якщо  $c > 0$ , збурення поширюються зліва направо в додатному напрямку осі  $x$ .

Розрахункова область під час розв'язання рівняння (7.13) може бути як необмеженою, так і обмеженою. У першому випадку, задавши початкові умови за  $t = 0$ :

$$u(x,0) = \mu(x), \quad (7.14)$$

одержимо задачу Коші для півплощини ( $t \geq 0, -\infty < x < \infty$ ).

На практиці, як правило, розв'язують рівняння перенесення (7.13) у деякій обмеженій області  $G$ , наприклад у прямокутнику (див. рис. 7.1).

Початкові умови (7.14) в цьому разі задають на відрізку  $[0, \alpha]$ , а крайові умови — на відрізку  $[0, T]$ , оскільки за  $c > 0$  збурення поширюються праворуч. Цю умову запишемо у вигляді

$$u(0,t) = \mu_1(t). \quad (7.15)$$

Отже, задача полягає в розв'язанні рівняння з частинними похідними (7.13) з початковими (7.14) і крайовими умовами (7.15) в області  $G = [0 \leq x \leq \alpha, 0 \leq t \leq T]$ .

Розглянемо різницевий метод розв'язання задачі (7.13) – (7.15).

Уведемо в області  $G$  прямокутну сітку, причому для спрощення рівномірну за обома координатами (7.9).

Для побудови різницевої схеми потрібно вибрати шаблон. Візьмемо його у вигляді правого нижнього кута (рис. 7.2, а) в околі точки  $(x_n, t_m)$  і замінимо похідні на різницеві співвідношення з використанням однобічних різниць

$$\begin{aligned} u_t(x_n, t_m) &= \frac{u_n^{m+1} - u_n^m}{\tau} + O(\tau); \\ u_x(x_n, t_m) &= \frac{u_n^m - u_{n-1}^m}{h} + O(h). \end{aligned} \quad (7.16)$$

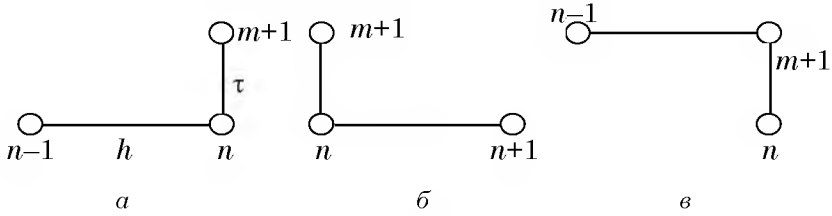


Рис. 7.2

Для скорочення запису тут і в подальшому замість індексу  $m$  іноді користуватимемося такими позначеннями:

$$u(x_n, t_m) = u_n, \quad u(x_n, t_{m+1}) = \hat{u}_n, \quad u(x_n, t_{m-1}) = \check{u}_n.$$

Підставивши вирази (7.16) у рівняння (7.13), одержимо таку різницеву схему:

$$\frac{\hat{y}_n - y_n}{\tau} + c \frac{y_n - y_{n-1}}{h} = \varphi_n \left[ \varphi_n = f\left(x_n - \frac{h}{2}, t_m + \frac{\tau}{2}\right) \right], \quad (7.17)$$

яка після розв'язання відносно невідомого наближеного значення  $\hat{y}_n$  шуканого розв'язку на  $(m + 1)$ -му шарі набуває такого вигляду:

$$\hat{y}_n = \lambda y_{n-1} + (1 - \lambda) y_n + \tau \varphi_n \quad (7.18)$$

$$\left( \lambda = \frac{c\tau}{h}; n = \overline{1, N}; m = \overline{0, M-1} \right).$$

Для початку розрахунку, тобто для обчислення  $u$  на першому шарі, потрібні його значення за  $m = 0$ . Вони визначаються початковими (7.14) і крайовими умовами (7.15), які можна записати так:

$$y_n^0 = \mu(x_n) \quad (n = \overline{0, N}); \quad y_0^m = \mu_1(t_m) \quad (m = \overline{0, M}). \quad (7.19)$$

Одержана різницєва схема — явна, оскільки значення  $u$  в кожному вузлі верхнього шару  $t = t_{m+1}$  за допомогою (7.18) можна виразити через раніше знайдені їхні значення на попередньому шарі  $m$ .

Отже, розв'язання вихідної задачі для рівняння з частинними похідними (7.13) – (7.15) зведено до розв'язання різнице-



вої задачі (7.18), (7.19). Знайдені значення сіткової функції  $y_n^m$  беруть за наближені значення шуканої функції  $u_n^m$  у вузлах сітки.

Зазначимо деякі властивості цієї різницевої схеми. Вона апроксимує вихідну задачу з першим порядком похибки  $O(\tau+h)$  і є умовно стійка.

Умова Куранта стійкості різницевої схеми (7.18)

$$0 < \tau \leq \frac{h}{c}.$$

Ці властивості мають місце, коли  $u(x,t)$ ,  $\mu(x)$  та  $\mu_1(t)$  двічі неперервно диференційовані, а  $f(x,t)$  має неперервні перші похідні.

Оскільки різницева схема (7.18) стійка й апроксимує вихідну задачу, то різницевий розв'язок збігається до точного з першим порядком точності за  $h \rightarrow 0$ ,  $\tau \rightarrow 0$ .

Явну різницеву схему (7.18) можна застосовувати і для необмеженої області  $G$ , оскільки крайову умову (7.15) не потрібно використовувати.

Зазначимо, що за  $c < 0$  умова стійкості не виконується і різницева схема не збіжна, але якщо взяти лівий нижній кут (див. рис. 7.2, б), то одержимо потрібну умовно стійку різницеву схему.

Неявну різницеву схему за  $c > 0$  можна одержати, беручи шаблон у вигляді правого верхнього кута (див. рис. 7.2, в). Ця різницева схема після перетворень набуває вигляду

$$\hat{y}_n = \frac{y_n + \lambda \hat{y}_{n-1} + \tau \varphi_n}{1 + \lambda}.$$

Це *безумовно стійка* двошарова різницева схема 1-го порядку точності, і хоча вона будувалась як неявна, практичні розрахунки за нею ведуть, як і за явною. Звичайно, вона зручніша для застосування, оскільки виключає обмеження на величину кроку  $\tau$ .

**Приклад 7.1.** Розглянемо застосування різницевого методу до розв'язання такої задачі перенесення:

$$u_t + u_x = xt, \quad u(x,0) = 1 - x, \quad u(0,t) = 1 + t$$

у прямокутнику  $G = \left[ 0 \leq x \leq 1, 0 \leq t \leq \frac{1}{2} \right]$ .

Для виконання умови стійкості  $\tau \leq \frac{h}{c}$  візьмемо такі

кроки по  $x$  та по  $t$ :  $h = \frac{1}{4}$ ,  $\tau = \frac{1}{8}$  (тут  $c = 1$ ).

Результати розрахунків за різницевою схемою (7.18) та (7.19) наведено в табл. 7.1 (див. дод. 33).

Таблиця 7.1

$t$	$x$				
	0,0000	0,2500	0,5000	0,7500	1,0000
0,000	1,0000	0,7500	0,5000	0,2500	0,0000
0,125	1,1250	0,8760	0,6279	0,3799	0,1318
0,250	1,2500	1,0034	0,7607	0,5186	0,2764
0,375	1,3750	1,3116	0,8967	0,6641	0,4316
0,500	1,5000	1,2601	1,0347	0,8146	0,5957

## 7.2. Еліптичні рівняння

До рівнянь цього типу належать, наприклад, рівняння в задачах кручення стержня (7.5), стаціонарного розподілу температури, руху потенційного потоку нестисливої рідини (7.2'') та, власне, усі наведені на початку розділу рівняння, в яких відсутні похідні за часом, тобто стаціонарні задачі. Отже, у всіх цих задачах шуканий розв'язок залежить лише від просторових координат.

Розглянемо *двовимірну задачу Діріхле* у довільній області  $g(x)$  [ $x = (x_1, x_2)$ ] з межею  $\gamma$  для рівняння Пуассона (крайова задача)

$$\Delta u(x) \equiv u_{,11}(x) + u_{,22}(x) = -f(x);$$

$$u(x)|_{\gamma} = \mu(x) \quad [x \in g(x)]. \quad (7.20)$$

## 7.2.1. Різницевий метод

Уведемо в області  $g$  прямокутну сітку, причому для спрощення — рівномірну за обома координатами (7.10).

Візьмемо такий шаблон (рис. 7.3) в околі точки  $(x_{1n}, x_{2m})$  і замінимо похідні в задачі (7.20) на різницеві співвідношення

$$(u_{,11})_{n,m} = (u_{n-1,m} - 2u_{nm} + u_{n+1,m})/h_1^2 + O(h_1^2);$$

$$(u_{,22})_{n,m} = (u_{n,m-1} - 2u_{nm} + u_{n,m+1})/h_2^2 + O(h_2^2).$$

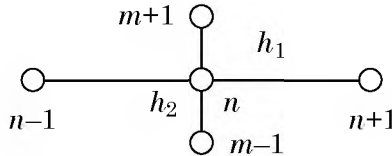


Рис. 7.3

У результаті прийдемо до СЛАР — різницевої схеми:

$$(y_{n-1,m} - 2y_{nm} + y_{n+1,m})h_1^{-2} + (y_{n,m-1} - 2y_{nm} + y_{n,m+1})h_2^{-2} = -f_{mn},$$

де  $u(x_{1n}, x_{2m}) \approx y_{nm}$ ,  $f_{nm} = f(x_{1n}, x_{2m})$ .

Крім того, потрібно врахувати задані крайові умови на  $\gamma$ , що особливо просто зробити, коли область  $g$  прямокутна. Це СЛАР відносно невідомих наближених значень  $y_{nm}$  шуканої функції  $u(x)$  у внутрішніх вузлах області  $g$ . Її можна розв'язувати методом Гаусса. Якщо існують неперервні  $u_{,1111}$  і  $u_{,2222}$ , то порядок одержуваного різницевого розв'язку у дорівнює  $O(h_1^2 + h_2^2)$ .

За великого числа невідомих одержану СЛАР розв'язують ітераційними методами. Розглянемо найпростіший із них — процес Лібмана (*усереднення*), який має таку розрахункову формулу:

$$y_{nm}^{(s+1)} = \left[ y_{n-1,m} + y_{n+1,m} + (y_{n,m-1} + y_{n,m+1})\alpha^2 + h_1^2 f_{nm} \right]^{(s)} / \left[ 2(1 + \alpha^2) \right] \\ \left( \alpha = \frac{h_1}{h_2}; s = 0, 1, \dots \right). \quad (7.21)$$

У разі рівномірного кроку сітки ( $h = \text{const}$ ) маємо

$$y_{nm}^{(s+1)} = (y_{n-1,m} + y_{n+1,m} + y_{n,m-1} + y_{n,m+1} + h^2 f_{mn})^{(s)} / 4, \quad (7.22)$$

Різницьові схеми (7.21) та (7.22) — абсолютно (безумовно) стійкі.

За нульові наближення  $y_{nm}^{(0)}$  можна взяти значення, які належать до відрізка  $[\min u(x)|_\gamma, \max u(x)|_\gamma]$ , або значення функції  $\mu(x)$  у відповідних точках області  $g(x)$ .

У разі довільної області  $g$  найчастіше вводять квадратну сітку  $h_1 = h_2 = h = \text{const}$  та покладають

$$y^{(0)}(A_h) \approx \mu(x_{1A}, x_{2A}),$$

де  $A_h$  — межовий вузол (точка межі  $\gamma_h$  сіткової області  $g_h$  з кроком  $h$ ). Тобто на  $\gamma_h$  переносять значення з межі  $\gamma$  області  $g$ , які в подальшому уточнюють за формулою

$$y^{(s+1)}(A_h) = u(A) + \frac{[y^{(s)}(B_h) - u(A)]\delta}{h + \delta}, \quad (7.23)$$

де  $A$  — найближча до точки  $A_h$  точка межі  $\gamma$ ;  $B_h$  — найближча внутрішня точка (на відстані  $h$ );  $\delta$  — відстань від  $A_h$  до  $A$ . Причому  $\delta > 0$ , якщо  $A_h$  знаходиться у середині області  $g$ , і  $\delta < 0$ , якщо точка  $A_h$  зовні  $g$ .

Зокрема, якщо  $\delta = 0$  (точка  $A$  на межі  $\gamma$ ), тобто  $A_h = A$ , то

$$y^{(s+1)}(A_h) \equiv u(A) = \mu(A).$$

Це дає змогу істотно підвищити точність обчислення  $y_{mn}$ .

Починаючи з деякого  $s$ ,  $y^{(s)}(A_h)$  можна не уточнювати, але тоді порядок точності розв'язку буде  $O(h)$ , тому для досягнення заданої точності потрібно використовувати дрібнішу сітку.

### 7.2.2. Наближені методи

Наближений розв'язок еліптичної крайової задачі (7.20) за аналогією з крайовою задачею для звичайного диференціального рівняння можна взяти у такому вигляді:

$$y_n(x; C) = \varphi_0(x) + \sum_{k=1}^n C_k \varphi_k(x) \quad [x = (x_1, x_2)], \quad (7.24)$$

де функція  $\varphi_0(x)$  задовольняє задані крайові умови, а функції  $\varphi_k(x)$  — такі ж самі крайові умови, але однорідні; причому функції  $\varphi_k(x)$  утворюють повну систему лінійно незалежних функцій.

Підставивши вираз (7.24) у рівняння (7.20), одержимо такий вираз для відхилення

$$R(x; C) \equiv \Delta y_n(x; C) + f(x) \neq 0. \quad (7.25)$$

Числові значення вільних параметрів  $C$  визначимо з прийнятих умов близькості відхилення  $R(x; C)$  до нуля.

За **методом колокації** складові  $C$  визначимо з умови обертання на нуль відхилення  $R(x; C)$  у внутрішніх точках  $\xi^{(j)}$  ( $j = \overline{1, n}$ ) області  $g$  (більш-менш рівномірно розміщених та таких, що враховують особливості рівняння), що приводить до такої СЛАР:

$$R(\xi^{(j)}; C) = 0 \quad (j = \overline{1, n}). \quad (7.26)$$

За **методом найменших квадратів** складові  $C$  визначимо з умови мінімальності таких величин:

$$I = \int_g R^2(x; C) dx \quad \text{або} \quad I = \sum_{j=1}^M R^2(\xi^{(j)}; C) \quad (M \gg n),$$

що приводить до СЛАР відносно  $C_k$  ( $k = \overline{1, n}$ ):

$$\partial I / \partial C_k = 0 \quad (k = \overline{1, n}). \quad (7.27)$$

За **методом Бубнова — Гальоркіна** потрібна ортогональність  $R$  до  $\varphi_k(x)$ , тобто

$$\int_g R(x; C) \varphi_k(x) dx = 0 \quad (k = \overline{1, n}). \quad (7.28)$$

що також є СЛАР відносно складових  $C$ .

Одержані СЛАР слід розв'язувати методом Гаусса.

**Приклад 7.2.** Визначимо функцію напружень у разі кручення прямого стержня квадратного перетину:

$$\Delta u(x) = -2, \quad u(x)|_\gamma = 0 \quad [g = (|x_i| \leq 1; i = 1, 2)].$$

1. Спочатку розв'яжемо її **методом колокації** (тут  $\varphi_0(x) \equiv 0$ ).

Очевидно, що базисні функції  $\varphi_k(x)$  для виконання заданих крайових умов повинні мати співмножник  $\Psi(x) = (x_1^2 - 1)(x_2^2 - 1)$  [для обертання їх на нуль на межі  $\gamma = (|x_i| = 1; i = 1, 2)$ ], а  $\varphi_0(x) \equiv 0$ .

Обмежившись у наближеному розв'язку (7.25) одним доданком ( $n = 1$ ), візьмемо його в такому вигляді:

$$y_1(x) = C_1 \Psi(x).$$

Тоді вираз для відхилення (7.26) буде таким:

$$\frac{R(x; C)}{2} \equiv C_1 [(x_1^2 - 1) + (x_2^2 - 1)] + 1.$$

Якщо за вузол колокації взяти точку  $\xi^{(1)} = (0, 0)$ , одержимо рівняння для визначення сталої  $C_1$ :

$$C_1 [(0 - 1) + (0 - 1)] + 1 = 0,$$

звідки  $C_1 = \frac{1}{2}$ .

Отже, матимемо такий наближений розв'язок:

$$y_1(x) = \frac{(x_1^2 - 1)(x_2^2 - 1)}{2}.$$

Якщо наближений розв'язок взяти у вигляді

$$y_2(x) = [C_1 + C_2(x_1^2 + x_2^2)]\Psi(x),$$

а за вузли — точки  $\xi^{(1)} = (0,0)$ ,  $\xi^{(2)} = \left(\pm\frac{1}{2}, 0\right)$  та

$\xi^{(3)} = \left(0, \pm\frac{1}{2}\right)$ , одержимо такі значення сталих:  $C_1 = \frac{31}{54}$ ,

$$C_2 = \frac{2}{27}.$$

Отже, тут наближений розв'язок має вигляд

$$y_2(x) = [31 + 4(x_1^2 + x_2^2)]\frac{\Psi(x)}{54}.$$

Якщо ж наближений розв'язок взяти у вигляді

$$y_3(x) = [C_1 + C_2(x_1^2 + x_2^2) + C_3x_1^2x_2^2]\Psi(x),$$

а до вузлів додати точки  $\xi^{(3)} = \left(\pm\frac{1}{2}, \pm\frac{1}{2}\right)$ , одержимо такі значення сталих:

$$C_1 = \frac{27}{46}, \quad C_2 = \frac{2}{23}, \quad C_3 = \frac{8}{69}.$$

Отже, тут матимемо такий наближений розв'язок:

$$y_3(x) = [81 + 12(x_1^2 + x_2^2) + 16x_1^2x_2^2]\frac{\Psi(x)}{138}. \quad (7.29)$$

Значення наближених розв'язків  $y_1(x)$ ,  $y_2(x)$ ,  $y_3(x)$  у деяких точках першої чверті області  $g$  наведено в табл. 7.2 – 7.4 відповідно.

Таблиця 7.2

$x_1$	$x_2$				
	0,000	0,250	0,500	0,750	1,000
0,000	0,5000	0,4688	0,3750	0,2188	0,0000
0,250	0,4688	0,4395	0,3516	0,2051	0,0000
0,500	0,3750	0,3516	0,2813	0,1641	0,0000
0,750	0,2188	0,2151	0,1641	0,0957	0,0000
1,000	0,0000	0,0000	0,0000	0,0000	0,0000

Таблиця 7.3

$x_1$	$x_2$				
	0,000	0,250	0,500	0,750	1,000
0,000	0,5741	0,5425	0,4444	0,2694	0,0000
0,250	0,5425	0,5127	0,4199	0,2545	0,0000
0,500	0,4444	0,4199	0,3438	0,2081	0,0000
0,750	0,2694	0,2545	0,2081	0,1258	0,0000
1,000	0,0000	0,0000	0,0000	0,0000	0,0000

Таблиця 7.4

$x_1$	$x_2$				
	0,000	0,250	0,500	0,750	1,000
0,000	0,5870	0,5554	0,4565	0,2782	0,0000
0,250	0,5554	0,5258	0,4331	0,2647	0,0000
0,500	0,4565	0,4331	0,3587	0,2211	0,0000
0,750	0,2782	0,2647	0,2211	0,1381	0,0000
1,000	0,0000	0,0000	0,0000	0,0000	0,0000

2. Числовий розв'язок цієї крайової задачі різницевим методом за різницевою схемою (7.22) за нульового на-

ближення  $y_{nm}^{(0)} = \frac{(x_{1n}^2 - 1)(x_{2m}^2 - 1)}{2}$ , кроків  $h_1 = h_2 = \frac{1}{8}$  та

умови закінчення ітерацій  $\max |y_{nm}^{(s+1)} - y_{nm}^{(s)}| \leq 10^{-4}$ , знайдений за 34 ітерації, наведено в табл. 7.5.

Таблиця 7.5

$x_1$	$x_2$				
	0,000	0,250	0,500	0,750	1,000
0,000	0,5817	0,5505	0,4528	0,2760	0,0000
0,250	0,5505	0,5214	0,4300	0,2632	0,0000
0,500	0,4528	0,4300	0,3571	0,2219	0,0000
0,750	0,2760	0,2632	0,2219	0,1422	0,0000
1,000	0,0000	0,0000	0,0000	0,0000	0,0000



Із порівняння даних останніх двох таблиць випливає, що майже у всіх вузлах перші дві значущі цифри розв'язків збігаються (див. дод. 34).

**Приклад 7.3.** Розв'яжемо різницевим методом задачу Діріхле для рівняння Лапласа, що визначає стаціонарний розподіл температури у такій квадратній області:

$$\{g(x) = x_i \in [0,1]; i = 1,2\},$$

тобто

$$\Delta u(x) = 0, \quad u(x)|_{\gamma} = \mu(x) \equiv (x_1 x_2)^2.$$

Різницевий розв'язок цієї крайової задачі за різницевою схемою (7.22) у разі  $y_{nm}^{(0)} = \mu(x_{1n}, x_{2m})$ ,  $h_1 = h_2 = \frac{1}{8}$  за умови закінчення ітерацій  $\max |y_{nm}^{(s+1)} - y_{nm}^{(s)}| \leq 10^{-6}$ , знайдений за 96 ітерацій, наведено в табл. 7.6. (див. дод.35).

Таблиця 7.6

$x_1$	$x_2$				
	0,000	0,250	0,500	0,750	1,000
0,000	0,0000	0,0000	0,0000	0,0000	0,0000
0,250	0,0000	0,0334	0,0670	0,0890	0,0625
0,500	0,0000	0,0670	0,1444	0,2206	0,2500
0,750	0,0000	0,0890	0,2206	0,3946	0,5625
1,000	0,0000	0,0625	0,2500	0,5625	1,0000

### 7.3. Параболічні рівняння

Розглянемо одну з найпростіших задач цього типу — *нестационарну задачу теплопровідності* (7.1), причому спочатку одновимірну задачу (7.1'), тобто коли є лише одна просторова змінна

$$u_t(x, t) = ku_{xx}(x, t) + f(x, t). \quad (7.30)$$

Розв'язок шукатимемо у прямокутній області

$$G = [0 \leq x \leq a, 0 \leq t \leq T]$$

за таких додаткових умов:

$$u(x, 0) = \mu(x) \quad (0 \leq x \leq a); \quad (7.31)$$

$$u(0, t) = \mu_1(t), \quad u(a, t) = \mu_2(t) \quad (0 \leq t \leq T). \quad (7.32)$$

Розглянемо декілька різницевих схем, які реалізують розв'язування різницевим методом задач цього типу. Як і раніше, скористаємося рівномірною сіткою за обома координатами (7.9).

### 7.3.1. Найпростіші різницеві схеми

Для побудови різницевої схеми візьмемо в околі точки  $(x_n, t_m)$  такі шаблони, поклавши для спрощення  $f(x, t) = 0$ :

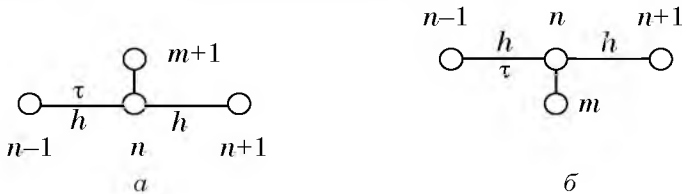


Рис. 7.4

Замінивши похідні у вихідному рівнянні (7.30) на співвідношення (рис. 7.4, а):

$$u_t(x_n, t_m) = \frac{\hat{u}_n - u_n}{\tau} + O(\tau);$$

$$u_{xx}(x_n, t_m) = \frac{u_{n+1} - 2u_n + u_{n-1}}{h^2} + O(h^2),$$

прийдемо до явної різницевої схеми  $[\gamma = h^2 / (k\tau)]$

$$\hat{y}_n = [y_{n-1} - (2 - \gamma)y_n + y_{n+1}] / \gamma$$

$$(n = \overline{1, N-1}; m = \overline{0, M-1}). \quad (7.33)$$

Якщо до неї додати сіткові початкові і крайові умови

$$\begin{aligned} y_n^0 &= \mu(x_n) \quad (n = \overline{0, N}), \quad y_0^{m+1} = \mu_1(t_{m+1}), \\ y_N^{m+1} &= \mu_2(t_{m+1}) \quad (m = \overline{0, M-1}), \end{aligned} \quad (7.34)$$

то задачу знаходження наближених значень  $u(x)$  можна легко розв'язувати, бо на початковому шарі ( $m = 0$ ) розв'язок відомий з початкових умов  $y_n^0 = \mu(x_n)$ , а за інших  $n$  ( $n = \overline{1, N-1}$ ) (7.33) на першому шарі лише одне невідоме ( $y_0^1$  і  $y_N^1$  беремо з крайових умов). За відомими  $y_n^1$  визначимо  $y_n^2$  і т. д. Тому цю різницеву схему називають *явною*.

У разі шаблону (див. рис. 7.4, б) різницева схема має такий вигляд:

$$\frac{\hat{y}_n - y_n}{\tau} = k \frac{\hat{y}_{n+1} - 2\hat{y}_n + \hat{y}_{n-1}}{h^2} \quad (n = \overline{1, N-1}). \quad (7.35)$$

Ця різницева схема містить у собі декілька невідомих на новому шарі, тому її називають *неявною*. Для її розв'язування перемістимо невідомі до лівої частини, унаслідок чого одержимо таку СЛАР разом з (7.34):

$$\hat{y}_{n+1} - (2 + \gamma)\hat{y}_n + \hat{y}_{n-1} = -\gamma y_n \quad (n = \overline{1, N-1}). \quad (7.36)$$

Оскільки ця СЛАР має матрицю тридіагональної структури, то тут доречно скористатись алгебричною прогонкою.

Порядок одержуваного різницевого розв'язку  $u$  за існування неперервних похідних  $u_{tt}$  і  $u_{xxxx}$  дорівнює  $O(\tau + h^2)$ , тобто похибка зменшується з такими порядками, коли  $\tau$  та  $h$  прямують до нуля.

Можна довести, що неявна різницева схема *безумовно стійка*, тобто стійка за довільних співвідношень між  $\tau$  і  $h$ , а явна різницева схема *умовно стійка*, тобто лише в разі виконання умови

$$\gamma \geq 2 \quad \text{або} \quad \tau \leq \frac{h^2}{2k}. \quad (7.37)$$

Крім того, з використаних апроксимацій похідних впливає збіжність наближеного розв'язку до точного.

## 7.3.2. Найпростіша двошарова різницева схема

Розглянемо цю саму параболічну задачу (7.30) – (7.32), але для побудови різницевої схеми скористаємося шеститочковим шаблоном такого вигляду:

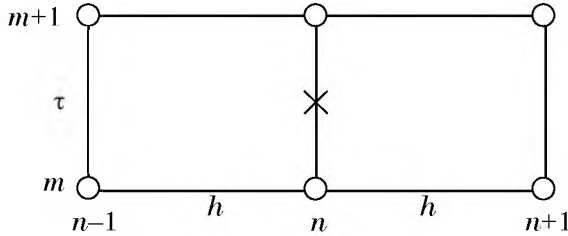


Рис. 7.5

Отримаємо

$$\begin{aligned} \frac{1}{\tau}(\hat{y}_n - y_n) &= \frac{k\sigma}{h^2}(\hat{y}_{n-1} - 2\hat{y}_n + \hat{y}_{n+1}) + \\ &+ \frac{k(1-\sigma)}{h^2}(y_{n-1} - 2y_n + y_{n+1}) + \varphi_n \\ &(n = \overline{1, N-1}; \sigma = \text{const} > 0). \end{aligned} \quad (7.38)$$

Тут невідомих більше, ніж рівнянь. Необхідні ще два рівняння на кожному шарі візьмемо з крайовими умовами (7.32):

$$\hat{y}_0 = \mu_1(\hat{t}), \quad \hat{y}_N = \mu_2(\hat{t}) \quad (\hat{t} = t_{m+1}). \quad (7.39)$$

За значення правої частини  $\varphi_n$  часто беруть

$$\varphi_n = \bar{f}_n = f\left(x_n, t_m + \frac{\tau}{2}\right).$$

Різницева схема (7.38), (7.39) має параметр  $\sigma$ , який слугує ваговим коефіцієнтом біля просторової похідної з верхнього шару. Тому ця різницева схема фактично є *однопараметрична сім'я різницевих схем*, тобто зміною значення параметра  $\sigma$  можна одержати ті чи інші різницеві схеми.

Очевидно, за  $\sigma = 0$  різницева схема (7.38) стає явною різницевою схемою і легко розв'язується, причому розв'язок існує і він єдиний, але стійкий лише за виконання умови (7.37).

Якщо  $\sigma \neq 0$ , різницєва схема є істотно неявна і її можна переписати так:

$$\begin{aligned} & \hat{y}_{n-1} - \left(2 + \frac{\gamma}{\sigma}\right) \hat{y}_n + \hat{y}_{n+1} = \\ & = \left(2 \frac{1-\sigma}{\sigma} - \frac{\gamma}{\sigma}\right) y_n - \frac{1-\sigma}{\sigma} (y_{n-1} + y_{n+1}) - \gamma \tau \varphi_n \\ & \quad (n = \overline{1, N-1}), \end{aligned} \quad (7.40)$$

ураховавши при цьому крайові умови:

$$\hat{y}_0 = \mu_1(\hat{t}), \quad \hat{y}_N = \mu_2(\hat{t}).$$

На кожному шарі різницєва схема (7.40) — СЛАР з невідомими  $\hat{y}_n$ , матриця якої тридіагональна, тому для її розв'язування можна також скористатись алгебричною прогонокою. Тут прогонка стійка, бо діагональні елементи матриці переважають. Отже, різницєвий розв'язок існує і він єдиний, оскільки знаходиться за невелику кількість арифметичних дій.

Якщо  $\sigma = 1$ , різницєва схема *суто неявна*, тобто переходить у (7.36).

Якщо  $\sigma \neq \frac{1}{2}$ , апроксимація має порядок  $O(\tau + h^2)$ , а коли  $\sigma = \frac{1}{2}$  (симетрична різницєва схема) — порядок  $O(\tau^2 + h^2)$ . Можна одержати різницєву схему підвищеної точності  $[O(\tau^2 + h^4)]$ , якщо взяти

$$\sigma = \frac{1}{2} - \frac{h^2}{12\tau k}; \quad \varphi_n = \left( \bar{f} + \frac{h^2 \bar{f}_{xx}}{12} \right)_n.$$

Різницєва схема (7.38) стійка, коли

$$\sigma = \frac{1}{2} - \frac{h^2}{4\tau k}.$$

На практиці слід застосовувати симетричну або підвищеної точності різницєву схему.

## 7.3.3. Багатовимірні задачі

Розглянемо двовимірну задачу теплопровідності зі сталими коефіцієнтами (7.1')

$$u_t(x, t) = k\Delta u(x, t) \quad [x = (x_1, x_2), k = \text{const} > 0]$$

в області

$$G = [0 \leq x_1 \leq a_1; 0 \leq x_2 \leq a_2; 0 \leq t \leq T]$$

за таких додаткових умов:

$$u(x, 0) = \mu(x) \quad [0 \leq x_1 \leq a_1; 0 \leq x_2 \leq a_2];$$

$$u(0, x_2, t) = \mu_1(x_2, t), \quad u(a_1, x_2, t) = \mu_2(x_2, t) \quad (0 \leq t \leq T; 0 \leq x_2 \leq a_2);$$

$$u(x_1, 0, t) = \mu_3(x_1, t), \quad u(x_1, a_2, t) = \mu_4(x_1, t) \quad (0 \leq t \leq T; 0 \leq x_1 \leq a_1).$$

Якщо взяти прямокутну сітку  $(x_{1n}, x_{2m})$  ( $n = \overline{0, N}; m = \overline{0, M}$ ), причому зі сталими кроками  $\left(h_1 = \frac{a_1}{N}, h_2 = \frac{a_2}{M}\right)$ , і скористатись на кожному шарі шаблоном у вигляді хреста, можна побудувати неявну двошарову різницеву схему з вагою  $\sigma$ , яка є узагальнення одновимірної:

$$\frac{1}{\tau}(\hat{y}_{nm} - y_{nm}) = (\Lambda_1 + \Lambda_2)[\sigma \hat{y}_{nm} + (1 - \sigma)y_{nm}], \quad (7.38')$$

де

$$\begin{aligned} \Lambda_1 y_{nm} &= \frac{k}{h_1^2}(y_{n-1, m} - 2y_{nm} + y_{n+1, m}); \\ \Lambda_2 y_{nm} &= \frac{k}{h_2^2}(y_{n, m-1} - 2y_{nm} + y_{n, m+1}). \end{aligned} \quad (7.41)$$

Крім того, потрібно додати крайові умови, тобто задати розв'язок у межових вузлах сітки за  $n = 0, N$  та  $m = 0, M$ .

Легко перевірити, що похибкою апроксимації на розв'язках із неперервними четвертими похідними є  $O(\tau^v + h_1^2 + h_2^2)$ , де

$$v = 2 \text{ за } \sigma = \frac{1}{2} \text{ і } v = 1 \text{ за } \sigma \neq \frac{1}{2}.$$

Умовою стійкості різницевої схеми тут є

$$\sigma \geq \frac{1}{2} - \frac{1}{4k\tau}(h_1^{-2} + h_2^{-2})^{-1}.$$

Це аналог умови стійкості одновимірної задачі теплопровідності.

Якщо  $\sigma = 0$ , різницева схема стає явною і буде стійкою за такої умови:

$$2k\tau \leq (h_1^{-2} + h_2^{-2})^{-1} \approx N^{-2}.$$

Якщо обчислювати за *абсолютно стійким варіантом* різницевої схеми  $\left(\sigma \geq \frac{1}{2}\right)$ , то можна брати  $\tau \approx h$ .

Зрозуміло, що цю різницеву схему можна поширити і на р-вимірну задачу.

**Приклад 7.4.** Визначимо різницевий розв'язок такої задачі теплопровідності:

$$u_t(x,t) = u_{xx}(x,t) + \sin(xt), \quad G = [0 \leq x \leq 1, 0 \leq t \leq 0,02]$$

з додатковими умовами

$$u(x,0) = \cos \frac{\pi x}{2} \quad (0 \leq x \leq 1);$$

$$u(0,t) = 1+t, \quad u(1,t) = t \quad (0 \leq t \leq 0,02),$$

скориставшись явною різницевою схемою (7.33).

Для виконання умови стійкості (7.37), якщо покласти крок по  $x$   $h = 0,2$ , крок  $\tau$  по  $t$  можна взяти не більшим за 0,02.

Значення різницевих розв'язків цієї задачі в однакових вузлах за  $\tau = 0,01$  (див. дод. 36) і  $\tau = 0,0025$  наведено відповідно в табл. 7.7 і 7.8.

Таблиця 7.7

$t$	$x$					
	0,0	0,2	0,4	0,6	0,8	1,0
0,00	1,00000	0,95106	0,80902	0,58779	0,30902	0,00000
0,01	1,01000	0,92778	0,78922	0,57340	0,30146	0,01000
0,02	1,02000	0,91372	0,76995	0,55943	0,29666	0,02000

Таблиця 7.8

$t$	$x$					
	0,0	0,2	0,4	0,6	0,8	1,0
0,00	1,00000	0,95106	0,80902	0,58779	0,30902	0,00000
0,01	1,01000	0,92949	0,78930	0,57348	0,30199	0,01000
0,02	1,02000	0,91737	0,77174	0,56024	0,29781	0,02000

Із порівняння даних цих таблиць випливає, що значення розв'язку різняться декількома одиницями третьої значущої цифри, тобто він прямує до однієї межі.

## 7.4. Гіперболічні рівняння

Визначимо розв'язок гіперболічного рівняння (7.3'), тобто

$$u_{tt}(x,t) = c^2 u_{xx}(x,t) + f(x,t),$$

яке описує малі коливання натягнутої однорідної пружної струни із розподіленим навантаженням  $f(x,t)$ , у прямокутній області

$$G = [0 \leq x \leq a, 0 \leq t \leq T]$$

за таких додаткових умов, наведених нижче:

$$u(x,0) = \mu_1(x), \quad u_t(x,0) = \mu_2(x) \quad (0 \leq x \leq a); \quad (7.42)$$

$$u(0,t) = \mu_3(t), \quad u(a,t) = \mu_4(t) \quad (0 \leq t \leq T). \quad (7.43)$$

Тут шукана функція  $u(x,t)$  — це відхилення струни від стану рівноваги.

Як і в разі параболічних рівнянь, розглянемо лише *різницевий метод* розв'язування цієї задачі на рівномірній сітці (7.9).

### 7.4.1. Схема «хрест»

Взявши шаблон у вигляді *хреста* (рис. 7.6) в околі точки  $(x_n, t_m)$

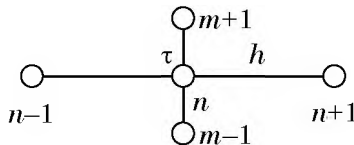


Рис. 7.6

одержимо тришарову РС



$$\frac{1}{\tau^2} \left( \overset{\wedge}{y}_n - 2y_n + \overset{\vee}{y}_n \right) = \left( \frac{c}{h} \right)^2 (y_{n-1} - 2y_n + y_{n+1}) + f_n$$

$$(n = \overline{1, N-1}), \quad (7.44)$$

до якої потрібно додати сіткові крайові умови:

$$y_0 = \mu_3(t), \quad y_N = \mu_4(t). \quad (7.45)$$

Розв'язок  $y$  на нульовому шарі знаходимо з першої початкової умови:

$$y_n^0 = \mu_1(x_n) \quad (n = \overline{0, N}), \quad (7.46)$$

а на першому шарі — із другої, з урахуванням того що

$$\frac{y_n^1 - y_n^0}{\tau} \approx u_t(x_n, 0) = \mu_2(x_n) \quad (n = \overline{1, N-1}).$$

Отже,

$$y_n^1 = \mu_1(x_n) + \tau \mu_2(x_n) \quad (n = \overline{1, N-1}). \quad (7.47)$$

Проте можна одержати кращі результати, якщо покласти

$$\frac{y_n^1 - y_n^0}{\tau} \approx u_t(x_n, 0) + \tau u_{tt}(x_n, 0)/2$$

і виразити  $u_{tt}(x, 0)$  з диференціального рівняння (7.3'). Остаточно отримаємо такий розв'язок на першому шарі:

$$y_n^1 = \mu_1(x_n) + \tau \mu_2(x_n) + \frac{\tau^2}{2} \left( c^2 \frac{d^2 \mu_1(x_n)}{dx^2} + f(x_n, 0) \right)$$

$$(n = \overline{1, N-1}), \quad (7.48)$$

де похідну  $\frac{d^2 \mu_1(x)}{dx^2}$  можна замінювати і на другу поділену різницю. Це *явна різницева схема*, бо  $\hat{y}$  можна виразити через значення  $y$  на двох попередніх шарах, а починаючи з другого шару, знаходити за різницевою схемою (7.44).

Описаний алгоритм показує, що різницевий розв'язок існує і він єдиний: має порядок  $O(\tau^2 + h^2)$ , якщо скористатись фор-

мулою (7.48), і порядок  $O(\tau + h^2)$ , якщо скористатись формулою (7.47).

Різницева схема «хрест» стійка за виконання умови Куранта

$$c\tau \leq h, \quad (7.49)$$

тобто вона умовно стійка. Ця різницева схема забезпечує добру точність для розв'язків  $u$ , які мають неперервні четверті похідні за обома аргументами.

### 7.4.2. Неявна схема

На жаль, у разі умовної стійкості малі порушення умови стійкості (7.49) можуть призвести до переповнення під час розрахунків на ЕОМ, тому перевагу віддають неявним різницевим схемам. Побудуємо для нашої задачі неявну різницеву схему за допомогою шаблона (рис. 7.7):

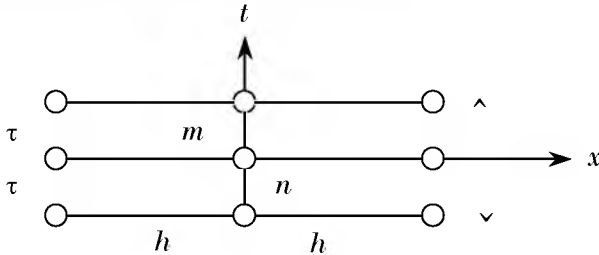


Рис. 7.7

і введемо ваги біля просторових координат на різних шарах. Як наслідок, отримуємо

$$\frac{\hat{y} - 2y + \check{y}}{\tau^2} = \Lambda[\sigma\hat{y} + (1 - 2\sigma)y + \sigma\check{y}] + f, \quad (7.50)$$

де

$$\Lambda y_n = \left[ \frac{c}{h} \right]^2 (y_{n-1} - 2y_n + y_{n+1}) \quad (n = \overline{1, N-1}).$$

Якщо  $\sigma = 0$ , ця різницева схема стає схемою «хрест». Для додатності всіх ваг слід брати  $0 \leq \sigma \leq \frac{1}{2}$ . Як і у схемі «хрест», розв'язок у граничних вузлах визначатимемо за формулами

(7.45), а значення на нульовому та першому шарах — відповідно за формулами (7.46) та (7.47) чи (7.48).

На решті вузлів ця різницєва схема уворює СІАР з тридіагональною матрицею, в якій діагональні елементи переважають. Тому розв'язок, знайдений за допомогою алгебричної прогонки, існує і він єдиний. Похибка при цьому буде  $O(\tau + h^2)$  або  $O(\tau^2 + h^2)$  залежно від застосування формули (7.47) чи (7.48).

Умова стійкості така:

$$(c\tau / h)^2 (1 - 4\sigma) \leq 1. \quad (7.51)$$

Звідси випливає, що за  $\sigma \geq \frac{1}{4}$  різницєва схема *безумовно стійка*. Якщо  $\sigma < 1/4$ , різницєва схема *умовно стійка* за  $c\tau \leq h / \sqrt{1 - 4\sigma}$ .

Отже, за  $1/4 \leq \sigma \leq 1/2$  неявна різницєва схема безумовно збігається до точного розв'язку з похибкою порядку  $O(\tau^2 + h^2)$ . Якщо  $\sigma = 0$ , то ця різницєва схема стає схемою «хрест», а її умова стійкості — в умову Куранта.

Без особливих труднощів за допомогою розглянутих різницєвих схем можна знайти розв'язок гіперболічних рівнянь зі змінними коефіцієнтами.

### 7.4.3. Багатовимірні задачі

Розглянемо побудову різницєвої схеми для рівняння *малих коливань мембрани*

$$u_{tt}(x, t) = c^2 \Delta u(x, t) + f(x, t) \quad [x = (x_1, x_2)]$$

з початковими і крайовим умовами 1-го роду:

$$u(x, 0) = \mu_1(x), \quad u_t(x, 0) = \mu_2(x) \quad [x \in g(x)];$$

$$u(x, t)|_\gamma = \mu_3(x, t) \quad (0 \leq t \leq T).$$

Просторовий аналог неявної різницєвої схеми (7.50) за використання такої самої сітки, як і в п. 7.3.3, записують так:

$$\frac{\hat{y} - 2y + \hat{y}}{\tau^2} = (\Lambda_1 + \Lambda_2)[\sigma \hat{y} +$$

$$+ (1 - 2\sigma)y + \sigma \hat{y}] + f,$$

де

$$\Lambda_1 y_{nm} = \left(\frac{c}{h_1}\right)^2 (y_{n-1,m} - 2y_{nm} + y_{n+1,m});$$

$$\Lambda_2 y_{nm} = \left(\frac{c}{h_2}\right)^2 (y_{n,m-1} - 2y_{nm} + y_{n,m+1}).$$

Для безумовної збіжності цієї різницевої схеми з похибкою  $O(\tau^2 + h_1^2 + h_2^2)$  необхідно виконати умову  $1/4 \leq \sigma \leq 1/2$ .

**Приклад 7.5.** Визначимо розв'язок рівняння малих коливань струни

$$u_{tt}(x,t) = u_{xx}(x,t) + \frac{t}{1+x}, \quad G = [0 \leq x \leq 0,5, 0 \leq t \leq 0,3]$$

з ДУ

$$u(x,0) = \sin \frac{\pi x}{2}, \quad u_t(x,0) = x^2 \quad (0 \leq x \leq 0,5);$$

$$u(0,t) = t^2, \quad u(0,5,t) = \sqrt{2}/2 + t \quad (0 \leq t \leq 0,3).$$

Результати розв'язування розглянутої задачі за  $h = 0,1$  та  $\tau = 0,1$  (див. дод. 30) і  $\tau = 0,025$  (при цьому умова Куранта виконується) за різницевою схемою «хрест» (7.44) та застосування формул (7.46), (7.47) з урахуванням (7.45) наведено відповідно в табл. 7.9 (див. дод. 37) і 7.10.

Таблиця 7.9

$t$	$x$					
	0,000	0,100	0,200	0,300	0,400	0,500
0,000	0,00000	0,15643	0,30902	0,45399	0,58779	0,70711
0,100	0,01000	0,15743	0,31302	0,46299	0,60379	0,80711
0,200	0,04000	0,16749	0,31233	0,46358	0,68303	0,90711
0,300	0,09000	0,19633	0,31972	0,53382	0,76833	1,00711

Таблиця 7.10

$t$	$x$					
	0,000	0,100	0,200	0,300	0,400	0,500
0,000	0,00000	0,15645	0,30902	0,45399	0,58779	0,70711
0,100	0,01000	0,15746	0,31147	0,46054	0,60988	0,80711
0,200	0,04000	0,16600	0,31086	0,47078	0,67457	0,90711
0,300	0,09000	0,19493	0,32902	0,52368	0,76850	1,00711

Із порівняння даних цих таблиць випливає, що розв'язки різняться в основному декількома одиницями третьої значущої цифри.



### Задачі для самостійного розв'язування

1. Різницевим методом у сукупності з методом Гауса розв'яжіть таку еліптичну крайову задачу ( $h_1 = h_2 = 0,2$ ):

$$u_{,11} + u_{,22} = 0; \quad u(x_1, x_2)|_\gamma = |x_1| + |x_2|$$

у квадраті  $g = \{x_1 \in [0,1], i = 1,2\}$ .

2. За допомогою процесу Лібмана розв'яжіть еліптичну крайову задачу ( $h_1 = h_2 = 1, \varepsilon = 1$ )

$$u_{,11} + u_{,22} = 0; \quad u(x_1, x_2)|_\gamma = (x_1 x_2)^2 \quad \text{у крузі } g = \{x_1^2 + x_2^2 \leq 16\}.$$

3. Визначте температуру в точках з координатами (5, 5), (10, 5), (15, 5) прямокутної області  $g = [0 \leq x_1 \leq 20, 0 \leq x_2 \leq 10]$ , три межові відрізки ( $x_1 = 0, x_2 = 0, x_2 = 10$ ) якої теплоізолювані ( $u^0 = 0$ ), а температура четвертого ( $x_2 = 20$ ) — стала ( $u = 100$  °C), розв'язавши задачу Діріхле для рівняння Лапласа, тобто задачу (7.20) за  $f(x) = 0$ .

4. Застосувавши різницевий метод (7.33) для розв'язання рівняння  $u_t = u_{xx}$  з крайовими умовами

$$u(0,t) = 0; \quad u(1,t) = 1 \quad (0 \leq t \leq 0,4) \quad \text{та початковою умовою}$$

$$u(x,0) = \sin \pi x + x \quad (0 \leq x \leq 1)$$

з різними значеннями кроків по  $x$  і  $t$ , проаналізуйте одержані розв'язки.

5. Знайдіть розв'язок рівняння коливань струни  $u_{tt} = u_{xx}$  з нульовими крайовими

$$u(0,t) = 0; \quad u(1,t) = 1 \quad (0 \leq t \leq 0,4)$$

та такими початковими умовами:

$$u(x,0) = \sin \pi x; \quad u_t(x,0) = 0 \quad (0 \leq x \leq 1),$$

застосувавши різницевий метод із різними шаблонами.

## Розділ 8

# ІНТЕГРАЛЬНІ РІВНЯННЯ

### 8.1. Вступні зауваження

Якщо шукана (невідома) функція  $u(x)$  знаходиться під знаком інтеграла, то рівняння називають *інтегральним*. Його самий загальний вигляд в одновимірному випадку такий:

$$\int_a^b K[x, \xi, u(\xi)] d\xi = F[x, u(x)] \quad (a \leq x \leq b), \quad (8.1)$$

де  $x$  — незалежна змінна;  $\xi$  — змінна інтегрування;  $K[x, \xi, u(\xi)]$  і  $F[x, u(x)]$  — відомі функції, які називають відповідно ядром і правою частиною інтегрального рівняння.

Інтегральні рівняння у дечому зручніші від диференціальних, бо, по-перше, містять у собі повну постановку задачі, а по-друге, в інтегральних рівняннях перехід від однієї змінної до багатьох є природний. Так, багатовимірним аналогом (8.1) є рівняння

$$\int_G K[x, \xi, u(\xi)] d\xi = F[x, u(x)];$$
$$x = (x_1, x_2, \dots, x_p) \in G(x), \quad (8.1')$$

яке відрізняється від (8.1) лише тим, що інтегрування проводять у багатовимірній області  $G$ . Оскільки обидва рівняння не потребують додаткових умов і повністю визначають задачу, аналогія є повна. Водночас у диференціальних рівняннях перехід від однієї до декількох (від звичайних диференціальних рівнянь до рівнянь в частинних похідних) призводить до принципових ускладнень та нових постановок і методів розв'язання.

До інтегральних рівнянь зводиться багато прикладних задач. Крім того, до інтегральних рівнянь можна звести задачі для диференціальних рівнянь.

Найбільш вивченими є такі лінійні інтегральні рівняння:

а) *Фредгольма 2-го роду*:

$$u(x) - \lambda \int_a^b K(x, \xi) u(\xi) d\xi = f(x). \quad (8.2)$$

ядро якого  $K(x, \xi)$  задано на квадраті  $a \leq x \leq b$ ,  $a \leq \xi \leq b$ , а  $\lambda$  — числовий параметр;

б) *Вольтерра 2-го роду*:

$$u(x) - \lambda \int_a^x K(x, \xi) u(\xi) d\xi = f(x). \quad (8.3)$$

ядро якого задано на трикутнику  $a \leq \xi \leq x \leq b$ .

Якщо шукана функція  $u$  знаходиться лише під знаком інтеграла, то ці рівняння називають *інтегральними рівняннями 1-го роду*. Вони *некоректно поставлені*, а отже, не будуть розглядатися.

Якщо в інтегральному рівнянні Фредгольма 2-го роду  $f(x) = 0$ , приходимо до задачі на *власні значення для ядра*  $K(x, \xi)$

$$u(x) - \lambda \int_a^b K(x, \xi) u(\xi) d\xi = 0. \quad (8.4)$$

У цій задачі потрібно знайти значення  $\lambda = \lambda_i$ , для яких це рівняння має ненульові розв'язки  $u(x) = \varphi_i(x)$ . Ці  $\lambda_i$  називають *характеристичними числами*, а величини  $\tilde{\lambda} = \frac{1}{\lambda_i}$  — *власними значеннями*; відповідні їм ненульові розв'язки  $\varphi_i(x)$  — *власними функціями* ядра інтегральні рівняння [точніше, ядра  $K(x, \xi)$ ].

Якщо ядро дійсне і симетричне  $[K(x, \xi) = K(\xi, x) = K^*(x, \xi)]$ , воно має хоча б одне власне значення. Усі власні значення та-

кого ядра дійсні, а власні функції ортогональні одна одній, але система власних функцій може бути неповною і скінченною.

Якщо  $\lambda$  в (8.2) не збігається з жодним із власних значень, розв'язок інтегрального рівняння Фредгольма 2-го роду існує і притому єдиний. Коли ж  $\lambda = \lambda_i$  у класі неперервних або досить гладких правих частин  $f(x)$ , задача для (8.2) є *некоректно поставлена*.

Інтегральне рівняння Вольтерра не має власних значень, бо за  $f(x) = 0$  воно матиме лише тривіальний розв'язок  $u(x) = 0$ . Тому неоднорідне інтегральне рівняння Вольтерра 2-го роду (8.3) завжди має розв'язок і притому єдиний.

## 8.2. Різницевий метод

### 8.2.1. Нелінійні рівняння

У разі розв'язування інтегрального рівняння (8.1) *різницевим методом* (метод скінченних сум або квадратур) інтеграл замінюють на квадратурну формулу, що приводить до такої системи алгебричних рівнянь відносно невідомих значень різницевого розв'язку  $u(x)$  у вузлах КФ:

$$\sum_{m=1}^N C_m K(x_n, x_m, y_m) = F(x_n, y_n) \quad (n = \overline{1, N}), \quad (8.5)$$

де  $u(x_n) \approx y_n$ , а  $x_m$  і  $C_m$  — вузли і ваги квадратурної формули.

Для одержання прийнятної точності різницевого розв'язку краще скористатись квадратурною формулою Гаусса, а систему алгебричних рівнянь розв'язувати методом Ньютона.

**Приклад 8.1.** Розглянемо застосування різницевого методу на прикладі розв'язування такого нелінійного інтегрального рівняння:

$$\lambda \int_0^1 x \xi u^2(\xi) d\xi = u(x) \quad (0 \leq x \leq 1),$$

яке має такі точні розв'язки:  $u_1(x) = 0$ ;  $u_2(x) = 4x / \lambda$ , де  $\lambda$  будь-яке дійсне число.



Отже, скористаємося різницеvim методом, тобто замі-  
нимо інтеграл квадратурною формулою з двома вузлами,  
що приведе до такої системи алгебричних рівнянь:

$$kx_1(x_1y_1^2 + x_2y_2^2) = y_1; \quad kx_2(x_1y_1^2 + x_2y_2^2) = y_2$$

$$(h = b - a) / 2, \quad k = h\lambda).$$

Після нескладних перетворень матимемо таку систе-  
му алгебричних рівнянь:

$$a_{11}y_1^2 + a_{12}y_2^2 = y_1 / k; \quad a_{21}y_1^2 + a_{22}y_2^2 = y_2 / k$$

$$(a_{11} = x_1^2, \quad a_{12} = a_{21} = x_1x_2, \quad a_{22} = x_2^2).$$

Розв'язавши цю систему, знайдемо значення різни-  
цевого розв'язку у вибраних вузлових точках

$$y_1^{(1)} = 0; \quad y_2^{(1)} = 0; \quad y_1^{(2)} = a_{12} / (a_{22}^2 + a_{11}a_{12}) / k;$$

$$y_2^{(2)} = \sqrt{y_1(1/k - a_{11}y_1) / a_{12}}.$$

Застосовуючи два значення різницевого розв'язку,  
знайдемо наближений вираз шуканої функції  $u(x)$  у  
вигляді інтерполяційного полінома Ньютона 1-го степеня:

$$u(x) \approx y(x) \equiv y(x_1) + (x - x_1)(y(x_2) - y(x_1)) / (x_2 - x_1).$$

Якщо вдатися до квадратурної формулу середніх, тоді

$$F_h^C \approx h \sum_{i=1}^2 f_i \quad [f_i = f(x_i), \quad x_1 = a + h/2, \quad x_2 = b - h/2].$$

Якщо ж скористатися формулою Гаусса, тоді

$$F^\Gamma \approx h \sum_{i=1}^2 \gamma_i f(x_i)$$

$$[x_i = (a + b) / 2 + h \xi_i \quad (i = \overline{1,2}); \quad \gamma_1 = \gamma_2 = 1, \quad \xi_1 = -\xi_2 = \sqrt{3} / 3]$$

Якщо  $a=0, b=1$ , тоді  $h=1/2$  і вузли у квадратурній  
формулі Гаусса такі:

$$x_1 = (1 - \sqrt{3} / 3) / 2, \quad x_2 = (1 + \sqrt{3} / 3) / 2.$$

Точні значення розв'язку та знайдені за інтерполяційним полінома Ньютона 1-го степеня наведено в табл. (8.1) (див. дод. 38):

Таблиця 8.1

	x				
	0,00000	0,25000	0,50000	0,75000	1,00000
ТР	0,00000	1,00000	2,00000	3,00000	4,00000
РР ФГ	0,00000	1,00000	2,00000	3,00000	4,00000
РР ФС	0,00000	1,14286	2,28571	3,42857	4,57143

*Примітка.* Тут ТР — точний розв'язок, РР ФГ — різницевий розв'язок за квадратурною формулою Гаусса, РР ФС — різницевий розв'язок за квадратурною формулою середніх

### 8.2.2. Лінійні рівняння

**Задача на власні значення.** Якщо у (8.4) замінити інтеграл на квадратурну формулу, дійдемо такої однорідної СЛАР:

$$\sum_{m=1}^N C_m K_{nm} y_m - \frac{y_n}{\lambda} = 0 \quad (n = \overline{1, N}), \quad (8.6)$$

де  $K_{nm} = K(x_n, x_m)$ .

Це задача на власні значення для матриці. Очевидно, ця матриця має  $N$  власних значень  $[\lambda_i^{(N)}]^{-1}$  ( $i = \overline{1, N}$ ), які є наближеннями до перших власних значень для ядра  $K(x, \xi)$ . Тут матриця  $K_{nm}$  — щільно заповнена й різницевий метод дає прийнятні результати для  $N$  до 50.

**Приклад 8.2.** Розглянемо застосування різницевого методу на прикладі розв'язування такого однорідного інтегрального рівняння :

$$u(x) - \lambda \int_0^1 x \xi^2 u(\xi) d\xi = 0 \quad (0 \leq x \leq 1),$$

яке має такий точний розв'язок:  $\lambda = 4$ .

Замінімо інтеграл квадратурною формулою середніх з вузлами

$x_1 = 1/4$ ,  $x_2 = 3/4$  й обчислимо в них значення ядра:

$$K_{11} = 1/4(1/4)^2 = 1/64; \quad K_{12} = 1/4(3/4)^2 = 9/64;$$

$$K_{21} = 3/4(1/4)^2 = 3/64; \quad K_{22} = 3/4(3/4)^2 = 27/64.$$

Отже, матимемо

$$\sum_{m=1}^N C_m K_{nm} y_m - \tilde{\lambda} y_n = 0$$

$$\left[ h = (b - a) / 2; \quad K_{nm} = K(x_n, x_m); \quad \tilde{\lambda} = 1 / \lambda; \quad n = \overline{1, N} \right].$$

Це приводить до такої однорідної СЛАР:

$$n = 1: \quad \sum_{m=1}^2 C_m K_{1m} y_m - \tilde{\lambda} y_1 = 0;$$

$$n = 2: \quad \sum_{m=1}^2 C_m K_{2m} y_m - \tilde{\lambda} y_2 = 0.$$

$$\begin{pmatrix} 1/128 - \tilde{\lambda} & 9/128 \\ 3/128 & 27/128 - \tilde{\lambda} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Знайдемо визначник цієї СЛАР, покладаючи  $z = 128\tilde{\lambda}$ :  $(1 - z)(27 - z) - 27 = 0$ . Звідси  $z_1 = 0$ ,  $z_2 = 28$ , а отже, наближене значення характеристичного числа таке:  $\lambda = 32/7 = 4,5714$ . За  $N=3$  і 4 значення будуть  $\lambda = 4,3200$  і  $\lambda = 4,1290$  відповідно.

Таким чином, за чотирьох відрізків похибка складає менше 3%.

**Неоднорідне інтегральне рівняння Фредгольма 2-го роду.** Замінивши в (8.2) інтеграл на квадратурну формулу, дійдемо такої СЛАР:

$$y_n - \lambda \sum_{m=1}^N C_m K_{nm} y_m = f_n \quad \left[ f_n = f(x_n); \quad n = \overline{1, N} \right]. \quad (8.7)$$

Одержану СЛАР з  $N$  до 150 можна розв'язати методом Гауса. Вона має єдиний розв'язок, якщо  $\lambda \neq \lambda_i^{(N)}$  ( $i = \overline{1, N}$ ) і різниця між наближеним та точним розв'язками незначна. Отже, цей алгоритм *добре обумовлений*.

Якщо ж  $\lambda \approx \lambda_i$ , то система (8.7) стає погано обумовленою й різницевий розв'язок може істотно відрізнитися від точного. Оскільки, як правило, власні значення  $\lambda_i^{-1}$  невідомі, то розрахунки слід проводити на згущуваних сітках, якщо ж у випадку згущення сітки  $y_n$  збігаються до однієї межі, то вона й буде розв'язком  $u$ . Якщо ж за  $h \rightarrow 0$  наближений розв'язок  $y_n$  не збігаються до однієї межі, то це означає, що  $\lambda \approx \lambda_i$ , тобто є характеристичне число ядра інтегрального рівняння.

**Приклад 8.3.** Визначимо різницевий розв'язок такого інтегрального рівняння Фредгольма 2-го роду:

$$u(x) + \int_0^1 x e^{x\xi} u(\xi) d\xi = e^x.$$

Результати розв'язання цього рівняння за квадратурною формулою середніх для трьох кроків інтегрування після застосування виразу

$$u(x) = f(x) + \lambda \sum_{m=1}^N C_m K(x, x_m) y_m$$

наведено в табл. 8.2 (див. дод. 39):

Таблиця 8.2

$h$	$x$				
	0,000	0,250	0,500	0,750	1,000
1,000	1,00000	0,99958	1,00409	1,02130	1,06282
0,500	1,00000	0,99981	1,00008	1,00166	1,00588
0,250	1,00000	0,99993	1,00002	1,00057	1,00207
0,125	1,00000	0,99998	1,00000	1,00017	1,00063

Зазначимо, що точним розв'язком цього інтегрального рівняння є  $u(x) \equiv 1$ .

Система (8.7) трансформується у систему для розв'язування **інтегрального рівняння Вольтерра 2-го роду** різницевим методом, якщо у сумі (8.7)  $N$  замінити на  $n$ , бо його ядро  $K(\cdot)$  задано на трикутнику

$$y_n - \lambda \sum_{m=1}^n C_m K_{nm} y_m = f_n \quad (n = \overline{1, N}). \quad (8.8)$$

Ця СЛАР має трикутну матрицю і тому її легко розв'язувати *зворотним ходом* методу Гаусса. До того ж обсяг обчислень невеликий і  $N$  може набувати значення до 1000, тобто можна одержати розв'язок з дуже високою точністю.

**Приклад 8.4.** Визначимо різницевого розв'язку такого інтегрального рівняння Вольтерра 2-го роду:

$$u(x) - \lambda \int_0^x u(\xi) d\xi = 1 \quad (0 \leq x \leq 1),$$

яке має точний розв'язок:  $u(x) = e^x$  за  $\lambda = 1$ .

Розв'яжемо його різницевим методом. Тут СЛАР може бути приведено до такого вигляду:

$$-\lambda \sum_{m=1}^{n-1} C_m K_{nm} y_m + (1 - \lambda C_n K_{nn}) y_n = f_n \quad (n = \overline{1, N}).$$

Звідси видно, що матриця СЛАР трикутна, що дозволяє послідовно знаходити розв'язок за рекурентною формулою, коли  $(1 - \lambda C_n K_{nn}) \neq 0$   $y_1 = f_1$  на підставі (8.8):

$$y_n = (f_n + \lambda \sum_{m=1}^{n-1} C_m K_{nm} y_m) / (1 - \lambda C_n K_{nn}) \quad (n = \overline{2, N}).$$

У випадку застосування квадратурної формули трапецій ці співвідношення набувають вигляду

$$y_1 = f_1;$$

$$y_n = \left( f_n + \lambda h K_{n1} y_1 / 2 + h \lambda \sum_{m=2}^{n-1} K_{nm} y_m \right) / (1 - \lambda h K_{nn} / 2) \quad (n = \overline{2, N})$$

Розглянемо цю систему з такими вузлами:

$$x_n = 0, 1/4, 1/2, 3/4, 1$$

та обчислимо в них значення правих частин та ядра.

Очевидно, що всі вони дорівнюють одиниці, тобто  $f_n \equiv f(x_n) = 1$ ;  $K_{nm} \equiv K(x_n, x_m) = 1$ . Це приводить до такого ланцюга обчислень ( $h = 1/4$ ):

$$\begin{aligned}
 y_1 &\equiv y(0) = f_1 = 1; \\
 y_2 &\equiv y(0.25) = (f_2 + hK_{21}y_1 / 2) / (1 - hK_{22} / 2) = \\
 &= [1 + (1/4) / 2] / (7/8) = 9/7 = 1,2857143; \\
 y_3 &\equiv y(0.5) = (f_3 + hK_{31}y_1 / 2 + hK_{32}y_2) / (1 - hK_{33} / 2) = \\
 &= [1 + (1/4) / 2 + (1/4)9/7] / (7/8) = 81/49 = 1,6530612; \\
 y_4 &\equiv y(0.75) = [f_4 + hK_{41}y_1 / 2 + h(K_{42}y_2 + \\
 &+ K_{43}y_3) / (1 - hK_{44} / 2) = [1 + (1/4) / 2 + (1/4)(9/7 + \\
 &+ 81/49)] / (7/8) = 729/343 = 2,1253644; \\
 y_5 &\equiv y(1) = [f_5 + hK_{51}y_1 / 2 + h(K_{52}y_2 + K_{53}y_3 + \\
 &+ K_{54}y_4)] / (1 - hK_{44} / 2) = 2,7326114.
 \end{aligned}$$

Результати вищезнайдених розрахунків наведено в табл. 8.3 (див. дод. 40).

Таблиця 8.3

	x				
	0.000	0.250	0.500	0.750	1.000
$u(x)$	1.00000	1.28403	1.64872	2.11700	2.71828
$y(x)$	1.00000	1.28571	1.65306	2.12536	2.73261

Як бачимо, середнє квадратичне відхилення наближеного розв'язку  $y$  від точного  $u$  складає 0,0077.

### 8.3. Наближені методи

#### 8.3.1. Метод простих ітерацій

Метод простих ітерацій — найпростіший наближений метод розв'язування інтегральних рівнянь. Його розрахункові формули для інтегрального рівняння Фредгольма 2-го роду (8.2) мають вигляд

$$y_0(x) = 0, \quad y_{s+1}(x) = f(x) + \lambda \int_a^b K(x, \xi) y_s(\xi) d\xi \quad (s = 0, 1, \dots).$$

За обмежених ядер  $K(x, \xi)$  та досить малих  $|\lambda|$  наближений розв'язок збігається до точного.

**Приклад 8.5.** Застосувавши метод простих ітерацій до інтегрального рівняння:

$$u(x) - \lambda \int_0^{\infty} e^{-(x+\xi)} u(\xi) d\xi = x,$$

одержимо такі послідовні наближення до розв'язку:

$$y_0(x) = 0; \quad y_1(x) = x;$$

$$y_2(x) = x + \lambda e^{-x}; \quad y_3(x) = x + \lambda e^{-x} + \frac{\lambda^2 e^{-x}}{2}, \dots$$

або

$$u(x) = x + \lambda \left( 1 + \frac{\lambda}{2} + \frac{\lambda^2}{4} + \dots \right) e^{-x} = x + \frac{2\lambda}{2-\lambda} e^{-x}.$$

Наближений розв'язок у цьому випадку збігається до точного лише у разі  $\lambda < 2$ .

### 8.3.2. Метод Бубнова — Гальоркіна

Розв'язок інтегрального рівняння (8.2) шукатимемо у такому вигляді:

$$u(x) \approx y_n(x) = f(x) + \lambda \sum_{k=1}^n C_k \varphi_k(x), \quad (8.9)$$

де  $\{\varphi_k(x)\}$  — повна система лінійно незалежних функцій.

Підставляючи наближений розв'язок (8.9) у неоднорідне рівняння Фредгольма (8.2) і вимагаючи ортогональності відхилення  $R$  до всіх  $\varphi_k(x)$  ( $k = \overline{1, n}$ ), тобто

$$\int_a^b \left[ y(x) - \lambda \int_a^b K(x, \xi) y(\xi) d\xi - f(x) \right] \varphi_m(x) dx = 0 \quad (m = \overline{1, n}),$$

одержимо таку СЛАР для визначення  $C_k$ :

$$\sum_{k=1}^n a_{mk} C_k = b_m \quad (m = \overline{1, n}),$$

$$a_{mk} = \int_a^b \varphi_m(x) \varphi_k(x) dx - \lambda \int_a^b \int_a^b K(x, \xi) \varphi_m(x) \varphi_k(\xi) dx d\xi, \quad (8.10)$$

$$b_m = \int_a^b \varphi_m(x) f(x) dx.$$

Для задачі на власні значення потрібно покласти у (8.10)  $f(x) = 0$ .

Метод Бубнова-Гальборкіна можна застосовувати й до нелінійних інтегральних рівнянь (8.1), але тоді він приведе до системи алгебричних рівнянь.

Оскільки у вираз (8.10) входять двократні інтеграли, у (8.9) обмежимося малим числом доданків.

**Приклад 8.6.** Методом Бубнова — Гальборкіна знайти наближений розв'язок інтегрального рівняння

$$u(x) - \int_{-1}^1 x\xi u(\xi) d\xi = x.$$

За повну систему функцій  $\varphi_k(x)$  на відрізку  $[-1, 1]$  виберемо поліноми Лежандра  $\varphi_k(x) = P_k(x)$ . Обмежувачись трьома доданками у формулі (8.9) та підставляючи вирази для  $\varphi_k(x)$  до (8.10), матимемо

$$\begin{aligned} C_1 \cdot 1 + C_2 x + C_3 (3x^2 - 1) / 2 = \\ = x + \int_{-1}^1 x\xi (C_1 \cdot 1 + C_2 \xi + C_3 (3\xi^2 - 1) / 2) d\xi = x \end{aligned}$$

або

$$C_1 \cdot 1 + C_2 x + C_3 (3x^2 - 1) / 2 = x + C_1 \cdot x \cdot 2 / 3.$$

Помноживши обидві частини цього виразу послідовно на 1,  $x$ ,  $(3x^2 - 1) / 2$  та проінтегрувавши по  $x$  в межах від -1 до 1, знайдемо

$$2 \cdot C_1 = 0, \quad 2 \cdot C_2 x / 3 = 2 / 3 + 4C_2 / 9, \quad 2 \cdot C_3 / 5 = 0.$$

Звідси  $C_1 = 0$ ,  $C_2 = 3$ ,  $C_3 = 0$ , а отже,  $y_3(x) = 3x$ . Не складно перевірити, що це точний розв'язок інтегрального рівняння (8.11).





---

**Задачі для самостійного розв'язування**

---

1. Знайдіть розв'язок інтегрального рівняння

$$u(x) - 0,25 \int_0^2 (\cos 2x - \xi^2) u(\xi) d\xi = x + 4$$

різницевим методом.

2. За методами простих ітерацій та Бубнова — Гальоркіна знайдіть розв'язок інтегрального рівняння

$$u(x) - \lambda \int_0^1 e^{-(x+\xi)} u(\xi) d\xi = x.$$

---

## Додатки

---

### Навчальні Scilab-програми, які реалізують деякі числові методи, викладені у підручнику, та результати їх виконання для наведених прикладів

Вихідні дані для Scilab-програм задаються операторами присвоювання. Імена змінних у сайлаб-програмах та тексти підручника, як правило, мають спільні елементи, що істотно спрощує розуміння структури програм та полегшує їх використання.

У всіх програмах, де можливе графічне зображення результатів їх виконання, воно передбачене у додатках, але для скорочення вони надані лише в додатках 1 й 33. У всіх інших додатках відповідні команди закоментовані (тобто перед ними стоять //). Якщо прибрати коментар //, відповідні графічні зображення будуть надруковані. Для зменшення обсягів коментарів у деяких додатках уведено скорочення.

#### Додаток 1

Знаходження наближених значень функції  $y(x)$  з прикл. 1.2 за допомогою інтерполяційного полінома Ньютон (ІПН).

```
//ipn
// Побудова ІПН на нерівномірній сітці за (1.4)
function [f] = ff(x) ; f = sin(%pi*x/6) ;
endfunction ;
// Задана функція
function [pln] = plnwt(n, xp, xx, yy) ;
//функція plnwt обчислення ІПН у точці xp
y=yy(n+1) ;
for k=1:n n1=n-k+1 ; y=yy(n1)+y*(xp-xx(n1)) ; end ;
pln=y ;
endfunction ;
//функція ipn формування значень ІПН у точках xd
function [Pn, y, x] = ipn(n, M, xx, yy, xd) ;
// n - степінь ІПН, M - число точок у масиві xd
//Знаходження поділених різниць
m=n+1 ;
for k=1:m nt=n-k+1 ;
for i=1:nt n1=n-i+1 ; n2=n1+1 ;
yy(n2)=(yy(n2)-yy(n1))/(xx(n2)-xx(n1-k+1)) ;
end
end ;
// Обчислення ІПН у точках xd
for k=1:M xp=xd(k) ; p(k)=plnwt(n, xp, xx, yy) ;
x(k)=xp ; y(k)=ff(xp) ; end ;
// Перетворення векторів на матриці-рядки
x=x.' ; y=y.' ; Pn=p.' ;
endfunction ;
```

```

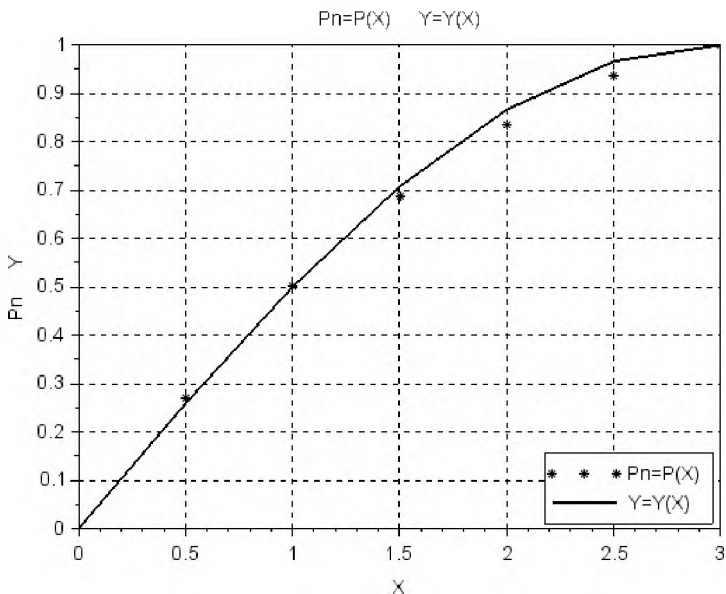
// main program
// Вихідні дані для побудови ІПН
xx=[0,1.0,3]; yy=[0,0.5,1];
// Координати точок для таблиць і графіків
xd=[0,0.5,1,1.5,2.,2.5,3];
// Коментар до фактичних параметрів функції ipn
//[Pn,y,x] = ipn(n,M,xx,yy,xd)
[Pn,y,x] = ipn(2,7,xx,yy,xd)
//y(4)-точно, а Pn(4)-наближене значення y(3/2)
dlt=abs((y(4)-Pn(4))/y(4))*100
//dlt - Відносна похибка обчислення ІПН
// Побудова графіків заданої функції та ІПН
plot(x,Pn,'k*',x,y,'k-'); xgrid();
xlabel(' Pn=P(X)      Y=Y(X) ', 'X', 'Pn      Y');
legend(' Pn=P(X) ', ' Y=Y(X) ', 4, %t);

```

Результати виконання *ipn* у табличному (збережено чотири значущі цифри за десятичною крапкою) вигляді:

x	=	0.	0.5	1.	1.5	2.	2.5	3.
y	=	0.	0.2588	0.5	0.7071	0.8660	0.9659	1.
Pn	=	0.	0.2708	0.5	0.6875	0.8333	0.9375	1.
dlt	=	2.77%						

та графічному вигляді:



## Додаток 2

Знаходження наближених значень функції  $y(x)$  з прикл. 1.3 за допомогою вільних кубічних сплайнів (ВКС).

```
// splncub
// Побудова ВКС на нерівномірній сітці
function [ac] = spln3(n, na, xx, yy);
// Функція spln3 обчислення коефіцієнтів ВКС
c(1)=0; aa(1)=0; na1=na+1; n1=n+1;
c(n1)=0; m=n-1;
for i=1:n h(i)=xx(i+1)-xx(i); end;
for i=2:n
aa(i-1)=h(i-1); bb(i-1)=-2*(h(i-1)+h(i));
cc(i-1)=h(i); ayp = yy(i+1); ay0 = yy(i);
aym = yy(i-1);
dd(i-1)=(ayp-ay0)/h(i)-(ay0-aym)/h(i-1))*3;
end;
// Обчислення коефіцієнтів c ВКС
[z] = algprog(na1, aa, bb, cc, dd);
// Формування масиву ac(n,4) коефіцієнтів ВКС
for i=1:m c(i+1)=z(i); end;
for i=1:n
// ac(i,1) - коефіцієнти a ВКС
ac(i,1)=yy(i); ayp = yy(i+1); ay0 = yy(i);
// ac(i,2) - коефіцієнти b ВКС
ac(i,2)=(ayp-ay0)/h(i)-...
h(i)*(c(i+1)+c(i)*2)/3;
// ac(i,3) - коефіцієнти c ВКС
ac(i,3)=c(i);
// ac(i,4) - коефіцієнти d ВКС
ac(i,4)=(c(i+1)-c(i))/(3*h(i));
end;
endfunction;
// Розв'язання СЛАР алгебричною прогонкою
function [x] = algprog(n1, a, b, c, d);
n=n1-1; k(1)=0; h(1)=0; c(n)=0;
a(1)=0; k(n1)=0; x(n1)=0;
for i=1:n ai =a(i); zn =b(i)-ai*k(i);
k(i+1)=c(i)/zn; h(i+1)=(ai*h(i)-d(i))/zn;
end;
j=n1;
for i=1:n j =j-1; x(j)=k(j+1)*x(j+1)+h(j+1);
end;
endfunction;
```

```

function [s,xd] = splncub(n,M,N,a,b,xx,yy,xd) ;
// Функція splncub формування ВКС у xd
h=(b-a)/(N-1);
// Обчислення ВКС у вузлах інтерполяції
na=n-1;
// Обчислення коефіцієнтів ВКС за spln3
[ac] = spln3(n,na,xx,yy) ;
for j=1:M xj=xd(j); x(j)=xj;
// Установлення розміщення точки xj
for i=1:n if(xj>=xx(i)&xj<xx(i+1)) then
k=i; break; end
end;
// Обчислення ВКС у точці xj
ss=ac(k,4); t=xj-xx(k);
for i=1:3 ss=ss*t+ac(k,4-i); end;
// формування значень ВКС у точках xd
s(j)=ss;
end;
// Перетворення вектора на матрицю-рядок
s=s.';
endfunction;
// main program
// Вихідні дані для побудови ВКС
xx=[-1,0,1,2]; yy=[-2,1,4,15];
// Координати точок для таблиць і графіків
xd=[-1,-0.5,0,0.5,1,1.5,2];
// Коментар до фактичних параметрів splncub
//[s,xd] = splncub(n,M,N,a,b,xx,yy,xd) ;
[s,xd] = splncub(3,7,4,-1,2,xx,yy,xd)
// Побудова графіка ВКС
//x=xd.';
//plot(x,s,'k*'); xgrid();
//xtitle(' S=S(X) ','S ');
//legend(' S=S(X) ',4,&t)

```

Результати виконання *splncub* у табличному вигляді:

```

x = - 1.   - 0.5   0.   0.5   1.   1.5   2.
s = - 2.   - 0.3   1.   1.9   4.   8.7  15.

```

### Додаток 3

Знаходження наближених значень функції  $u=u(x,t)$  за допомогою інтерполяційного полінома 1-го степеня (1.9) з прикл. 1.5.

```

// linint2z
function [fi,t,x]=linintf2z(x,t,vx,vt,vy) ;
v=[1;1;1];

```

```

M=[vx vt vy]; D0=det(M) ;
M1=[vt vy v ]; D1=det(M1);
M2=[vx vy v ]; D2=det(M2);
M3=[vx vt v ]; D3=det(M3);
fi = (D0-D1*x+D2*t)/D3;
endfunction;
//main program
vx=[0;2;4];
vt=[0;4;-2];
vy=[0;-3;1];
//[fi,t,x]=linintf2z(x,t,vx,vt,vy);
[fi,t,x]=linintf2z(2,1,vx,vt,vy)
Результати виконання linint2z:
x = 2. t = 1. fi = - 0.9

```

#### Додаток 4

Знаходження наближених значень функції  $y(x)$  з прикл. 1.7 та коефіцієнтів лінійної функції, яка її апроксимує, за допомогою методу найменших квадратів (МНК).

```

// mnk
function [f] = ff(x); f = x^2; endfunction;
// Задана функція
function [dskv,C,y,fi,x] = mnk(n,N,M,a,b);
h=(b-a)/(N-1); m=n+1;
// Обчислення вузлів та значень y(x) для МНК
for i=1:N xx(i)=a+(i-1)*h; yy(i)=ff(xx(i)); end;
//Формування масивів для побудови А і В
n2=n*2; n1=n+1;
for j=1:n2 s=0;
for i=1:N s=s+xx(i)^j; end; p(j)=s; end;//j
for i=1:n1 j1=1+(i-1)*n1; j2=i*n1; k=i-j1-1;
for j=j1:j2 if (j==1) then d(1)=N; end;
if (~j==1) then d(j)=p(j+k); end; end; end;
// формування вектора ПЧ В СЛАР (1.13)
s=0; for i=1:N s=s+yy(i); end; B(1)=s;
for i=2:n1 s=0;
for j=1:N s=s+yy(j)*xx(j)^(i-1); end;
B(i)=s; end;
// формування матриці А СЛАР (1.13)
for i=1:m for j=1:m A(i,j)=d(i+m*(j-1)); end;
end;
// Розв'язання СЛАР (1.13) методом Гаусса
CC=rref([A B]); [NN,MM]=size(CC); C=CC(:,MM);

```

```

// Обчислення ДСКВ
dskv=0; for i=1:N xi=xx(i); fi =C(1);
    for j=1:n fi=fi+C(j+1)*xi^j; end;
dskv=dskv+(fi-yy(i))^2; end;
dskv=sqrt(dskv/N); // ДСКВ (dskv) - дискретне
// середнє квадратичне відхилення
// Обчислення значень у(х) та АФ fi
h=(b-a)/(M-1);
for i=1:M x(i)=a+(i-1)*h; xi=x(i); ffi=C(1);
    for j=1:n ffi=ffi+C(j+1)*xi^j; end;
    fi(i)=ffi; y(i)=ff(xi);
    end;
// Перетворення векторів на матриці-рядки
x=x.'; fi=fi.'; y=y.'; C=C.';
endfunction;
// main program
// Коментар до фактичних параметрів функції mnk
//[dskv,C,y,fi,x] = mnk(n, N,M,a,b)
    [dskv,C,y,fi,x] = mnk(1, 4,5,0,1)
        pp=poly(C, 'x', 'c')
// Побудова графіків заданої функції та АФ
//plot(x,fi,'k.',x,y,'k-'); xgrid();
//xtitle(' FI=FI(X) Y=Y(X)', 'X', 'FI Y');
//legend(' FI=FI(X) ', ' Y=Y(X) ', 4, &t)

```

Результати виконання *mnk* у табличному (збережено чотири значущі цифри за десятичною крапкою) вигляді:

```

x = 0. 0.25 0.5 0.75 1.
fi = -0.1111 0.1389 0.3889 0.6389 0.8889
y = 0. 0.0625 0.25 0.5625 1.
C = - 0.1111 1. dskv = 0.1111
fi(x)= pp = -0.1111 + x

```

## Додаток 5

Знаходження наближених значень функції  $y(x)$  з прикл. 1.8 за допомогою апроксимації Паде (далі КФ - квадратурна формула).

```

// apade
function [YY] =yy(x); YY = log(1+x); endfunction;
function [FI1] =fi1(x); FI1 = x; endfunction;
function [FI2] =fi2(x); FI2 = x*log(1+x);endfunction;
function [FI]=fi(x,alf,bet);FI=alf*x/(1+bet*x);
endfunction;
// Визначення параметрів апроксимації Паде
function [dskv,sf,ab,af,ay,ax] =pade(n,M,a,b,sm,eps);
    h=(b-a)/n; s=0; c1=1; c2=0;

```

```

while (s < sm) s=s+1;
//Обчислення коефіцієнтів СЛАР за КФ середніх
x=a-h/2; a11=0; a12=0; a22=0; b1=0; b2=0;
for i=1:n
x=x+h; y=yy(x); f1=fi1(x); f2=fi2(x);
if(s==1) then r=1; else r= 1/(1-c2*x)**2; end;
a11=a11+f1*f1*r; a12=a12+f1*f2*r;
a22=a22+f2*f2*r; b1=b1+y*f1*r;
b2=b2+y*f2*r;
end;//for
as(1,1)=a11; as(1,2)=a12; as(2,1)=a12;
as(2,2)=a22; bs(1)=b1; bs(2)=b2;
//Розв'язання СЛАР методом Гаусса
C=rref([as bs]); [NN,MM]=size(C); ds=C(:,MM);
if(s == 1) then tr=%f; end;
if(s <> 1) then
d1=abs((ds(1)-c1)/ds(1)); d2=abs((ds(2)-c2)/ds(2));
tr= d1 <= eps & d2 <= eps; c1=ds(1); c2=ds(2); end;
sf=s; if (tr) then break; end;
end;//while
alf= c1; bet=-c2; hm=(b-a)/(M-1); x=a-hm; dskv=0;
for j=1:M x=x+hm; ax(j)=x; yx=yy(x); fx=fi(x,alf,bet);
ay(j)=yx; af(j)=fx; dskv=dskv+(yx-fx)^2; end;//j
dskv=sqrt(dskv/M); // ДСКВ (dskv) - дискретне
// середнє квадратичне відхилення
// Перетворення векторів на матриці-рядки
ax=ax.'; ay=ay.'; af=af.'; ab=[alf bet];
endfunction;
//main program
// Коментарі до фактичних параметрів функції apade
//sf-число виконаних ітерацій, ab-коефіцієнти apade
//[dskv,sf,ab,af,ay,ax] =pade(n,M,a,b,sm,eps);
[dskv,sf,ab,af,ay,ax] =pade(100,7,-0.5,1,20,0.001)
//ay(5)-точне, а af(5)-наближене значення y(1/2)
dlt=abs((ay(5)-af(5))/ay(5))*100
//dlt - Відносна похибка обчислення apade
//Побудова графіків заданої функції та apade
//plot(ax,af,'k*',ax,ay,'k-'); xgrid();
//xtitle(' FI=FI(X) Y=Y(X)', 'X', 'FI Y');
//legend(' FI=FI(X) ', ' Y=Y(X) ', 4, &t);

```

Результати виконання *apade* у табличному (збережено чотири значущих цифри за десятичною крапкою) вигляді:



```

ax = -0.5    -0.25    0.    0.25    0.5    0.75    1.
ay = -0.6931 -0.2877 0.    0.2231 0.4055 0.5596 0.6931
af = -0.6797 -0.2919 0.    0.2276 0.4100 0.5595 0.6842
alfa=1.0230 beta=0.4950 sf = 4.  dskv=0.0067 dlt=1.128

```

## Додаток 6

Знаходження наближених значень функції  $y(x)$  з прикл. 1.9 за допомогою рівномірного наближення.

```

// rivnab
deff('YY =yy (x)', 'YY =x^2');
deff('FI1=fi1(x)', 'FI1=1 ');
deff('FI2=fi2(x)', 'FI2=x ');
deff('FI =fi (x,c1,c2)', 'FI =c1+c2*x');
deff('F110=f110(x)', 'F110=ro(x,s)*fi1(x)^2');
deff('F220=f220(x)', 'F220=ro(x,s)*fi2(x)^2');
deff('F120=f120(x)', 'F120=ro(x,s)*fi1(x)*fi2(x)');
deff('YF10=yf10(x)', 'YF10=ro(x,s)*fi1(x)*yy(x) ');
deff('YF20=yf20(x)', 'YF20=ro(x,s)*fi2(x)*yy(x) ');
function [r] = ro(x,s);
if(s==0) then r=1;
else r=(yy(x)-fi(x,c1,c2))^(s*2);end;
endfunction;//ro
function [sf,cc,afi,ay,ax] = rivnab(n,M,a,b,ni);
s=0; for k=1:ni
as(1,1)=intg(a,b,f110); as(2,2)=intg(a,b,f220);
as(1,2)=intg(a,b,f120); as(2,1)=as(1,2);
bs(1) =intg(a,b,yf10); bs(2) =intg(a,b,yf20);
//Розв'язання СЛАР методом Гаусса
CC0=rref([as bs]); [NNO,MM0]=size(CC0);
x0=CC0(:,MM0); cs=x0; c1=cs(1); c2=cs(2);
s=s+1; sf=s; hm=(b-a)/(M-1); x=a-hm;
for j=1:M x=x+hm; ax(j)=x;
ay(j)=yy(x); afi(j)=fi(x,c1,c2); end;//j
cc=[c1 c2];
end;//k
endfunction;
//main program
// Коментар до фактичних параметрів функції rivnab
//sf-число виконаних ітерацій, cc-коефіцієнти rivnab
//[sf,cc,afi,ay,ax] =rivnab(n ,M,a,b,nit);
[sf,cc,afi,ay,ax] =rivnab(100,5,0,1,2); s=sf;
FI=poly(cc,'x','c')
T=-0.125; H=cc(1);

```

```
//Т-точно, а Н-наближене значення коефіцієнта C1
dlt=abs((Т-Н)/Т)*100
//dlt-Відносна похибка обчислення коефіцієнта C1
sf=s
// sf - число виконаних ітерацій
Результати виконання rivnab (збережено чотири значущих
цифри за десятичною крапкою):
FI = - 0.1190+ x dlt = 4.76% sf = 2.
```

## Додаток 7

Знаходження наближених значень 1-ї похідної функції  $y(x)$  з прикл. 2.2 за симетричними формулами із застосуванням правила Рунге.

```
// numdyf
deff('FF = ff(x)', 'FF = log(1+x)');
deff('F1 = f1(x)', 'F1 = 1/(1+x)');
function [ap1,y1,y,x] = numdyf(a,b,N);
deff('P1 = p1(y1,yr,h)', 'P1=(yr-y1)/(h^2)');
deff('PR = pr(y1,y2,g,p)', 'PR=(y1-y2)/(g^p-1)');
h1=(b-a)/N; h2=h1*2; g=2; p=2;
N1=N+1; NM=N-1; K=N/2; K1=K+1; KM=K-1;
for n=1:N1 xx=a+(n-1)*h1; x(n)=xx;
y(n)=ff(xx); y1(n)=f1(xx); end;
for n=1:N1 apn1(n)=0; apk1(n)=0; ad1(n)=0;
ap1(n)=0; end;
for n=2:N apn1(n)=p1(y(n-1),y(n+1),h1); end;
for k=1:KM apk1(2*k+1)=p1(y(k*2-1),y(k*2+1),h2);end;
for m=3:2:NM ad1(m)=pr(apn1(m),apk1(m),g,p);
ap1(m)=apn1(m)+ad1(m); end;
// Перетворення векторів на матриці-рядки
x = x.'; y = y.'; y1 = y1.'; ap1 = ap1.';
endfunction;
// main program
//y1-точно значення 1-ї похідної, p1-наближене
[p1,y1,y,x] = numdyf(0,0.8,4)
//y1(3)-точно, а p1(3)-наближене значення похідної
dlt=abs((y1(3)-p1(3))/y1(3))*100
//dlt-відносна похибка обчислення похідної при x=0.4
//Побудова графіків розв'язків
//plot(x,p1,'k*',x,y1,'k-'); xgrid();
//xtitle(' P1=P1(X) Y1=Y1(X) ','X','P1 Y1');
//legend(' P1=P1(X) ',' Y1=Y1(X) ',3,&t);
```

Результати виконання *numdyf* у табличному (збережено чотири значущих цифри за десятичною крапкою) вигляді:

x	= 0.	0.2	0.4	0.6	0.8
y	= 0.	0.1823	0.3365	0.4700	0.5878
y1	= 1.	0.8333	0.7143	0.625	0.5556
p1	= 0.	0.	0.7140	0.	0. d1t=0.04%

### Додаток 8

Знаходження значень інтеграла з прикл. 3.1 за допомогою квадратурних формул (КФ) середніх, трапецій (ФТ) та парабол (ФП).

```
// qfpr
//Підінтегральна функція (ПФ)
function [f] =ff(x); f = log(1+x); endfunction;
// Обчислення інтеграла за різними КФ
function [trap,ser,par] = trapserpar(a,b,N);
// Крок сітки
h=(b-a)/N;
// Формування значень ПФ для КФ середніх
for i=1:N xi=a+(i-0.5)*h; y(i)=ff(xi); end;
// КФ середніх
ser=0; for i=1:N ser=ser+y(i); end; ser=ser*h;
// формування значень ПФ для ФТ та ФП
N1=N+1; for i=1:N1 xi=a+(i-1)*h; y(i)=ff(xi); end;
// КФ трапецій
trap=(y(1)+y(N1))/2;
for i=2:N trap=trap+y(i); end; trap=trap*h;
// КФ парабол
K=N/2; s1=y(1)+y(N1); s2=0; s4=0;
for i=2:K s2=s2+y(i*2-1); end;
for i=1:K s4=s4+y(i*2); end;
par=(s1+s2*2+s4*4)*h/3
endfunction;
// Уточнення значень КФ за правилом Рунге
function [Ftrap,Fser,Fpar] = qfpr(a,b,N);
[trap1,ser1,par1] = trapserpar(a,b,N);
[trap2,ser2,par2] = trapserpar(a,b,N*2);
Ftrap=trap2+(trap2-trap1)/3; T=[trap1 trap2];
Fser=ser2+(ser2-ser1)/3; S=[ser1 ser2];
if (N==1) then Fpar=0; P=[0 0]; else
Fpar=par2+(par2-par1)/15; P=[par1 par2]; end;
endfunction;
// main program
// N - парне
```

```
// Точне значення інтеграла F = 0.3863
// Коментар фактичних параметрів функції qfpr
// [Ftrap,Fser,Fpar] = qfpr(a,b,N);
// [Ftrap,Fser,Fpar] = qfpr(0,1,2)
```

Результати виконання *qfpr* (збережено чотири значущих цифри за десятичною крапкою):

```
N=2 Fpar=0.3863 Fser=0.3863 Ftrap=0.3863
```

## Додаток 9

Знаходження наближеного значення кратного інтеграла з прикл. 3.3 за допомогою формули комірок без урахування межових комірок.

```
// mкехmb
// Без урахування межових комірок
function [FF] =ff(t1,t2);
FF=t1*t1+t2*t2; endfunction;
function [FN] =fn(t); FN = 0; endfunction;
function [FV] =fv(t); FV = t*2-1; endfunction;
function [GB] =gb(t1,t2);
GB=(t2 > fn(t1)) & (t2 < fv(t1))
endfunction;
function [Ik]=metkom(nn,n1,a1,b1,n2,a2,b2);
n1m=n1-1; n2m=n2-1; for n=1:n1m
for m=1:n2m k=m+(n-1)*n1m; af(k)=0; end;
end;
if(nn == 1) then xf=b1-(b1-a1)/3; h1=b1-a1;
xs=a2+(b2-a2)/3; h2=b2-a2;
ikom=ff(xf,xs)*h1*h2/2; end;
if(nn <> 1) then h1=(b1-a1)/n1m;
h2=(b2-a2)/n2m; ss=h1*h2;
for n=1:n1 x1(n)=a1+h1*(n-1); end;
for m=1:n2 x2(m)=a2+h2*(m-1); end; t=0;
for n=1:n1m t1=x1(n)+h1/2; x11=x1(n);
x12=x1(n+1);
for m=1:n2m k=m+(n-1)*n1m; t2=x2(m)+h2/2;
x21=x2(m); x22=x2(m+1);
if (gb(x11,x21)) then af(k)=ff(t1,t2);
else af(k)=0; end; //if
t=t+af(k); end; //m
end; //n
end; //if
Ik=ss*t;
endfunction;
```

## Додатки

---

```
// Уточнення значень за правилом Рунге
function [Ikom,Ik2,Ik1] = ...
Ikpr(nn,n1,a1,b1,n2,a2,b2,it);
[Ik1]=metkom(nn,n1,a1,b1,n2,a2,b2);
[Ik2]=metkom(nn,n1*2,a1,b1,n2*2,a2,b2);
Ikom=Ik2+(Ik2-Ik1)/3;
endfunction;
//main program
it=0.21875; //it-точне значення інтеграла
//Ikom-наближене значення інтеграла
[Ikom,Ik2,Ik1]=Ikpr(2,5,0.5,1,9,0,1,it)
// (nn,n1,a1,b1,n2,a2,b2,it)
//Ikom - наближене значення інтеграла
dlt=abs((it-Ikom)/it)*100
//dlt - відносна похибка обчислення інтеграла
Результати виконання mkexmb (збережено чотири значущих
цифри за десятичною крапкою):
Ik1=0.1300 Ik2=0.1926 Ikom 0.2135 dlt=2.40%
```

## Додаток 10

Знаходження наближеного значення кратного інтеграла з прикл. 3.3 за допомогою формули комірок за урахування межових комірок.

```
//mkexmz
// За урахування межових комірок,
//центри яких в області G
function [FF]=ff(t1,t2); FF=t1*t1+t2*t2; ...
endfunction;
function [FN] =fn(t); FN = 0; endfunction;
function [FV] =fv(t); FV = t*2-1; endfunction;
function [GB] =gb(t1,t2);
GB=(t2 > fn(t1)) & (t2 < fv(t1))
endfunction;
function [Ik]=metkom(nn,n1,a1,b1,n2,a2,b2);
n1m=n1-1; n2m=n2-1; for n=1:n1m
for m=1:n2m k=m+(n-1)*n1m; af(k)=0; end; end;
if(nn == 1) then xf=b1-(b1-a1)/3; h1=b1-a1;
xs=a2+(b2-a2)/3; h2=b2-a2;
ikom=ff(xf,xs)*h1*h2/2; end;
if(nn <> 1) then h1 =(b1-a1)/n1m;
h2 =(b2-a2)/n2m; ss=h1*h2;
for n=1:n1 x1(n)=a1+h1*(n-1); end;
```

```

for m=1:n2 x2(m)=a2+h2*(m-1); end; t=0;
for n=1:n1m t1=x1(n)+h1/2;
for m=1:n2m k=m+(n-1)*n1m; t2=x2(m)+h2/2;
x21=x2(m); x22=x2(m+1);
if (gb(t1,t2)) then af(k)=ff(t1,t2);
else af(k)=0; end; //if
t=t+af(k); end; //m
end; //n
end; //if
Ik=ss*t;
endfunction;
// Уточнення значень за правилом Рунге
function [Ikom,Ik2,Ik1] = ...
Ikpr(nn,n1,a1,b1,n2,a2,b2,it);
[Ik1]=metkom(nn,n1,a1,b1,n2,a2,b2);
[Ik2]=metkom(nn,n1*2,a1,b1,n2*2,a2,b2);
Ikom=Ik2+(Ik2-Ik1)/3;
endfunction;
//main program
it=0.21875; //it-точне значення інтеграла
//Ikom-наближене значення інтеграла
[Ikom,Ik2,Ik1]=Ikpr(2,5,0.5,1,9,0,1,it)
//(nn,n1,a1,b1,n2,a2,b2,it)
//Ikom - наближене значення інтеграла
dlt=abs((it-Ikom)/it)*100
//dlt - відносна похибка обчислення інтеграла
Результати виконання mkexmz (збережено чотири значущих
цифри за десятичною крапкою):
Ik1=0.2148 Ik2=0.2146 Ikom=0.2145 dlt=1.94%

```

## Додаток 11

Знаходження наближеного значення кратного інтеграла послідовним інтегруванням за використання вузлів і ваг квадратурної формули Гаусса з прикл. 3.3.

```

//piqfg
function [FF] = ff(t1,t2);FF = t1*t1+t2*t2;
endfunction;
function [FN] = fn(t); FN = 0; endfunction;
function [FV] = fv(t); FV = t*2-1; endfunction;
function [ipi] = pig(nn,n1,a1,b1,n2,it);
ks=[-0.577350 0.577350 0 0 0 ; ...
-0.774597 0.0 0.774597 0 0; ...

```

## Додатки

```
-0.861136 -0.339981 0.339981 0.861136 0; ...
-0.906180 -0.538469 0 0.538469 0.906180];
gm=[ 1 1 0 0 0; 0.55556 0.88889 0.55556 0 0; ...
0.347855 0.652145 0.652145 0.347855 0; ...
0.236927 0.478629 0.568889 0.478629 0.236927];
if(nn == 0) then
  xf=b1-(b1-a1)/3; as=fn(b1); bs=fv(b1);
  xs=as+(bs-as)/3; ipi=ff(xf,xs)*(b1-a1)*(bs-as)/2;
end;
if(nn <> 0) then ipi=0;
  for n=1:n1 x1(n)=((a1+b1)+(b1-a1)*ks(nn,n))/2;
    t=x1(n); a2=fn(t); b2=fv(t);
    for m=1:n2 x2(m)=((a2+b2)+(b2-a2)*ks(nn,m))/2;
  ipi=ipi+ff(t,x2(m))*(b1-a1)*(b2-a2)* ...
    gm(nn,n)*gm(nn,m);
  end;
end; ipi=ipi/4;
end;
endfunction;
//main program
it=0.21875; //точне значення інтеграла
//ipi-наближене значення інтеграла
//dlt-відносна похибка обчислення інтеграла за piqfg
//[ipi] = pig(nn,n1, a1,b1,n2,it);
[ipi] = pig( 1, 2,0.5, 1, 2,it)
//ipi - наближене значення інтеграла
dlt=abs((it-ipi)/it)*100
//dlt - відносна похибка обчислення інтеграла
Результати виконання piqfg (збережено чотири значущих цифри за десятичною крапкою):
ipi= 0.2187499 it = 0.21875 dlt = -0.00003%
```

## Додаток 12

Знаходження наближеного значення кратного інтеграла з прикл. 3.3 за допомогою методу Монте-Карло.

```
// mmk3
//Обчислення інтеграла за (3.20')
deff('PF=pf(x1,x2)', 'PF=x1^2+x2^2');
deff('FV=fv(x1)', 'FV=x1*2-1');
function [IH,np]=mmk3(a,b,K,N,fmax,IT);
K1=K+1; V=1;
for i=1:K V=V*(b(i)-a(i)); end; V=V*fmax;
X=grand(K1,N,"def"); n=0;
for j=1:N for i=1:K ksi(i)=X(i,j);
```

```

x(i)=a(i)+(b(i)-a(i))*ksi(i); end;//i
x1=x(1); x2=x(2); eta=pf(x1,x2)/fmax;
t1=a(1)<=x1&x1<=b(1); t2=a(2)<=x2&x2<=fv(x1);
t3=(X(K1,j)<=eta);
if (t1&t2&t3) then n=n+1; end;
end;//j
np=n; IH=V*np/N;
endfunction;
// main program
IT=0.21875; //точне значення інтеграла
a=[0.5 0]; b=[1 1]; K=2; fmax=2; N=1000;
[IH,np]=mmk3(a,b,K,N,fmax,IT)
// IH - наближене значення інтеграла
// N - число випадкових випробувань,
// np - число попадань під поверхню
// підінтегральної функції в одиничному кубі
// IH - наближене значення інтеграла
dlt=abs((IT-IH)/IT)*100
//dlt - відносна похибка обчислення інтеграла

```

Результати виконання *mmk3* (збережено чотири значущих цифри за десятичною крапкою):

```

N=1000; np=205. IH=0.205 dlt=6.9%
N=10000; np=2170. IH=0.217 dlt=0.8%

```

### Додаток 13

Знаходження наближеного значення кратного інтеграла з прикл. 3.3 за допомогою методу Монте-Карло з використанням реальних значень підінтегральної функції.

```

// mmk2
//Обчислено інтеграл за (3.20), тобто беремо
//значення підінтегральної функції у
//випадкових вузлах області G
deff('PF=pf(x1,x2)','PF=x1^2+x2^2');
deff('FV=fv(x1)','FV=x1*2-1');
function [IH,np]=mmk2(a,b,K,N);
K1=K+1; V=1; for i=1:K V=V*(b(i)-a(i)); end;
n=0; si=0; X=grand(K1,N,"def");
for j=1:N for i=1:K ksi(i)=X(i,j);
x(i)=a(i)+(b(i)-a(i))*ksi(i); end;//i
x1=x(1); x2=x(2);
t1=a(1)<=x1&x1<=b(1); t2=a(2)<=x2&x2<=fv(x1);
if (t1&t2) then n=n+1; si=si+pf(x1,x2); end;//if
end;//j
np=n; IH=V*si/N;
endfunction;

```



## Додатки

---

```
// main program
IT=0.21875; //точне значення інтеграла
a=[0.5 0]; b=[1 1]; K=2; N=1000;
  [IH,np]=mmk2(a,b,K,N)
//IH - наближене значення інтеграла
//N - загальне число випадкових випробувань,
//np - число попадань в область інтегрування G
//IH - наближене значення інтеграла
  dlt=abs((IT-IH)/IT)*100
//dlt - відносна похибка обчислення інтеграла
Результати виконання mmkr (збережено чотири значущих цифри
за десятичною крапкою):
N=1000; np=495. IH=0.2126 dlt=2.83%
N=10000; np=4989. IH=0.2180 dlt=0.33%
```

## Додаток 14

Знаходження найменшого ненульового кореня нелінійного алгебричного рівняння (НАР) з прикл. 4.1 за методом половинного ділення (МПД).

```
// неспрд - Розв'язання НАР  $f(x)=0$  МПД
function [F] = f(t); F= t-sin(t)/cos(t)endfunction;
function [s, x2] = mpd(x0, x1, eps, smax);
// x0, x1 - відрізок відокремлення кореня,
// eps - точність обчислення кореня,
// smax - допустиме число виконуваних ітерацій.
// s - число виконаних ітерацій, x2 - розв'язок
  i=0; f0=f(x0); f1=f(x1);
  while ( i < smax) i=i+1; x2=(x0+x1)/2; f2=f(x2);
    if (f0*f2 < 0) then x1=x2; f1=f2; end;
    if (f1*f2 < 0) then x0=x2; f0=f2; end;
  if ((f0*f2==0)|(f1*f2==0)|(abs(x1-x0) <= eps) then
    x2=(x0+x1)/2; break; end;
  end; //while
  s = i+2;
endfunction;
// main program функції mpd
//[s, x2] = mpd( x0, x1,eps,smax);
  [s, x2] = mpd(4.4, 4.5,0.001,99)
```

Результати виконання *mpd* (збережено чотири значущих цифри за десятичною крапкою):  
x2=4.475 s=3. x2=4.4875 s=4. x2=4.4934 s=9.

## Додаток 15

Знаходження найменшого ненульового кореня нелінійного алгебричного рівняння (НАР) з прикл. 4.1 за методом простих ітерацій.

```
// neqnpri - метод простих ітерацій розв'язання НАР x=fi(x) методом простих ітерацій
function [FI] = fi(t); FI=%pi+atan(t); endfunction;
function [s , xs1] = npri(xs , eps , smax);
// xs - наближення кореня,
// eps - точність обчислення кореня,
// smax - допустиме число виконуваних ітерацій.
// s - число виконаних ітерацій, xs1 - розв'язок
i=0; while (i < smax) i=i+1; xs1=fi(xs);
if (abs(xs1-xs) <= eps) then break; end;
xs=xs1; end; s=i-1; xs1=xs;
endfunction;
// main program функції
//[s , xs1] = npri(xs , eps , smax);
[s , xs1] = npri(4.5,0.0001,9)
```

Результати виконання *neqnpri* (збережено чотири значущих цифри за десятичною крапкою)

```
eps=0.001 xs1=4.49372 s=1.
```

## Додаток 16

Знаходження найменшого ненульового кореня нелінійного алгебричного рівняння (НАР) з прикл. 4.1 за методом дотичних.

```
// dotych - метод дотичних
// Розв'язання НАР f(x)=0 методом дотичних
function [s , xs1] = dotych(xs , eps , smax)
// xs - нульове наближення кореня,
// eps - точність обчислення кореня,
// smax - допустиме число виконуваних ітерацій.
// s - число виконаних ітерацій, xs1 - розв'язок
deff('FF = ff(t)', 'FF = t - sin(t)/cos(t)');
// Похідна від заданої ff (x)
deff('FP = fp(t)', 'FP = 1 - 1/cos(t)^2');
// Ітераційний процес (3,2)
i=0; while (i < smax) i=i+1;
xs1 = xs - ff(xs)/fp(xs);
d = abs(xs1-xs); xs = xs1;
if d <= eps then break; end; end;
// Формування результату виконання dotych
```

```
s = i-1; xs1 = xs
endfunction;
// main program функції dotych
//[s , xs1] = dotych(xs,eps,smax)
[s , xs1] = dotych(4.4,0.01,9)
```

Результати виконання *dotych* (збережено чотири значущих цифри за десятичною крапкою):

eps=0.035 xs1=4.50186 s=1. eps=0.01 xs1=4.4937 s=2.

### Додаток 17

Знаходження найменшого ненульового кореня нелінійного алгебричного рівняння (НАР) з прикл. 4.1 за методом січних.

```
// neqsich - метод січних
function [F] = f(t); F= t - tan(t); endfunction;
function [s , xsp1] = msich(xsm1,xs, eps , smax);
// xsm1,xs - нульові наближення кореня,
// eps - точність обчислення кореня,
// smax - допустиме число виконуваних ітерацій.
// s - число виконаних ітерацій, xs2 - розв'язок
i=0; while (i < smax)
    i=i+1; fsm1=f(xsm1); fs=f(xs);
    xsp1=xs-(xs-xsm1)*fs/(fs-fsm1);
    xsm1=xs; xs=xsp1;
    if (abs(xsm1-xs) <= eps) then break; end;
end;    s=i; xsp1=xs;
endfunction;
// main program функції msich
//[s , xsp1] = msich(xsm1, xs, eps,smax);
[s , xsp1] = msich(4.55,4.50,0.001, 9)
```

Результати виконання *neqsich* (збережено чотири значущих цифри за десятичною крапкою):

eps=0.035 xsp1= 4.4952 s=1. eps=0.003 xsp1=4.4935 s=2

### Додаток 18

Знаходження найменшого ненульового кореня нелінійного алгебричного рівняння (НАР) з прикл. 4.1 за методом парабол.

```
// meparab - метод парабол
function [F] = f(t); F= t - tan(t); endfunction;
function [s,xsp1] = mprb(xsm2,xsm1,xs,eps,smax);
// xsm2,xsm1,xs - нульові наближення кореня,
// eps - точність обчислення кореня,
// smax - допустиме число виконуваних ітерацій.
// s - число виконаних ітерацій, xs2 - розв'язок
```

```

i=0; while (i < smax) i=i+1;
      fsm2=f(xsm2); fsm1=f(xsm1); fs =f(xs);
      pr11=(fsm2-fsm1)/(xsm2-xsm1);
      pr12=(fsm1-fs)/(xsm1-xs);
      pr2 =(pr11-pr12)/(xsm2-xs);
c=fs; a=pr2; b=a*(xs-xsm1)+pr11; ds=b*b-a*c*4;
z1=(-b+sqrt(ds))/a/2; z2=(-b-sqrt(ds))/a/2
      if (abs(z1) >= abs(z2)) then z=z2; end;
      if (abs(z1) < abs(z2)) then z=z1; end;
      xsm2=xsm1; xsm1=xs; xs=xs+z;
      if (abs(xs-xsm1) <= eps) then break; end;
      end; s=i; xsp1= xs;
endfunction;
// main program функції mprb
//[s , xsp1] = mprb(xsm2,xsm1,xs, eps , smax);
[s , xsp1] = mprb(4.51,4.50,4.49,0.001,9)

```

Результати виконання *неграпараб* (збережено чотири значущих цифри за десятичною крапкою):

eps=0.01 xsp1=4.4931 s=1. eps=0.001 xsp1=4.4934 s=2.

#### Додаток 19

Знаходження розв'язку системи нелінійних алгебричних рівнянь з прикл. 4.2 методом простих ітерацій.

```

// sneqmsi - метод простих ітерацій
function [s , x1] = mri(x, eps , smax);
// x - нульове наближення, eps - точність
// визначення розв'язку, smax - допустиме
// число ітерацій.
// s - число виконаних ітерацій, x1 - розв'язок
b1=[-1; 0]; b2=[0; -1]; [p1] = sa(x); p2=p1;
CC=rref([p1 b1]); [NN,MM]=size(CC); lm1 =CC(:,MM);
CC=rref([p2 b2]); [NN,MM]=size(CC); lm2 =CC(:,MM);
lm=[lm1(1) lm1(2); lm2(1) lm2(2)];
s=0; while (s<=smax) then s=s+1;
[fi] = sf(x,lm); fi1=fi(1); fi2=fi(2);
dx1=fi1-x(1); dx2=fi2-x(2); x(1)=fi1; x(2)=fi2;
// Перевірка виконання умови завершення ітерацій
if(sqrt((dx1^2+dx2^2)/2)<=eps) then break; end;
end; x1=x;
endfunction;
function [pp] = sa(x);
x1=x(1); x2=x(2); ch=cosh(x1); sh=sinh(x1);
co=cos(x2); si=sin(x2);
// Обчислення матриці лямбда
pp (1,1) = ch*co+1; ; pp(1,2) = sh*si;
pp (2,1) = -sh*si; pp(2,2) = ch*co+1;

```

```

endfunction;
function [fi] = sf(x,lm);
x1=x(1); x2=x(2);
ch=cosh(x1); sh=sinh(x1);
co=cos(x2); si=sin(x2);
f(1)=sh*co+x1; f(2)=ch*si+x2;
fi(1)=x1+(lm(1,1)*f(1)+lm(1,2)*f(2));
fi(2)=x2+(lm(2,1)*f(1)+lm(2,2)*f(2));
endfunction;
//main program
// Коментар до фактичних параметрів функції sneqmsi
//[s , x1] = mpi(      x,      eps , smax)
[s , x1] = mpi([2.2 4.5], 0.01 , 99)

```

Результати виконання *sneqmsi* (збережено чотири значущих цифри за десятичною крапкою):

```

eps=0.03  x1=2.2055  4.2107  s=1  eps=0.1  x1=2.2477
4.1989 s=2 eps=0.01 x1=2.25476 4.2106  s=3.

```

## Додаток 20

Знаходження розв'язку системи нелінійних алгебричних рівнянь (СНР) з прикл. 4.2 методом Ньютона використання метода Гаусса (МГ).

```

// sneqnew - метод Ньютона
function [s , x1] = newton(x, eps , smax);
// x - нульове наближення, eps - точність
// розв'язку, smax - допустиме число ітерацій.
// s - число ітерацій, x1 - розв'язок (3.6)
s=0; while (s<=smax) then s=s+1;
// Обчислення складових СНР (3.9)
[ff ,dd] = sdf(x);
//Модифікація ПЧ СНР (3.9)
ff(1)=-ff(1); ff(2)=-ff(2);
// Розв'язання СЛАР (3.9) МГ на кожній ітерації
CC=rref([dd ff]); [NN,MM]=size(CC); dx =CC(:,MM);
// Уточнення розв'язку за (3.10)
x(1)=x(1)+dx(1); x(2)=x(2)+dx(2);
//Перевірка виконання умови (3.11)
if(sqrt((dx(1)^2+dx(2)^2)/2)<=eps) then break; end;
end; x1=x;
endfunction;
// Функція sdf формування складових СНР (3.9)
function [ff ,dd] = sdf(x); x1=x(1); x2=x(2);
ch=cosh(x1); sh=sinh(x1); co=cos(x2); si=sin(x2);
// Обчислення елементів матриці СНР (3.9)

```

```

dd (1,1) = ch*co+1; dd(2,1) = sh*si;
dd (1,2) = -sh*si;   dd(2,2) = ch*co+1;
// Обчислення ПЧ СНР (3.9)
ff(1)=sh*co+x1; ff(2)=ch*si+x2;
endfunction;
//main program
// Коментар до фактичних параметрів функції newton
ton
//[s , x1] = newton( x, eps , smax)
[s , x1] = newton([2.2 4.5], 0.3 , 99)

```

Результати виконання *sneqnew* (збережено чотири значущих цифри за десятичною крапкою):

```

eps=0.3 x1=2.20551 4.21065 s=1
eps=0.1 x1=2.25188 4.2122585 s=2
eps=0.01 x1=2.25073 4.21239 s=3.

```

## Додаток 21

Знаходження розв'язку СЛАР методом Гаусса з вибором головного елемента з прикл. 5.1.

```

//metgauss - метод Гаусса
// Вибір головного елемента
function [D]=Eme(D,i,N);
k=k+1; while (D(k,i)==0) k=k+1; end;
for j=1:N p=D(i,j); D(i,j)=D(k,j);
D(k,j)=p; end;
endfunction;
// Прямий хід
function [D]=priam(A,b,N);
D=cat(2,A,b); for i=1:N-1
if (D(i,i)==0) then D=Eme(D,i,N); end;
for j=0:N D(i,N+1-j)=D(i,N+1-j)/D(i,i); end;
for m=i+1:N q=D(m,i); for j=i:N+1
D(m,j)=D(m,j)-q*D(i,j); end; end; end;
D(N,N+1)=D(N,N+1)/D(N,N); D(N,N)=1;
endfunction;
// Зворотний хід
function [Y]=zvorot(A,b);
N=size(A,1); D=priam(A,b,N); P(N)=D(N,N+1);
for j=1:N-1 s=0;
for k=0:j-1 s=s+D(N-j,N-k)*P(N-k); end;
P(N-j)=(D(N-j,N+1)-s)/D(N-j,N-j); end; Y=P.';
endfunction;
//main program
A=[3 2 2; 2 -3 3; 4 1 -6]; b=[12;14;9];

```

## Додатки

---

```
x=zvorot(A,b)
```

```
X=x.'
```

Результати виконання *metgauss*

```
x = 4. -1. 1.
```

## Додаток 22

Знаходження розв'язку тридіагональної СЛАР з прикл. 5.2 методом алгебричної прогонки.

```
// algprog
// Розв'язання СЛАР алгебричною прогонкою
function [x] = algprog(n1,a,b,c,d);
// Зворотний хід
n=n1-1; k(1)=0; h(1)=0; c(n)=0;
a(1)=0; k(n1)=0; x(n1)=0;
for i=1:n ai =a(i); zn =b(i)-ai*k(i);
    k(i+1)=c(i)/zn; h(i+1)=(ai*h(i)-d(i))/zn; end;
// Прямий хід
j=n1;
for i=1:n j = j-1; x(j)=k(j+1)*x(j+1)+h(j+1);
    end;
// Перетворення вектора на матрицю-рядок
x=x.'
endfunction;
//main program
// Коефіцієнти СЛАР (5.4)
a=[0 1 -1]; b=[-2 1 2]; c=[ 1 1 0]; d=[4 2 -8];
n1=4; n=n1-1;
// Виклик функції algprog
[X] = algprog(n1,a,b,c,d);
for i=1:n xp(i)=X(i); end; x=xp.'
```

Результати виконання *algprog* (збережено чотири значущих цифри за десятичною крапкою):

```
x = 1. 2. 3.
```

## Додаток 23

Знаходження розв'язку СЛАР з прикл. 5.3 методом Гаусса — Зейделя.

```
// mgz - метод Гаусса - Зейделя
function [x1,s]=mgz(n,a,b,smax,eps)
// n - порядок СЛАР, a - матриця A, b - вектор ПЧ,
// smax - допустиме число ітерацій, eps - точність.
// s - число виконаних ітерацій, x1 - розв'язок.
// формування початкового наближення розв'язку
```

```

for k=1:n x0(k)=0; end;
// Початок ітераційного процесу (5.7)
s=0;
// Перевірка умови (s <= smax)
while (s <= smax) s=s+1
    for k=1:n km=k-1; kp=k+1; ss=0;
        if (km == 0) then
            for j=kp:n ss=ss+a(k,j)*x0(j); end; end;
// Реалізація (4.6)
        if (km <> 0) then
            for j=1:km ss=ss+a(k,j)*x1(j); end;
            for j=kp:n ss=ss+a(k,j)*x0(j); end;
            end; x1(k)=(b(k)-ss)/a(k,k);
        end;
// Середнє квадратичне відхилення dlt розв'язку
ss=0; for k=1:n ss=ss+(x0(k)-x1(k))^2; end;
dlt=sqrt(ss/n); for k=1:n x0(k)=x1(k); end;
// Перевірка умови збіжності процесу (5.7)
if (dlt <= eps) then break; end; end; x1=x1.';
endfunction;
//main program
// Значення складових СЛАР (5.7)
a=[4. -1. 1.; 2. 6. -1.; 1. 2. -3.]; b=[4. 7. 0.];
// Коментар до фактичних параметрів функції mgz
//[x1,s]=mgz(n,a,b,smax,eps)
[x1,s]=mgz(3,a,b,99,0.5)

```

Результати виконання *mgz* (збережено чотири значущих цифри за десятичною крапкою):

```

eps=0.99 s=1. x1=1.      0.8333 0.8889
eps=0.5 s=2. x1=0.9861 0.9861 0.9861
eps=0.1 s=3. x1=1.      0.9977 0.9985
eps=0.01 s=4. x1=0.9998 0.9998 0.9998
eps=0.001s=5. x1=1.    1.0000 1.0000

```

## Додаток 24

Знаходження власних значень (ВЗ) та відповідних власних векторів (ВВ) квадратної матриці з прикл. 5.8 методом інтерполяції (далі інтерполяційний поліном Ньютона – ППН).

```

// mintrpl - ППН на нерівномірній сітці
function [f] = ff(x); f =det(A-x*E); endfunction;
// Задана функція
function [pln] = plnwt(n,xr,xx,yy);
//функція plnwt обчислення ППН у точці xr
y=yy(n+1);

```



## Додатки

```
for k=1:n n1=n-k+1; y=yу(n1)+у*(xp-xx(n1)); end;
pln=y
endfunction;
// Функція ірп формування значень ІПН у точках xd
function [C,pr,Pn,y,x] = ірп(n,M,xx,yy,xd);
// n - степінь ІПН, M - число точок у масиві xd
// Знаходження поділених різниць
m=n+1;
for k=1:m nt=n-k+1; for i=1:nt n1=n-i+1; n2=n1+1;
    yy(n2)=(yy(n2)-yy(n1))/(xx(n2)-xx(n1-k+1));
    pr(k,i)=yy(n2); end
end;
x1=xx(1); y1=yy(1); x2=xx(2); y12=pr(1,2); C3=pr(2,1);
C(3)=C3; C(2)=y12-(x1+x2)*C3; C(1)=y1+(x2*C3-y12)*x1;
// Обчислення ІПН у точках xd
for k=1:M xp=xd(k); p(k)=plnwt(n,xp,xx,yy);
    x(k)=xp; y(k)=ff(xp); end;
// Перетворення векторів на матриці-рядки
x=x.'; y=y.'; Pn=p.'; pr=pr.';
endfunction;
// main program
// Вихідні дані для побудови ІПН
A=[2 -1; -1 1]; E=eye(2,2); n=2; n1=n+1; M=5;
NA=norm(A,'inf'); a=-NA; b=NA;
for k=1:n1 xx(k)=a+(k-1)*(b-a)/(n1-1);
    yy(k)=ff(xx(k)); end;
// Координати точок для таблиць і графіків
for k=1:M xd(k)=a+(k-1)*(b-a)/(M-1); end;
[C,pr,Pn,y,x] = ірп(n,M,xx,yy,xd)
p=poly(C,'x','c')
lam=roots(p)
for i=1:n x2(i)=-(A(1,1)-lam(i))/A(1,2); end;
x1=[1 x2(1)]
x2=[1 x2(2)]
```

Результати виконання `mintpr1` (збережено чотири значущих цифри за десятичною крапкою):

```
x = - 3.    0.    3.
y =  19.    1.    1.
Pn =  19.    1.    1.
pr = ( поділені різниці)
     0.    1.
    - 6.    0.
```

2

$p = 1 - 3x + x^2$  - Характеристичний поліном  
 $\lambda_1 = 2.6180$  ( $\lambda_1$ )  $0.3820$  ( $\lambda_2$ ) ВЗ  
 $x_1 = 1.$   $-0.6180$  ВВ1 для ( $\lambda_1$ )  
 $x_2 = 1.$   $1.6180$  ВВ2 для ( $\lambda_2$ )

## Додаток 25

Знаходження найбільшого власного значення (ВЗ) квадратної матриці та відповідного йому власного вектора (ВВ) з прикл. 5.8 степеневим методом.

```

// stepmet
// Добуток вектора на матрицю
function [v] = Ay (n,A,y) ;
for i=1:n sm=0; for j=1:n
sm = sm +A(i,j)*y(j); end; v(i)= sm; end;
endfunction;
function [lm,y] = norma (n,y) ;
ymax=abs(y(1)); for i=2:n ayi=abs(y(i));
if(ay1>ymax) then ymax=ayi; end; end;
// Нормування компонент вектора
for i=1:n y(i)=y(i)/ymax; end; lm=ymax;
endfunction;
// Реалізація степеневого методу
function [y,lam,s] = stepmet (n,y,A,eps,smax,lam0) ;
s=0;
// Перевірка умови (s <= smax)
while (s<=smax) then s=s+1;
[v] = Ay (n,A,y) ; for j=1:n y(j)=v(j); end;
[lam2,y] = norma (n,y) ;
// Перевірка умови завершення ітерацій
if(abs((lam0-lam2)/lam2)<=eps) then break; end;
lam0=lam2; end; lam=lam2;
endfunction;
// main program
// Складові довільного вектора x та матриці A
y = [1, 0]; A=[2 -1; -1 1];
// Коментар до фактичних параметрів функції stepmet
// sf-виконане число ітерацій, lambda-ВЗ, x - ВВ
//[x,lambda, s] = stepmet (n,y,A, eps,smax,lam0)
[x,lambda,sf] = stepmet (2,y,A,0.0001, 99, 10)
  
```

Результати виконання *stepmet* (збережено чотири значущих цифри за десятичною крапкою):

sf=7. lambda=2.6180 x= 1. - 0.6180

Знаходження числового розв'язку задачі Коші для звичайного диференціального рівняння (ЗДР) з прикл. 6.3 методом Рунге — Кутти 2-го порядку (МРК2) за  $\alpha = 1/2$ .

```
//zkzdrk2 (alfa=1/2)
// Розв'язання ЗК для ЗДР МРК2 (alfa=1/2)
deff('FF = ff(x,y)', 'FF = -y/(1+2*x)*0.9');
deff('UU= uu(x)', 'UU = 1/(1+2*x)^0.45');
function [u,y,x] = zkzdrk2(y0,N,a,b);
// Крок сітки
h=(b-a)/(N-1); xx=a; y1=y0;
for n=1:N x(n)=xx; y(n)=y1;
f1=ff(xx,y1); f2=ff(xx+h,y1+h*f1);
y1=y1+h*(f1+f2)/2; xx=xx+h;
end;
// Обчислення точного розв'язку
for n=1:N xn=x(n); u(n)=uu(xn); end;
// Перетворення векторів на матриці-рядки
x = x.'; y = y.'; u = u.';
endfunction;
// main program
// Коментар до фактичних параметрів zkzdrk2
//[u,y,x] = zkzdrk2(y0,N,a,b);
[u,y,x] = zkzdrk2(1,6,0,1)
//Побудова графіків розв'язків
//plot(x,y,'k*',x,u,'k-'); xgrid();
//xtitle(' Y=Y(X) U=U(X) ', 'X', 'Y U');
//legend(' Y=Y(X) ', ' U=U(X) ');
```

Результати виконання `zkzdrk2 (alfa=1/2)` у табличному (збережено п'ять значущих цифр за десятичною крапкою) вигляді:

x =	0.	0.2	0.4	0.6	0.8	1.
y =	1.	0.85729	0.76482	0.69842	0.64765	0.60715
u =	1.	0.85949	0.76759	0.70131	0.65052	0.60995

Знаходження числового розв'язку задачі Коші (рівняння коливань математичного маятника) після переходу до системи для звичайного диференціального рівняння з прикл. 6.4 методом Рунге — Кутти 2-го порядку за  $\alpha = 1$ .

```
//szdrmrk2
// Числове розв'язання системи ЗДР МРК2 (alfa=1)
```

```

function [dmrk2,u,y,x] = szdrk2(y10,y20,N,a,b)
// ПЧ рівнянь (6.1) та точний розв'язок u(x)
deff('ff1 = f1(x,y1,y2)', 'ff1 = y2');
deff('ff2 = f2(x,y1,y2)', 'ff2 = -y1');
deff('uu = f(x)', 'uu = 2*cos(x)');
// Крок сітки
h=(b-a)/(N-1); xx=a; hh=h/2;
// Початкові умови
y1=y10; y2=y20;
// Числовий розв'язок системи ЗДР МРК2 (alfa=1)
for n=1:N x(n) =xx; y(n)=y1;
    f11=f1(xx,y1,y2); f21=f2(xx,y1,y2);
    f12=f1(xx+hh,y1+hh*f11,y2+hh*f21);
    f22=f2(xx+hh,y1+hh*f11,y2+hh*f21);
    y1=y1+h*f12; y2=y2+h*f22;
    xx=xx+h
end ;
// Обчислення ЧР та точного розв'язку
for n=1:N xn=x(n); u(n)=f(xn); end;
// Обчислення ДСКВ ЧР від точного розв'язку
dmrk2=0; // ДСКВ (dmrk2) - дискретне
// середнє квадратичне відхилення
for k=1:N dmrk2=dmrk2+(u(k)-y(k))^2; end;
dmrk2=sqrt(dmrk2/N);
// Перетворення векторів на матриці-рядки
x = x.'; y = y.'; u = u.';
endfunction;
// main program
// Коментар до фактичних параметрів szdrk2
//[dmrk2,u,y,x] = szdrk2(y10,y20,N,a,b)
[dmrk2,u,y,x] = szdrk2( 2, 0,5,0,1)
// Побудова графіків розв'язків
//plot(x,y,'k*',x,u,'k-'); xgrid();
//xtitle(' Y=Y(X) U=U(X) ', 'X', 'Y U');
//legend(' Y=Y(X) ', ' U=U(X) ');

```

Результати виконання *szdrmrk2* у табличному (збережено чотири значущих цифри за десятичною крапкою) вигляді:

x = 0.	0.25	0.5	0.75	1.
y = 2.	1.9375	1.7520	1.4550	1.0654
u = 2.	1.9378	1.7552	1.4634	1.0806
dmrk2 =	0.00789			

Знаходження числового розв'язку лінійної крайової задачі для системи звичайних диференціальних рівнянь з прикл. 6.5 балістичним методом із застосуванням методу Рунге — Кутти 4-го порядку (МРК4) у сукупності з методом січних.

```
//bmlbprk4 - Числове розв'язання балістичним методом
//ЛКЗ для ЗДР МРК4 у сукупності з методом січних
function [yfin,ay1,ay,ax] = ff(u0,y10,N,a,b);
deff('F0 = f0(x,y,y1)', 'F0 =y1');
deff('F1 = fi(x,y,y1)', 'F1 =-(1+x*x)*y-1');
h=(b-a)/(N-1); x=a; hh=h/2; y=u0; y1=y10;
for n=1:N ax(n) =x;
    f1= f0(x,y,y1); f11=fi(x,y,y1)
    f2= f0(x+hh,y+hh*f1,y1+hh*f11);
    f12=fi(x+hh,y+hh*f1,y1+hh*f11);
    f3= f0(x+hh,y+hh*f2,y1+hh*f12);
    f13=fi(x+hh,y+hh*f2,y1+hh*f12);
    f4 =f0(x +h,y +h*f3,y1 +h*f13);
    f14=fi(x +h,y +h*f3,y1 +h*f13);
    ay(n)=y; ay1(n)=y1;
    if(n ~= N) then
        y =y +h*(f1 +2.*f2 +2.*f3+f4)/6;
        y1=y1+h*(f11+2.*f12+2.*f13+f14)/6;
    end; x=x+h;
end; yfin=y; ay1=ay1.'; ay=ay.'; ax=ax.';
endfunction;
// main program
u0=0; uf=0; y10=1; y11=2; N=9;a=-1; b=1;
//[yfin,ay1,ay,ax] = ff(u0,y10,N, a,b);
[yf0 ,ay1,ay,ax] = ff(u0,y10,N, a,b);
[yf ,ay1,ay,ax] = ff(u0,y11,N, a,b);
y12 =y10+(y11-y10)*(uf-yf0)/(yf-yf0);
[yf ,ay1,ay,ax] =ff(u0,y12,N, a,b)
// Побудова графіків розв'язків
//plot(ax,ay,'k-',ax,ay1,'k*'); xgrid();
//xlabel(' Y=Y(X) Y1=Y1(X) ','X','Y Y1');
//legend(' Y=Y(X)', ' Y1=Y1(X)');
```

Результати виконання *bmlbprk4* у табличному (збережено чотири значущих цифри за десятичною крапкою) вигляді:

ax =	-1.	-0.5	0.	0.5	1.
ay =	0.	0.6906	0.9314	0.6906	0.
ay1=	1.7362	0.9578	0.0003	-0.9569	-1.7335

## Додаток 29

Знаходження числового розв'язку (ЧР) лінійної крайової задачі (КЗ) для системи звичайних диференціальних рівнянь з прикл. 6.5 різницевим методом (РМ).

```
//rmlbpr
//Обчислення ЧР КЗ лінійного ЗДР 2-го порядку
// зі змінними коефіцієнтами (6.48) РМ
function [y,x] = rmlinbpr(N,a,b,a1,b1,g1,a2,b2,g2);
//N - число підвідрізків; a, b - відрізок [a,b];
// a1,b1,g1,a2,b2,g2 - коефіцієнти крайових умов
// (6.48). y - ЧР, x - вузли сітки
// Коефіцієнти і ПЧ ЗДР та точний розв'язок u(x)
deff('pp = p(x)', 'pp = 0');
deff('qq = q(x)', 'qq = 1+x^2');
deff('ff = f(x)', 'ff = -1');
// Крок сітки
h=(b-a)/N; h1=h/2; h2=h*h;
// Розв'язання (6.48) алгебричною прогонкою (АП)
x(1)=a; k(1)=0; e(1)=0;
// Знаходження коефіцієнтів СЛАР (6.48)
xx=a-h; n1=N+1;
for n=1:N xx=xx+h; x(n+1)=xx+h;
if(n<2) then
    aa=0; bb=b1-a1*h; cc=b1; dd=g1*h
    else
    aa=h1*p(xx); cc=aa+1; aa=1-aa; bb=2-h2*q(xx);
    dd=f(xx)*h2;
    end;
// Прямий хід АП
zn=bb-aa*k(n);
k(n+1)=cc/zn; e(n+1)=(aa*e(n)-dd)/zn; end;
aa=-b2; dd=g2*h; bb=-(b2+a2*h);
y(n1)=(aa*e(n1)-dd)/(bb-aa*k(n1));
//Завершення знаходження ЧР y(n) - зворотний хід
for n=1:N ii=n1-n+1; y(ii-1)=k(ii)*y(ii)+e(ii);
    end;
// Перетворення векторів на матриці-рядки
x = x.'; y = y.';
endfunction;
// main program
// Коментар до фактичних параметрів функції
//[y,x]=rmlinbpr(N,a, b ,a1,b1,g1,a2,b2, g2);
```

```
[y,x]=rmlinbp(8,-1,1, 1, 0, 0, 1, 0, 0)
// Побудова графіків розв'язків
//plot(x,y,'k*'); xgrid();
//xtitle(' Y=Y(X) ', 'X', 'Y ');
//legend(' Y=Y(X) ',4,&t);
```

Результати виконання *rmlbp* у табличному (збережено чотири значущих цифри за десятичною крапкою) вигляді:

```
x=-1. -0.75 -0.5 -0.25 0. 0.25 0.5 0.75 1.
y=0. 0.4004 0.6992 0.8808 0.9415 0.8808 0.6992 0.4004 0.
```

### Додаток 30

Знаходження числового розв'язку нелінійної крайової задачі для системи звичайних диференціальних рівнянь з прикл. 6.6 балістичним методом із застосуванням методу Рунге — Кутти 4-го порядку у сукупності з методом січних.

```
// bmnbp4
// Числове розв'язання балістичним методом
// НКЗ для ЗДР МРК4 у сукупності з методом січних
function [yfin,au,ay,ax] = ff(bl,u0,y10,N,a,b,ad);
function [UX]=ux(x);
    UX=log(abs(cos((x-0.5)*as)/cos(as*0.5)))/ad;
endfunction;
deff('F0 = f0(x,y,y1)', 'F0 =y1');
deff('FI = fi(x,y,y1)', 'FI =-(ad*y1*y1+1)');
h=(b-a)/(N-1); hh=h/2; y=u0; y1=y10; as=sqrt(ad);
if ~bl then for n=1:N
    ax(n) =0; au(n)=0; ay(n)=0; end; end; x=a;
for n=1:N
    f1= f0(x,y,y1); f11=fi(x,y,y1)
    f2= f0(x+hh,y+hh*f1,y1+hh*f11);
    f12=fi(x+hh,y+hh*f1,y1+hh*f11);
    f3= f0(x+hh,y+hh*f2,y1+hh*f12);
    f13=fi(x+hh,y+hh*f2,y1+hh*f12);
    f4 =f0(x+h,y+h*f3,y1+h*f13);
    f14=fi(x+h,y+h*f3,y1+h*f13);
    if bl then
        ax(n) = x; ay(n)=y; au(n)=ux(x); end;
        if(n ~= N) then
            y =y+h*(f1+2.*f2+2.*f3+f4)/6;
            y1=y1+h*(f11+2.*f12+2.*f13+f14)/6;
        end;
        x=x+h; end; yfin=y;
//Перетворення векторів на матриці-рядки
au=au.'; ay=ay.'; ax=ax.';
```

```

endfunction;
function [sf,au,ay,ax]= ...
bmlbp(u0,uf,y10,y11,N,a,b,ad,eps,smax);
s=0; while (s <= smax) then s=s+1;
    //[yfin,au,ay,ax] = ff(b1,u0,y10,N, a,b,ad);
    [yf0 ,au,ay,ax] = ff(%f,u0,y10,N, a,b,ad);
    [yf1 ,au,ay,ax] = ff(%f,u0,y11,N, a,b,ad);
    y12 =y11-(y11-y10)*yf1/(yf1-yf0);
    [yf ,au,ay,ax] =ff(%t,u0,y12,N, a,b,ad);
    y10=y11; y11=y12;
if (abs(uf-yf1)<= eps) then break; end; end; sf=s;
endfunction;
// main program
[sf,au,ay,ax]=bmlbp(0,0,1,0.9,5,0,1,0.49,0.0001,99)
//plot(ax,au,'k-',ax,ay,'k*'); xgrid();
//xtitle(' U=U(X) Y=Y(X) ', 'X', 'U Y');
//legend(' U=U(X) ', ' Y=Y(X) ');

```

Результати виконання *bmlbprk4n* (збережено чотири значущих цифри за десятичною крапкою) у табличному вигляді:

```

ax = 0. 0.125 0.25 0.375 0.5
ay = 0. 0.0565 0.0962 0.1198 0.1276
au = 0. 0.0565 0.0962 0.1198 0.1276 . sf = 4.

```

### Додаток 31

Знаходження числового розв'язку (ЧР) нелінійної крайової задачі для системи звичайних диференціальних рівнянь різницевим методом (РМ) з застосуванням ітераційної схеми (6.51) з прикл. 6.6.

```

// rmlnelp
//Обчислення ЧР КЗ нелінійного ЗДР 2-го порядку РМ
function [s,u,y,ax]= ...
rmlnelp(N,a,b,a1,b1,g1,a2,b2,g2,ad,eps,smax);
function [UX]=ux(x);
UX=log(abs(cos((x-0.5)*as)/cos(as*0.5)))/ad
endfunction;
function [F]=f(ym,yp); F=-(h2+ad*(yp-ym)**2/4);
endfunction;
deff('Y0 = y0(x)', 'Y0 = 0');
as=sqrt(ad); h=(b-a)/N; h2=h*h; N1=N+1;
x=-h; y1(1)=g1; y1(N1)=g2;
for n=1:N1
    x=x+h; ax(n)=x; u(n)=ux(x); y(n)=y0(x);
end;//n

```



```

k(1)=0; e(1)=0; s=-1;
while (s <= smax) s=s+1;
for n=1:N if (n == 1) then
aa=0; bb=b1-a1*h; cc=b1; dd=g1*h; end;
if (n >= 2) then
cc=1; aa=1; bb=2; dd=f(y(n-1),y(n+1));
end;
zn=bb-aa*k(n); k(n+1)=cc/zn;
e(n+1)=(aa*e(n)-dd)/zn;
end;//n
aa=-b2; dd=g2*h; bb=-(b2+a2*h);
y1(N1)=(aa*e(N1)-dd)/(bb-aa*k(N1));
for n=1:N
ii=N1-n+1; y1(ii-1)=k(ii)*y1(ii)+e(ii);
end;//n
ss=0;
for i=2:N ss=ss+(y(i)-y1(i))^2; end;//i
ss=sqrt(ss/(N-1));
if (ss <= eps) then break; end;
for i=2:N y(i)=y1(i); end;
end;
//Перетворення векторів на матриці-рядки
ax = ax.'; y = y.'; u = u.';
endfunction;
// main program
// Коментар до фактичних параметрів функції
//[s,u,y,ax]= ...
//rmnelbp(N,a,b,a1,b1,g1,a2,b2,g2, ad,eps ,smax);
[s,u,y,ax]= ...
rmnelbp(8,0,1, 1, 0, 0, 1, 0, 0,0.49,0.00001,99)
// Побудова графіків розв'язків
//plot(ax,y,'k',ax,u,'k-'); xgrid();
//xtitle(' Y=Y(X) U=U(X) ', 'X', 'Y U');
//legend(' Y=Y(X) ', ' U=U(X) ',2,&t);Результати вико-
нання rmnelbp у табличному (збережено чотири значущих цифр-
ри за десятичною крапкою) вигляді:
ax=0. 0.125 0.25 0.375 0.5
y=0. 0.0564 0.0961 0.1197 0.1275
u=0. 0.0565 0.0962 0.1198 0.1276 s=3.- число ітерацій

```

## Додаток 32

Знаходження власного значення (ВЗ) та відповідної йому власної функції лінійної крайової задачі для системи звичайних диференціальних рівнянь з прикл. 6.9 балістичним методом (БМ) із застосуванням методу Рунге — Кутти 4-го (МРК4) порядку у сукупності з методом січних (МС).

```
//bmvzdr
//Числове розв'язання задачі на ВЗ (6.71)
// БМ у сукупності з МС та МРК4
function [yy1] =fyu(lmd,N,u10,u20,h,x) ;
// Розв'язання задачі Коші МРК4
// Коефіцієнти ЗДР
    deff('pp = p(t)', 'pp = 0') ;
    deff('qq = q(t)', 'qq = 0') ;
    deff('ff1 = f1(t,y1,y2)', 'ff1 = y2') ;
    deff('ff2 = f2(t,y1,y2)', ...
        'ff2=-p(t)*y2-(lmd+q(t))*y1') ;
// Початкові умови ЗК ЗДР
y1=u10; y2=u20; hh=h/2;
// ЧР задачі Коші МРК4
    for n=1:N      t=x(n); yy1(n)=y1;
        f11 =f1(t,y1,y2); f21 =f2(t,y1,y2);
        f12=f1(t+hh,y1+hh*f11,y2+hh*f21);
        f22=f2(t+hh,y1+hh*f11,y2+hh*f21);
        f13=f1(t+hh,y1+hh*f12,y2+hh*f22);
        f23=f2(t+hh,y1+hh*f12,y2+hh*f22);
        f14=f1(t+h,y1+h*f13,y2+h*f23);
        f24=f2(t+h,y1+h*f13,y2+h*f23);
        y1=y1+h*(f11+2.*f12+2.*f13+f14)/6;
        y2=y2+h*(f21+2.*f22+2.*f23+f24)/6
    end;
endfunction;
//функція bmvzdr розв'язання з-чі на ВЗ для ЗДР
function [lam,s,dvz,u,y,x] = ...
    bmvzdr(N,a,b,l,u10,u20,lam1,lamn,smax,eps) ;
// N - число вузлів сітки, a, b - відрізок [a,b]
// l - логічна змінна,u10 - початкове значення
//u1,u20 - початкове значення u2,
//lam1 - перше наближення до
// ВЗ, lamn - друге наближення до ВЗ,
```

## Додатки

---

```
//smax - допустиме число ітерацій,  
//eps - точність. lam - знайдене ВЗ,  
//s- число ітерацій,dvz - ДСКВ ЧР,u - ТР,y - ЧР,  
// x - вузли сітки  
// Функція точного розв'язку задачі  
    deff('uu = ff(t)', 'uu = sin(%pi*t)');  
// Крок сітки та формування вузлів сітки  
    h=(b-a)/(N-1);  
    for n=1:N    x(n)=a+(n-1)*h;    end;  
// Формування даних для методу січних  
    [ay] =fyy(lamn,N,u10,u20,h,x); y10=ay(N);  
    [ay1] =fyy(lam1,N,u10,u20,h,x);  
    y11 =ay1(N); day1=lam1-lamn; dd1=abs(day1) ;  
// Початок ітераційного процесу  
    s=0; while ((s<=smax)|(dd1>eps)) then  
        s=s+1;    dy1=y11-y10;  
        if(dy1==0) then break end;  
        da=dy1/day1;    dd0=dd1;  
        if(da==0) then break end; ad2=lam1-y11/da;  
        lamn=lam1; lam1=ad2;  
        [ay2] =fyy(ad2,N,u10,u20,h,x);  
        y12=ay2(N); y10=y11;    y11=y12;  
        for n=1:N    ay(n) =ay1(n); ay1(n)=ay2(n); end;  
        day1=lam1-lamn; dd1=abs(day1);  
        end;  
        [ay] =fyy(lamn,N,u10,u20,h,x);  
        anm=ay((N-1)/2+1);  
        for n=1:N    ay(n)=ay(n)/anm;    end; lam=lamn;  
        for n=1:N    xn = x(n);    y(n)=ay(n);  
        u(n)=ff(xn);    end;  
// ДСКВ dvz ЧР від точного  
dvz=0; for k=1:N    dvz=dvz+(u(k)-y(k))^2; end;  
dvz=sqrt(dvz/N); // ДСКВ (dvz) - дискретне  
// середнє квадратичне відхилення  
// Перетворення векторів на матриці-рядки  
    x = x.';    y = y.';    u = u.';  
    endfunction;  
// main program  
    lamt=9.8696; //Точне власне значення
```

```

[lamn,sf,dvz,u,y,x] = ...
  bmvzdr( 5,0,1,&t, 0, 1, 9, 10, 999,0.00001)
//bmvzdr( N,a,b, 1, u10,u20,lam1,lamn,smax, eps)
// Коментар до фактичних параметрів функції mlinbr
//lamn - наближене власне значення
  dlt=abs((lamt-lamn)/lamt)*100
//dlt - Абсолютна відносна похибка обчислення ВЗ
// Побудова графіків числового та точного розв'язків
//plot(x,y,'k*',x,u,'k-'); xgrid();
//xtitle(' Y=Y(X)      U=U(X)      ','X','Y      U');
//legend(' Y=Y(X) ',' U=U(X) ');

```

Результати виконання *bmvzdr* у табличному (збережено чотири значущих цифри за десятичною крапкою) вигляді:

```

x = 0.    0.25    0.5    0.75    1.
y = 0.    0.7082    1.    0.7060    2.631D-16
u = 0.    0.7071    1.    0.7071    1.225D-16
dvz=0.0007 sf=9. lamn=9.9197 dlt=0.51%

```

### Додаток 33

Знаходження різницевого розв'язку задачі перенесення з прикл. 7.1 за допомогою явної різницевої схеми (PC).

```

//pereneq
// Числове розв'язання задачі перенесення
deff('F = f(x,t)', 'F = x*t');
deff('MU = mu(x)', 'MU = 1-x');
deff('MU1= mu1(t)', 'MU1 = t+1');
// ЧР рівняння перенесення за явною PC
function [yt]=perenes(N,M,a,T,c);
// N і M - числа під відрізків за осями x і t
// відповідно; a, T - довжини відрізків
// по осях x і t відповідно, c - стала.
// yt - числовий розв'язок задачі
// Кроки за обома аргументами
  h=a/N; tau=T/M; h2=h/2; t2=tau/2;
//Параметр явної PC
  lmb=c*tau/h;
// Сітка по x і заповнення 1-го рядка ПУ
for n=1:N+1 xn=(n-1)*h;x(n)=xn; y(1,n)=mu(xn);end;
//Сітка по T і заповнення 1-го стовпчика сітки КУ

```

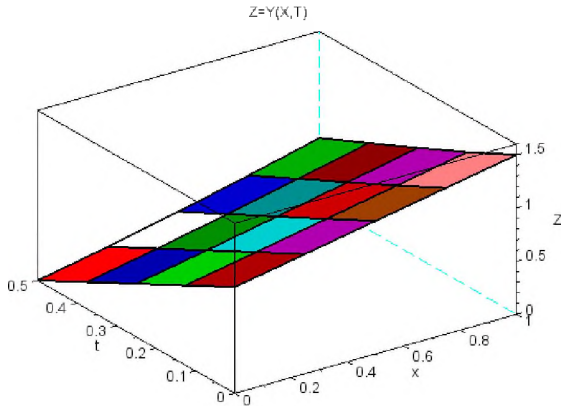
## Додатки

```
for m=1:M+1 tm=(m-1)*tau;t(m)=tm;y(m,1)=mu1(tm); end;
// ЧР рівняння перенесення за явною РС
for m=1:M+1 tm=t(m)+t2; for n=2:N+1 xn=x(n)-h2;
y(m+1,n)=lmb*y(m,n-1)+(1-lmb)*y(m,n)+tau*f(xn,tm);
end; end;
//Формування значень ЧР для друку у вигляді таблиці
for m=2:M+2 for n=2:N+2 yt(m,n)=y(m-1,n-1); end
end;
//Розміщення вузлів по X у 1-му стовпчику таблиці
for n=2:N+2 yt(1,n)=x(n-1); end;
// Розміщення вузлів по X2 у 1-му рядку таблиці
for m=2:M+2 yt(m,1)=t(m-1); end;
yt(1,1)=0;
// Слід замінити це значення
// в таблиці результату yt на "t\х" вручну
endfunction;
// main program
N=4; M=4;
// Коментар до фактичних параметрів функції
//[yt]=perenes(N,M,a, T,c);
[yt]=perenes(N,M,1,0.5,1)
// Формування значень для побудови поверхні ЧР
for m=2:M+2
for n=2:N+2 y(m-1,n-1)=yt(m,n); end; end;
for n=2:N+2 x(n-1)=yt(1,n); end;
for m=2:M+2 t(m-1)=yt(m,1); end;
// Побудова поверхні ЧР
mesh(x,t,y'); title('Z=Y(X,T)');
xlabel('x'); ylabel('t');
```

Результати виконання `pereneg` у табличному (збережено чотири значущих цифри за десятичною крапкою) вигляді:

t\х	0.	0.25	0.5	0.75	1.
0.	1.	0.75	0.5	0.25	0.
0.125	1.125	0.8760	0.6279	0.3799	0.1318
0.25	1.25	1.0034	0.7607	0.5186	0.2764
0.375	1.375	1.1316	0.8967	0.6641	0.4316
0.5	1.5	1.2601	1.0347	0.8146	0.5957

та графічному вигляді:



## Додаток 34

Знаходження функції кручення з прикл. 7.2 за допомогою процесу Лібмана.

```
//funkruch
//Процес Лібмана знаходження функції кручення
function [s, yt]=funkruch(a1,b1,a2,b2,N,M,eps,smax);
// a1,b1 та a2,b2 - початки і кінці відрізків
// області по осях x1 і x2 відповідно,
// N та M - числа підвідрізків по осях x1 і x2
// відповідно, eps - точність, smax - допустиме
// число ітерацій. yt - розв'язок задачі,
// s - число виконаних ітерацій
// Крайові умови задачі
deff('FI = fi(x1,x2)', 'FI = (x1^2-1)*(x2^2-1)/2');
// Кроки по осях x1, x2
h1=(b1-a1)/N; hh=h1^2; h2=(b2-a2)/M;
// Обчислення сталих
a1 =(h1/h2)^2; a11=(1+a1)*2;
// Обчислення координат вузлових точок по осях
for n=1:N+1 x1(n)=a1+(n-1)*h1; end;
for m=1:M+1 x2(m)=a2+(m-1)*h2; end;
// Формування початкових наближень розв'язку по X1
for n=1:N+1 for m=1:M+1 y(n,m)=fi(x1(n),x2(m));end;
end;
// Ітераційний процес (7.21)
for s=1:smax ym=0; sp=s;
for m=2:M for n=2:N w=((y(n-1,m)+y(n+1,m))+...
```

```

        (y(n,m-1)+y(n,m+1))*al+hh*2)/al1;
        d=abs(w-y(n,m)); y(n,m)=w;
// Обчислення модуля максимальної різниці
// між попереднім і наступним наближеннями ЧР
        if (d>y(m)) then y(m)=d; end; end; end;
// Завершення ітерацій (7.21) за точністю
        if (y(m)<=eps) then break; end; end;
// Формування значень ЧР для друку
for m=2:M+2 for n=2:N+2 yt(n,m)=y(n-1,m-1); end
end;
// Розміщення вузлів по X1 у 1-му стовпчику таблиці
for n=2:N+2 yt(n,1)=x1(n-1); end;
// Розміщення вузлів по X2 у 1-му рядку таблиці
for m=2:M+2 yt(1,m)=x2(m-1); end;
yt(1,1)=0;
// Слід замінити це значення
// в таблиці на x2\x1 вручну
endfunction;
// main program
N=8; M=8;
// Коментар до фактичних параметрів funkruch
//[s,yt]=funkruch(a1,b1,a2,b2,N,M, eps, smax);
[s,yt]=funkruch(-1,1,-1,1,N,M,0.0001,100); ytp=yt.'
// Формування значень для побудови поверхні ЧР
for m=2:M+2
for n=2:N+2 y(n-1,m-1)=yt(n,m); end; end;
for n=2:N+2 x1(n-1)=yt(n,1); end;
for m=2:M+2 x2(m-1)=yt(1,m); end;
sp=s;
//mesh(x1,x2,y'); title('Z=Y(X1,X2)');
//xlabel('X1'); ylabel('X2');

```

Результати виконання *funkruch* для 1-го квадранта у табличному вигляді (збережено чотири значущих цифри за десятичною крапкою):

x2\x1	0.	0.25	0.5	0.75	1.	
0.	0.5817	0.5505	0.4528	0.2760	0.	
0.25	0.5505	0.5214	0.4299	0.2632	0.	
0.5	0.4528	0.4298	0.3571	0.2219	0.	
0.75	0.2760	0.2632	0.2219	0.1422	0.	
1.	0.	0.	0.	0.	0.	s = 11.

## Додаток 35

Знаходження числового розв'язку (ЧР) задачі Діріхле для рівняння Лапласа з прикл. 7.3 за допомогою процесу Лібмана.

```
// elpdlrm
// Визначення стаціонарного розподілу температури
function [s, yt]=elpdlrm(a1,b1,a2,b2,N,M,eps,smax);
// a1,b1 та a2,b2 - початки і кінці відрізків
// області по осях x1 і x2 відповідно,
// N та M - числа підвідрізків по осях x1 і x2
// відповідно, eps - точність, smax - допустиме
// число ітерацій. yt - розв'язок задачі,
// sf - число виконаних ітерацій
// Крайові умови задачі
deff('f1 = fi(x1,x2)', 'f1 = (x1*x2)^2');
// Кроки по осях x1, x2
h1=(b1-a1)/N; h2=(b2-a2)/M;
// Обчислення сталих
a1 = (h1/h2)^2; a11=(1+a1)*2;
// Обчислення координат вузлових точок по осях
for n=1:N+1 x1(n)=a1+(n-1)*h1; end;
for m=1:M+1 x2(m)=a2+(m-1)*h2; end;
// Формування початкових наближень розв'язку по X1
for n=1:N+1 for m=1:M+1
    y(n,m)=fi(x1(n),x2(m)); end; end;
// Ітераційний процес (7.21)
for s=1:smax
    ym=0; sp=s;
    for m=2:M for n=2:N
        w=(y(n-1,m)+y(n+1,m))+...
            (y(n,m-1)+y(n,m+1))*a1/a11;
        d=abs(w-y(n,m)); y(n,m)=w;
    // Обчислення модуля максимальної різниці
    // між попереднім і наступним наближеннями ЧР
        if (d>ym) then ym=d; end;
    end;
end;
// Завершення ітерацій (7.21) за точністю
if(ym<=eps) then break; end; end;
// Формування значень ЧР для друку
for m=2:M+2 for n=2:N+2 yt(n,m)=y(n-1,m-1);
end; end;
//Розміщення вузлів по X1 у 1-му стовпчику таблиці
for n=2:N+2 yt(n,1)=x1(n-1); end;
```



```
// Розміщення вузлів по X2 у 1-му рядку таблиці
for m=2:M+2      yt(1,m)=x2(m-1); end;
yt(1,1)=0;
// Слід замінити це значення в таблиці
// на x1\X2 вручну
endfunction;
// main program
N=4; M=4;
// Коментар до фактичних параметрів elpdirm
//[s,yt]=elpdirm(a1,b1,a2,b2,N,M, eps,smax);
[sf,yt]=elpdirm(0,1,0,1,N,M,0.0001,100); ytp=yt.'
// Формування значень для побудови поверхні ЧР
for m=2:M+2
    for n=2:N+2
        y(n-1,m-1)=yt(n,m); end; end;
for n=2:N+2      x1(n-1)=yt(n,1); end;
for m=2:M+2      x2(m-1)=yt(1,m); end;
//mesh(x1,x2,y'); title('Z=Y(X1,X2)');
//xlabel('X1');
//ylabel('X2');
s=sf
```

Результати виконання *elpdirm* у табличному вигляді (збережено чотири значущих цифри за десятичною крапкою):

```
x2\X1  0.  0.25  0.5  0.75  1.
0.  0.  0.  0.  0.  0.
0.25  0.  0.0323  0.0647  0.0859  0.0625
0.5  0.  0.0647  0.1406  0.2165  0.25
0.75  0.  0.0859  0.2165  0.3895  0.5625
1.  0.  0.0625  0.25  0.5625  1.  s=11
```

### Додаток 36

Знаходження числового розв'язку (ЧР) задачі теплопровідності з прикл. 7.4 за допомогою явної різницевої схеми (РС).

```
// Явна РС Розв'язання параболічного рівняння (ПР)
// Права частина рівняння
function u=f(x,t); u=sin(x*t); endfunction
// Початкова умова
function u=mu(x); u=cos(%pi*x/2); endfunction
// Ліва крайова умова
function u=mul(t); u=t+1; endfunction
// Права крайова умова
```

```

function u=mu2(t); u=t; endfunction
// ЧР ПР за допомогою явної РС
function [yt]=parabeq(N,M,a,T,k)
// N і M - числа підвідділків по осях x і t
//відповідно, // a, T - довжини відрізків
//по осях x і t відповідно, k - стала ПР.
// yt - числовий розв'язок задачі
// Кроки за обома аргументами
h=a/N; tau=T/M;
// Сітка по X, заповнення першого стовпчика ПУ
for n=1:N+1 x(n)=(n-1)*h; y(n,1)=mu(x(n));
end;
//Заповнення першого і останнього рядків
// сітки по T,
for m=1:M+1 t(m)=(m-1)*tau;
y(1,m)=mu1(t(m)); y(N+1,m)=mu2(t(m)); end;
// Параметр явної РС
gam=h^2/(k*tau);
// ЧР ПР за допомогою явної РС (7.33)
for m=1:M
for n=2:N
y(n,m+1)=(y(n-1,m)-(2-gam)*y(n,m)+...
y(n+1,m))/gam+tau*f(x(n),t(m));end;
end;
// формування ЧР для друку у вигляді таблиці
for m=2:M+2 for n=2:N+2 yt(n,m)=y(n-1,m-1)
end
end;
// Розміщення вузлів по X у 1-му стовпчику таблиці
for n=2:N+2 yt(n,1)=x(n-1); end;
// Розміщення вузлів по X2 у 1-му рядку таблиці
for m=2:M+2 yt(1,m)=t(m-1); end;
yt(1,1)=0
// Замінити це значення в таблиці на t\х вручну
endfunction;
//
// main program
N=5;M=8;
// Коментар до фактичних параметрів функції
//[yt]=parabeq(N,M,a, T,k)
[yt]=parabeq(N,M,1,0.02,1); ytp=yt.'
// формування значень для побудови поверхні ЧР
for m=2:M+2

```

## Додатки

```
    for n=2:N+2    y(n-1,m-1)=yt(n,m) ; end;
end;
for n=2:N+2    x(n-1)=yt(n,1) ; end;
for m=2:M+2    t(m-1)=yt(1,m) ; end;
// Побудова поверхні ЧР
//mesh (x,t,y'); title('Z=Y(X,T)');
//xlabel('x'); ylabel('t');
```

Результати виконання *parabeq* у табличному вигляді (збережено чотири значущі цифри за десятичною крапкою):

t\х	0.	0.2	0.4	0.6	0.8	1.0
0.	1.	0.95106	0.80902	0.58779	0.30902	0.0
0.01	1.01	0.92778	0.78922	0.57340	0.30146	0.01
0.02	1.02	0.91372	0.76995	0.55943	0.29666	0.02

## Додаток 37

Знаходження числового розв'язку (ЧР) рівняння малих коливань струни з прикл. 7.5 за допомогою різницевої схеми «хрест».

```
// giperbeq
// Права частина рівняння
function u=f(x,t); u=t/(1+x); endfunction;
// Перша початкова умова (ПУ)
function u=mu1(x); u=sin(%pi*x/2); endfunction;
// Друга початкова умова
function u=mu2(x); u=x*x; endfunction ;
// Ліва крайова умова
function u=mu3(t); u=t*t; endfunction ;
// Права крайова умова
function u=mu4(t); u=sqrt(2)/2+t; endfunction;
// Обчислення різницевого розв'язку гіперболічного
//рівняння (ГР) за допомогою різницевої схеми РС
// «хрест»
function [yt]=giperbeq(N,M,a,T,c);
// N і M - число підвідрізків по осях x і t
відповідно,
// a, T - довжини відрізків по осях X і T відповідно,
// c - стала ГР. yt - розв'язок задачі
// Кроки за обома аргументами
h=a/N; tau=T/M;
// Сітка по X, заповнення 1-го й 2-го стовпчиків ПУ
for n=1:N+1    x(n)=(n-1)*h;
    y(n,1)=mu1(x(n));
    y(n,2)=y(n,1)+tau*mu2(x(n)); end;
```

```

// Сітка по T, заповнення першого і останнього рядків
// крайових умов
for m=1:M+1 t(m)=(m-1)*tau;
y(1,m)=mu3(t(m)); y(N+1,m)=mu4(t(m)); end;
// Параметр явної РС «хрест»
gam=(c*tau/h)^2;
// ЧР ГР за допомогою явної РС (6.18) - (6.21)
for m=2:M for n=2:N
y(n,m+1)=-y(n,m-1)+gam*y(n-1,m)+2*(1-gam)*...
y(n,m)+gam*y(n+1,m)+tau^2*f(x(n),t(m));
end;
end;
// Формування значень ЧР для друку у вигляді таблиці
for m=2:M+2 for n=2:N+2 yt(n,m)=y(n-1,m-1); end; end;
// Розміщення вузлів по X у першому стовпчику таблиці
for n=2:N+2 yt(n,1)=x(n-1); end;
// Розміщення вузлів по X2 у першому рядку таблиці
for m=2:M+2 yt(1,m)=t(m-1); end;
yt(1,1)=0;
// замінити це значення в таблиці на t\х вручну
endfunction;
//main program
N=5; M=3;
// Коментар до фактичних параметрів функції parabeq
// [yt]=giperbeq(N,M,a, T ,c);
[yt]=giperbeq(N,M,0.5,0.3,1); ytp=yt.'
// Формування значень ЧР для побудови поверхні ЧР
for m=2:M+2 for n=2:N+2 y(n-1,m-1)=yt(n,m);
end; end;
for n=2:N+2 x(n-1)=yt(n,1); end;
for m=2:M+2 t(m-1)=yt(1,m); end;
// Побудова поверхні ЧР
//mesh(x,t,y'); title('Z=Y(X,T)');
//xlabel('X'); ylabel('T');

```

Результати виконання *giperbeq* у табличному вигляді (збережено п'ять значущих цифр за десятичною крапкою):

t\х	0.	0.1	0.2	0.3	0.4	0.5
0.	0.	0.15643	0.30902	0.45399	0.58779	0.70711
0.1	0.01	0.15743	0.31302	0.46299	0.60379	0.80711
0.2	0.04	0.16749	0.31224	0.46358	0.68303	0.90711
0.3	0.09	0.19663	0.31972	0.53382	0.76833	1.00711

Знаходження числового розв'язку (ЧР) нелінійного інтегрального рівняння (НІР) з прикл. 8.1 за використання квадратурної формули (КФ) з двома вузлами.

```
// nirgam
// Розв'язання НІР різницевим методом
deff('FU = fu(x)' , 'FU = 4*x/lam');
deff('FY = fy(x,t1,t2,f1,f2)', ...
'FY = f1+(x-t1)*(f2-f1)/(t2-t1)');
function [y1,y2] =CAP(k,x1,x2);
a11=x1^2; a22=x2^2; a12=x1*x2;
y1=a12/(a22^2+a11*a12)/k;
y2=sqrt(y1*(1/k-a11*y1)/a12);
endfunction;
// main program
lam=1; a=0; b=1; N=5; hx=(b-a)/(N-1); k=(b-a)*lam/2;
s=1/sqrt(3); for n=1:N ax(n)=a+(n-1)*hx; end;
x1s=0.25; x2s=0.75; //Вузли КФ середніх
[y1s,y2s] =CAP(k,x1s,x2s); xys=[x1s y1s x2s y2s]
x1g=(1-s)/2; x2g=(1+s)/2; // Вузли КФ Гаусса
[y1g,y2g] =CAP(k,x1g,x2g);
xyg=[x1g y1g x2g y2g]
for n=1:N xn=a+(n-1)*hx; x(n)=xn; au(n)=fu(xn);
ays(n)=fy(xn,x1s,x2s,y1s,y2s);
ayg(n)=fy(xn,x1g,x2g,y1g,y2g); end;
xp=ax.'
yp=au.'
ygp=ayg.'
ysp=ays.'
// Побудова графіків заданої функції та ЧР
//plot(x,au,'k-',x,ayg,'k+',x,ays,'k*'); xgrid();
//xtitle(' U=U(X) Yg=Yg(X) Ys=Ys(X)', 'X', 'U Yg Ys');
//legend(' U=U(X)', ' Yg=Yg(X)', ' Ys=Ys(X)', 4, &t);
Результати виконання nirgam у табличному вигляді (збережено
п'ять значущих цифр за десятичною крапкою):


|     | x1s      | y1s     | x2s     | y2s     | КФ середніх (ФС) |
|-----|----------|---------|---------|---------|------------------|
| xys | 0.25     | 1.14286 | 0.75000 | 3.42857 |                  |
|     | x1g      | y1g     | x2g     | y2g     |                  |
| xyg | 0.21133  | 0.84530 | 0.78868 | 3.15470 | КФ Гаусса        |
| xp  | 0.025    | 0.50    | 0.75    | 1.0     | Вузли на [0, 1]  |
| yp  | 0.1      | 2.      | 3.      | 4.      | Точний розв'язок |
| ygp | 0.1      | 2.      | 3.      | 4.      | За КФ Гаусса     |
| ysp | 0.114286 | 2.28571 | 3.42857 | 4.57143 | За КФ ФС         |


```

## Додаток 39

Знаходження різницеvim методом (PM) числового розв'язку (ЧР) інтегрального рівняння Фредгольма 2-го роду (ІРФ2) з прикл. 8.3 за використання квадратурної формули середніх.

```
//ief2
// Розв'язання ІРФ2 РМ
function [dirf,u,y,x] = ief2(N,M,a,b,lam)
//Визначення ядра, ПЧ ІРФ та точного розв'язку (ТР)
deff('KK = K(x,ksi)', 'KK = x*exp(x*ksi)'); //Ядро
deff('FF=f(x)', 'FF=exp(x)'); // Права частина (ПЧ)
deff('FU = fu(x)', 'FU = 1'); // ТР
// Крок сітки
h=(b-a)/N;
// формування ваг та вузлів КФ
for n=1:N c(n)=h; ax(n)=a+(n-0.5)*h; end;
//формування значень ПЧ ІР r та матриці d СЛАР (8.7)
for n=1:N xn=ax(n); r(n)=f(xn);
    for m=1:N d(n,m)=-lam*c(m)*K(xn,ax(m)); end;
    d(n,n)=d(n,n)+1; end;
//Розв'язання (8.7) методом Гаусса із пакета Scilab
C=rref([d r]); [NN,MM]=size(C); R=C(:,MM);
// Обчислення ЧР ІР в M точках відрізка інтегрування
xx=a; hh=(b-a)/(M-1);
for m=1:M yy=f(xx);
    for n=1:N yy=yy+lam*c(n)*K(xx,ax(n))*R(n); end;
// Обчислення значень u(x) та ЧР
u(m)=fu(xx); y(m)=yy; x(m)=xx; xx=xx+hh; end;
// Обчислення ДСКВ dirf ЧР від u(x)
dirf=0; for k=1:M dirf=dirf+(u(k)-y(k))^2; end;
dirf=sqrt(dirf/M); // ДСКВ (dirf) - дискретне
// середнє квадратичне відхилення
// Перетворення векторів на матриці-рядки
x = x.'; u = u.'; y = y.';
endfunction;
// main program
// Коментар до фактичних параметрів функції ief2
//[dirf,u,y,x] = ief2(N ,M,a,b,lam)
[dirf,u,y,x] = ief2(9,5,0,1, -1)
// Побудова графіків заданої функції та ЧР
//plot(x,y,'k*',x,u,'k-'); xgrid();
//xtitle(' Y=Y(X) U=U(X) ', 'X', 'Y U');
//legend(' Y=Y(X) ', ' U=U(X) ',2,&t);
```

Результати виконання `ief2` у табличному вигляді (збережено п'ять значущих цифр за десятичною крапкою):

```
x = 0. 0.25 0.5 0.75 1.
y = 1. 0.99998 1.00000 1.00017 1.00063
u = 1. 1. 1. 1. 1.
dirf = 0.00029%
```

#### Додаток 40

Знаходження різницевим методом (PM) числового розв'язку (ЧР) інтегрального рівняння Вольтерра 2-го роду (IPB2) за використання квадратурної формули трапецій (ФТ) з прикл. 8.2.

```
// iev2t
// Розв'язання IPB2 роду PM i ФТ
deff('KK = K(x,ksi)', 'KK = 1');
deff('FF = f(x)', 'FF = 1');
deff('UU = uu(x)', 'UU = exp(x)');
function [dirv,y,au,ax] = iev2t(a,b,n,lm);
n1=n-1; h=(b-a)/n1; for i=1:n xi=a+(i-1)*h; ax(i)=xi;
af(i)=f(xi); au(i)=uu(xi); end; y(1)=af(1);
for i=2:n x=ax(i); fi=af(i); i1=i-1;
for j=1:i1 ksi=a+(j-1)*h;
if(j==1) then c=0.5; else c=1; end;
fi=fi+h*c*lm*K(x,ksi)*y(j); end; //j
y(i)=fi/(1-h*lm*K(x,x)/2); end; //i
// Обчислення ДСКВ dirv ЧР від u(x)
dirv=0; for i=1:n dirv=dirv+(au(i)-y(i))^2; end;
dirv=sqrt(dirv/n); // ДСКВ (dirv) - дискретне
// середнє квадратичне відхилення
endfunction; //iev2t
//main program
[dirv,y,u,x] = iev2t(0,1,5, 1);
// Побудова графіків заданої функції та ЧР
//plot(x,u,'k-',x,y,'k*'); xgrid();
//xtitle(' U=U(X) Y=Y(X) ', 'X', 'U Y');
//legend(' U=U(X)', ' Y=Y(X) ', 2, &t);
x=x.'
u=u.'
y=y.'
dirvp=dirv
```

Результати виконання `iev2t` у табличному вигляді (збережено п'ять значущих цифр за десятичною крапкою):

```
x=0. 0.25 0.5 0.75 1.
u=1. 1.28403 1.64872 2.117 2.71828
y=1. 1.28571 1.65306 2.12536 2.73261 dirvp=0.0077
```

## Основні елементи мови пакета *Scilab*

Пакет *Scilab* – діалогова система, яка дозволяє вводити команди й відстежувати результати їх виконання. Після пуску пакета *Scilab* в OS Windows за допомогою *scilab-5.5.2.exe* на екрані дисплея з'явиться наведене нижче командне вікно (консоль):

**Команда запуску:  
завантаження початкового середовища**  
->

Символ `-->` є запрошення до введення команд. Після цього символу вводять команду або програму. Їх введення завершується натисненням на клавішу `<Enter>`.

Прикладом найпростішої програми мовою *Scilab* є програма *gauss*, яка реалізує метод Гаусса під час розв'язання СЛАР:

$$\begin{pmatrix} 1 & 2 & -3 \\ 4 & -1 & 1 \\ 2 & 6 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \\ 7 \end{pmatrix}.$$

```
// gauss0
A=[1 2 -3; 4 -1 1; 2 6 -1]; b=[ 0; 4; 7];
C=rref([A b]); [NN,MM]=size(C); x=C(:,MM); x=x.'
```

Результат виконання програми *gauss*:  $x = 1.1.1$ .

Перший рядок – коментар, другий – формування матриці  $A$  та вектора  $b$  СЛАР. Перші три команди 3-го рядка саме й реалізують метод Гаусса розв'язання СЛАР, а 4-й – транспонує матрицю стовпчик (вектор) на матрицю-рядок для його друку у вигляді рядка.

Якщо команда завершується крапкою з комою (;), то результат роботи команди не виводиться на консоль, в іншому ж випадку результат виводиться. Довгі командні рядки можна розбивати на два, додаючи в місці розбиття три крапки (...).

Математичні вирази, як і знаки дій, записують подібно до інших мов програмування.

Як і в інших мовах програмування, об'єктами *Scilab* є скалярні величини, вектори (одновимірні масиви) та матриці (багатовимірні масиви).



## Додатки

---

Визначені в *Scilab* стандартні скалярні величини починаються з % (%*pi* - число  $\pi$ , %*e* – число  $e=2.71828 \dots$ ).

Порядок формування матриць і векторів у *Scilab* показано на прикладі програми *gauss*. Індeksi елементів векторів та матриць починаються з одиниці. Імена змінних завжди починаються з букв.

У пакеті *Scilab* є вбудовані функції (*sin(x)*, *cos(x)*, ...). а в тім їх можуть будувати і користувачі двома способами:

```
deff('f = ff(x)', 'f = log(1+x)'); y=ff(2)
або
function [f] =ff(x);
f = log(1+x)
endfunction; y=ff(2)
```

Результат виконання обох програм однаковий:  $y= 1,098612$ .

У пакеті *Scilab* прийнята загальновідома конструкція циклу. Цикл, як і умовний оператор, завершується командою *end*, наприклад:

```
for n=n1:step:n2
// n - змінна циклу, n1 і n2 початкове і кінцеве значення
змінної циклу n,
// step - крок змінювання i (якщо крок циклу step=1, його
можна опускати)
a=n*n; // тіло циклу
end
Приклад циклу while
x=0;
while x<10
x=x^2+1
end
```

Виконання циклів *while* і *for* можна перервати командою *break*, наприклад:

```
for i=1:n
if (xj)>=xx(i) &xj<xx(i+1)) then
k=i;
break;
end
end;
```

## Список рекомендованої літератури

---

- Алексеев, Е.Р.* Scilab: Решение инженерных и математических задач [Текст] пособие / Е.Р. Алексеев, О.В. Чеснокова, Е.А. Рудченко. — М.: ALTLinux. Бинум. Лаборатория знаний, 2008. — 260 с.
- Бабенко, К.И.* Основы численного анализа [Текст] / К.И. Бабенко. — М.: Наука, 1986. — 744 с.
- Балашова, С.Д.* Чисельні методи [Текст]: у 2 ч. / С.Д. Балашова. — К.: НМК ВО, 1992. — Ч.1. — 280 с.; Ч.2. — 328 с.
- Бахвалов, Н.С.* Численные методы [Текст] / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. — М.: Наука, 1987. — 600 с.
- Боглаев, Ю.П.* Вычислительная математика и программирование [Текст] / Ю.П. Боглаев. — М.: Высш. шк., 1990. — 544 с.
- Бородич, Л.И.* Справочное пособие по приближенным методам решения задач высшей математики [Текст] / Л.И. Бородич [та ін.]. — Минск: Вышэйш. шк., 1986. — 189 с.
- Верлань, А.Ф.* Интегральные уравнения: методы, алгоритмы, программы [Текст]: справочное пособие / А.Ф. Верлань, В.С. Сизиков. — К.: Наук. думка, 1986. — 544 с.
- Годунов, С.К.* Разностные схемы [Текст] / С.К. Годунов, В.С. Рябенский. — М.: Наука, 1977. — 439 с.
- Григоренко, Я.М.* Розв'язання лінійних і нелінійних задач теорії оболонки на ЕОМ [Текст] / Я.М. Григоренко, А.П. Мукоєд. — К.: Либідь, 1992. — 152 с.
- Григоренко, Я.М.* Обчислювальні методи в задачах прикладної математики [Текст] / Я.М. Григоренко, Н.Д. Панкратова. — К.: Либідь, 1995. — 277 с.
- Джордж, А.* Численное решение больших разреженных систем уравнений [Текст] / А. Джордж, Д. Лю. — М.: Мир, 1984. — 333 с.
- Завьялов, Ю.С.* Методы сплайн-функций [Текст] / Ю.С. Завьялов, Б.И. Квасов, В.Л. Мирошниченко. — М.: Наука, 1980. — 352 с.
- Иванов, В.В.* Методы вычислений на ЭВМ [Текст] / В.В. Иванов. — К.: Наук. думка, 1986. — 584 с.
- Калиткин, Н.Н.* Численные методы [Текст] / Н.Н. Калиткин. — М.: Наука, 1978. — 512 с.
- Коллатц, Л.* Численные методы решения дифференциальных уравнений [Текст] / Л. Коллатц. — М.: Иностран. лит., 1953. — 459 с.
- Лященко, М.Я.* Чисельні методи [Текст] / М.Я. Лященко, М.С. Головань. — К.: Либідь, 1996. — 288 с.

## Список рекомендованої літератури

---

На, Ц. Вычислительные методы решения прикладных граничных задач [Текст] / Ц. На. — М.: Мир. 1982. — 296 с.

Ортега, Дж. Введение в численные методы решения дифференциальных уравнений [Текст] / Дж. Ортега, У. Пул — М.: Наука. 1986. — 288 с.

Поттер П. Вычислительные методы в физике [Текст] / П. Поттер. — М.: Мир. 1975. — 392 с.

Самарский, А.А. Теория разностных схем [Текст] / А.А. Самарский. — М.: Наука, 1989. — 616 с.

Самарский, А.А. Численные методы [Текст] / А.А. Самарский, А.В. Гулин. — М.: Наука. 1989.— 432 с.

Современные численные методы решения обыкновенных дифференциальных уравнений [Текст] / под ред. Дж.Холла, Дж.Уатта. — М.: Мир. 1979. — 312 с.

Стечкин, С.Б. Сплайны в вычислительной математике [Текст] / С.Б. Стечкин, Ю.Н. Субботин.—М.:Наука.1976.—248с.

Тихонов, А.Н.Методы решения некорректных задач [Текст] / А.Н. Тихонов, В.Арсенин. — М.: Наука. 1986. — 288 с.

Турчак, Л.И. Основы численных методов [Текст] / Л.И. Турчак. — М.: Наука. 1987. — 318 с.

Фельдман, Л.П. Чисельні методи в інформатиці [Текст] / Л.П. Фельдман. — К.: Вид. група ВНУ, 2006. — 480 с.

Форсайт, Дж. Машинные методы математических вычислений [Текст] / Дж. Форсайт [та ін.]. — М.: Мир. 1980. — 279 с.

Хайпер, Э. Решение обыкновенных дифференциальных уравнений [Текст] / Э. Хайпер, С. Нерсетт, Г. Ваннер. — М.: Мир. 1990. — 512 с.

Чабан, В.Й. Чисельні методи [Текст] / В.Й. Чабан. — Львів: Нац. ун-т «Львів. політехніка», 2001. — 186 с.

Шуп Т. Прикладные численные методы в физике и технике [Текст] / Т. Шуп. — М.: Высш. шк., 1990. — 255 с.

Baudin, M. Введение в Scilab [Текст] / M. Baudin; пер. с англ.: A. Glebov. 2010. — 88 с. [Электронный ресурс]. — Режим доступа: <https://forge.scilab.org/index.php/p/docintrotoscilab/./get/introscilab-v1.4-ru.pdf>. — Заглавие с экрана.

Baudin, M. Программирование в Scilab [Текст] / M. Baudin. 2011. — 159 с. [Электронный ресурс]. — Режим доступа:

[http://forge.scilab.org/upload/doc/progscilab/files/progscilab\\_v0.9.1.1.dif](http://forge.scilab.org/upload/doc/progscilab/files/progscilab_v0.9.1.1.dif). — *Exec-file scilab-5.5.2.exe*. [Electronic resource]. — Access mode: <http://www.scilab.org>. Title from the screen.

[Scilab beginners.pdf](#). [Electronic resource]. — Access mode: <http://www.scilab.org>. — Title from the screen.

## Предметний покажчик

---

- Адамса метод 125  
Апроксимація 178  
— абсолютна 165  
— безумовна 165  
— дробово-лінійна (Паде) 31  
— лінійна 25  
— нелінійна 31  
— умовна 165  
Бубнова—Гальоркіна метод 147, 154, 173, 198  
Вбудована функція 4, 54  
Вибір квадратурних формул 58  
— кубатурних формул 72  
Вирівнювання 17  
Вилучення коренів 76  
Випадкова рівномірно розподілена величина 66  
Відокремлення коренів 75  
Власні вектори 94, 103, 111, 152  
— значення 94, 103, 111, 114  
— функції 152, 157  
Гарвіка прийом 81  
Гаусса—Зейделя метод 100  
Діріхле задача 167  
Екстраполяція 15  
Залишковий член (похибка) інтеполяції 13  
Згладжування 47  
Згущення сітки 128  
Інтегралі невластні 60  
— з необмеженими межами 56  
— з особливими точками 61  
Інтегрування за допомогою сплайнів 57  
— розривних та знакозмінних функцій 58  
Інтегральні рівняння 189  
— Вольтерра 190, 192, 195  
Інтегральні рівняння нелінійні 191  
— — Фредгольма 190, 194  
Інтерполяційний поліном 9  
— — Ерміта 16  
— — Лагранжа 9  
— — Ньютона 11  
Інтерполяція 8  
— застосування 16  
— збіжність 16  
— квазілінійна 17  
— лагранжова 8  
— лінійна 8  
— нелінійна 16  
— обернена 16  
— функції двох змінних 22  
Квазілінійна залежність 18, 31  
Квазірівномірні сітки 45  
Квадратурні формули 46  
— — ваги 49  
— — вузли 49  
— — Гаусса 54  
— — залишковий член 49, 52  
— — збіжність 58  
— — нелінійні 59  
— — нестандартні 57  
— — парабол 51  
— — похибка 50  
— — середніх 49  
— — Сімпсона 51  
— — трапецій 50

- Коректність постановки задачі 6  
Коші задача 116  
Крайова задача 132  
Кубічний сплайн 20  
Куранта умова 168, 185  
**Метод алгебричної прогонки** 98  
— балістичний 133, 153  
— —, лінійні задачі 134, 138  
— —, нелінійні задачі 133, 136  
— Гаусса з вибором головного елемента 96  
— — —, зворотний хід 96, 196  
— — —, прямий хід 95  
— дотичних 79  
— інтерполяції 106  
— ітерованої ваги 33, 36  
— колокації 146, 172  
— комірок 62  
— ламаних 122  
— лінеаризації 108  
— малого параметра 118  
— Монте-Карло 66  
— найменших квадратів 27, 146, 172  
— — —, вибір ваг 27  
— Ньютонa 88, 141, 191  
— парабол 82  
— половинного ділення 75  
— простих ітерацій 77, 86, 141, 197  
— різницевий 153, 162, 166, 170, 177, 183, 191  
— — —, лінійні задачі 134  
— — —, нелінійні задачі 138  
— Рунге—Кутти 123  
— — —, оцінка точності 125  
— січних 81  
— степеневий 111  
**Норма матриці** 107  
Обернена інтерполяція 16  
Оцінка похибки апостеріорна 13  
— апріорна 14  
**Пікара метод** 117  
Погана обумовленість 95, 122  
Поділені різниці 11  
Поліном характеристичний 104  
Поліноміальні формули 40  
Послідовне інтегрування 64  
Похибка апроксимації 164  
— інтерполяції 13  
— неусувна 6  
— округлення 6, 62  
Правило Рунге 43, 52, 128, 165  
Процес Лібмана 170  
Регуляризація числового диференціювання 47  
Середнє квадратичне наближення 24  
— — — дискретне 27  
— — — інтегральне 25  
— — відхилення, дискретне 28  
— — —, інтегральне 28  
Симетричні формули 42  
Сплайн 18  
Сплайн-інтерполяція 18  
Стійкість 165  
— абсолютна (безумовна) 165, 168, 178, 186  
— умовна 165, 168, 178, 184, 185  
Субтаблювання 16  
Схема різницева 166, 170, 177  
— — двошарова 179  
— — неявна 168, 179, 180, 185  
— — явна 166, 178, 184  
— — «хрест» 183  
**Характеристичне рівняння** 104  
**Шаблон** 164

## Зміст

---

<b>Передмова</b> .....	3
<b>Вступ</b> .....	5
<b>Розділ 1. НАБЛИЖЕННЯ ФУНКЦІЙ</b> .....	8
1.1. Інтерполяція.....	8
1.1.1. Лінійна інтерполяція .....	9
1.1.2. Нелінійна інтерполяція .....	16
1.1.3. Сплайн-інтерполяція .....	18
1.1.4. Інтерполяція функції двох змінних .....	22
1.2. Середнє квадратичне наближення .....	24
1.2.1. Лінійна апроксимація .....	25
1.2.2. Нелінійна апроксимація .....	31
1.3. Рівномірне наближення .....	35
<i>Задачі для самостійного розв'язування</i> .....	39
<b>Розділ 2. ЧИСЛОВЕ ДИФЕРЕНЦІЮВАННЯ</b> .....	40
2.1. Поліноміальні формули .....	40
2.2. Симетричні формули .....	42
2.3. Правило Рунге .....	43
2.4. Квазірівномірні сітки .....	45
2.5. Числове диференціювання швидкозмінних функцій .....	46
2.6. Регуляризація числового диференціювання .....	46
<i>Задачі для самостійного розв'язування</i> .....	47
<b>Розділ 3. ЧИСЛОВЕ ІНТЕГРУВАННЯ</b> .....	48
3.1. Інтерполяційні формули .....	49
3.1.1. Формула середніх .....	49
3.1.2. Формула трапецій .....	50
3.1.3. Формула парабол (Сімпсона).....	51
3.1.4. Правило Рунге .....	52
3.1.5. Квадратурна формула Гаусса .....	54

3.1.6. Інтегрування за допомогою сплайнів .....	57
3.1.7. Вибір квадратурних формул та їх збіжність .....	58
3.2. Нестандартні квадратурні формули .....	58
3.2.1. Розривні та знакозмінні функції .....	58
3.2.2. Нелінійні квадратурні формули .....	59
3.2.3. Змінна межа інтегрування .....	59
3.2.4. Невласні інтеграли .....	60
3.3. Кратні інтеграли .....	61
3.3.1. Метод комірок .....	62
3.3.2. Послідовне інтегрування .....	64
3.3.3. Метод Монте-Карло .....	66
3.3.4. Вибір кубатурних формул .....	72
<i>Задачі для самостійного розв'язування</i> .....	73
<b>Розділ 4. НЕЛІНІЙНІ АЛГЕБРИЧНІ РІВНЯННЯ</b> .....	74
4.1. Рівняння з одним невідомим .....	74
4.1.1. Метод половинного ділення .....	75
4.1.2. Вилучення коренів .....	76
4.1.3. Метод простих ітерацій .....	77
4.1.4. Метод дотичних .....	79
4.1.5. Метод січних .....	81
4.1.6. Метод парабол .....	82
4.2. Системи нелінійних рівнянь .....	86
4.2.1. Метод простих ітерацій .....	86
4.2.2. Метод Ньютона .....	88
<i>Задачі для самостійного розв'язування</i> .....	93
<b>Розділ 5. ЗАДАЧІ ЛІНІЙНОЇ АЛГЕБРИ</b> .....	94
5.1. Розв'язання системи лінійних алгебричних рівнянь .....	94
5.1.1. Прямі (точні) методи .....	95
5.1.2. Ітераційні методи .....	100
5.2. Обчислення визначника .....	102
5.3. Відшукування елементів оберненої матриці .....	103
5.4. Проблема власних значень і векторів матриць .....	103
5.4.1. Повна проблема власних значень .....	103
5.4.2. Часткова проблема власних значень .....	108
<i>Задачі для самостійного розв'язування</i> .....	114
<b>Розділ 6. ЗВИЧАЙНІ ДИФЕРЕНЦІАЛЬНІ РІВНЯННЯ</b> .....	115
6.1. Задача Коші .....	116
6.1.1. Наближені методи .....	117

6.1.2. Числові методи .....	121
6.2. Крайова задача .....	132
6.2.1. Числові методи .....	133
6.2.2. Наближені методи .....	146
6.3. Задачі на власні значення .....	152
6.3.1. Числові методи .....	153
6.3.2. Наближені методи .....	154
<i>Задачі для самостійного розв'язування</i> .....	158
<b>Розділ 7. РІВНЯННЯ В ЧАСТИННИХ ПОХІДНИХ</b> .....	159
7.1. Основні поняття .....	161
7.1.1. Вступні зауваження .....	161
7.1.2. Класифікація рівнянь у частинних похідних 2-го порядку .....	162
7.1.3. Різницевий метод розв'язування рівнянь у частинних похідних .....	162
7.1.4. Рівняння 1-го порядку .....	165
7.2. Еліптичні рівняння .....	169
7.2.1. Різницевий метод .....	170
7.2.2. Наближені методи .....	172
7.3. Параболічні рівняння .....	176
7.3.1. Найпростіші різницеві схеми .....	177
7.3.2. Найпростіша двошарова різницева схема .....	179
7.3.3. Багатовимірні задачі .....	181
7.4. Гіперболічні рівняння .....	183
7.4.1. Схема «хрест» .....	183
7.4.2. Неявна схема .....	185
7.4.3. Багатовимірні задачі .....	186
<i>Задачі для самостійного розв'язування</i> .....	188
<b>Розділ 8. ІНТЕГРАЛЬНІ РІВНЯННЯ</b> .....	189
8.1. Вступні зауваження .....	189
8.2. Різницевий метод .....	191
8.2.1. Нелінійні рівняння .....	191
8.2.2. Лінійні рівняння .....	193
8.3. Наближені методи .....	197
8.3.1. Метод простих ітерацій .....	197
8.3.2. Метод Бубнова — Гальоркіна .....	198
<i>Задачі для самостійного розв'язування</i> .....	200
Додатки .....	201
Список рекомендованої літератури .....	249
Предметний покажчик .....	251



---

Темплан 2017, поз. 10

Навчальне видання

Вольєн Григорович Мусіяка

**ОСНОВИ ЧИСЛОВИХ МЕТОДІВ**

підручник

Редактор *В.О. Наскан*  
Техредактор *Т.І. Севост'янова*  
Коректор *О.В. Бец*

---

Підписано до друку 03.10.2017. Формат 60 x 90/16. Папір друкарський. Друк плоский. Ум. друк. арк. 12,5. Ум. фарбовідб. 12,5. Обл.-вид. арк. 14,4. . Тираж 50 прим. Зам. № 235

---

РВВ ДНУ, просп. Гагаріна, 72, м. Дніпро, 49010.  
Видавництво «Ліра ЛТД», вул. Погребняка, 25, м. Дніпро, 49010.  
Свідцтво про внесення до Державного реєстру серія ДК № 188  
від 19.09.2000. Фактична адреса: вул. Наукова, 5

```
+ sBig);
Console.WriteLine("");
Console.WriteLine("The rever
+ oLittle);
Console.WriteLine("The rever
+ oString);

Console.WriteLine("End chang
Console.ReadLine();
}
```

```
x = 0. 0.25 0.5
fi = -0.1111 0.1389 0.3889
y = 0. 0.0625 0.25
C = - 0.1111 1. dskv =
fi(x) = pp = -0.1111 + x
```

```
es(int val)
```

```
BitConverter.GetBytes
```