

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

**А. І. МОМОТ
О. Я. ОЛІХ**

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

**Методичні вказівки
до практичних робіт**



Рецензенти:

канд. фіз.-мат. наук, доц. В. Я. Чолій,
канд. фіз.-мат. наук, доц. О. В. Романенко

*Рекомендовано до друку вченою радою фізичного факультету
(протокол № 8 від 21 лютого 2011 року)*

Момот, А. І.

Математичне моделювання : метод. вказівки до практ. робіт / А. І. Момот, О. Я. Оліх. – К. : Видавничо-поліграфічний центр "Київський університет", 2011. – 72 с.

Наведено короткий огляд числових методів розв'язування алгебраїчних рівнянь і систем лінійних алгебраїчних рівнянь, інтерполяції, апроксимації та інтегрування функцій однієї змінної, а також алгоритми їх практичної реалізації. Викладений матеріал та завдання відповідають робочій програмі дисципліни "Математичне моделювання".

Для студентів фізичного факультету Київського національного університету імені Тараса Шевченка.

Зміст

I. Загальні властивості числових методів	4
II. Алгебраїчні рівняння	10
III. Системи лінійних алгебраїчних рівнянь: методи Гаусса та LU-розвинення	21
IV. Системи лінійних алгебраїчних рівнянь: метод прогонки, ітераційні методи	32
V. Інтерполяція за допомогою поліномів Лагранжа та Ньютона	40
VI. Інтерполяція сплайнами	49
VII. Апроксимація	57
VIII. Числове інтегрування	63
Список літератури	71

Практичне заняття I. Загальні властивості числових методів

Одним із найважливіших параметрів числового розв'язання задач, поряд із простотою обчислень та кількістю математичних операцій, є величина похибки. Під час застосування числових методів похибки виникають завжди. Використовуючи найбільш загальну класифікацію, розрізняють усунні та неусунні похибки. У свою чергу, до усунних належать похибки методу та похибки округлень, до неусунних — похибки побудови математичної моделі та похибки, пов'язані з наближеністю вхідних даних.

Розглянемо деякі особливості обчислювальної математики, які безпосередньо пов'язані з величиною похибки кінцевого результату. До них належать такі.

1. Обчислювальна математика оперує дискретними об'єктами: дійсне число розглядається як число із плаваючою крапкою, відрізок — як система точок, неперервна функція — як таблицна, тобто як набір значень аргументу та функції. Одним із наслідків цього є поява похибки методу.

2. У комп'ютерному представленні числа мають обмежену кількість знаків після коми і це є причиною того, що при обчисленнях з'являтиметься машинна похибка округлення. Розглянемо її появу на прикладі. Припустимо, що необхідно розв'язати рівняння

$$x^4 - 4x^3 + 8x^2 - 16x + 15,99999999 = 0.$$

Один зі шляхів розв'язання — виділити четвертий степінь різниці:

$$(x - 2)^4 - 10^{-8} = 0,$$
$$(x - 2)^2 = \pm 10^{-4}.$$

Таким чином, матимемо чотири розв'язки: $x_1 = 2,01$; $x_2 = 1,99$; $x_3 = 2 + 0,01i$; $x_4 = 2 - 0,01i$. Водночас, якщо

похибка заокруглення $\delta_m > 10^{-8}$, то число 15,99999999 буде замінене на 16 і отримаємо рівняння

$$(x - 2)^4 = 0.$$

Його чотири розв'язки є однаковими $x_1 = x_2 = x_3 = x_4 = 2$ і відрізняються від точних розв'язків.

Для характеристики цього явища вводять поняття машинного епсілон: це таке число ϵ_m , для якого при комп'ютерних розрахунках

$$1 + \epsilon_m = 1.$$

ϵ_m описує відносну похибку машинного заокруглення (або відносну відстань між машинними числами); його значення залежить від того, який обсяг пам'яті виділяється для збереження одного числа.

3. Існує поняття зумовленості розв'язку, тобто чутливості розв'язку до малих змін вхідних даних. Наприклад, система рівнянь

$$\begin{cases} x + 10y = 11; \\ 100x + 1001y = 1101 \end{cases}$$

має розв'язок $x = 1$, $y = 1$. Якщо додати до правої частини першого рівняння всього одну соту (змінити її на 0,01), то матимемо систему

$$\begin{cases} x + 10y = 11,01; \\ 100x + 1001y = 1101 \end{cases}$$

із розв'язком $x = 11,01$, $y = 0$. Тобто, вихідна система погано зумовлена: відносно невелика зміна вхідних даних призвела до значних змін розв'язку.

Зазначимо, що приклад з попереднього пункту також погано зумовлений: абсолютна похибка у вільному члені порядку 10^{-8} призвела до похибок розв'язку близько 10^{-2} .

4. Вибір алгоритму обчислень може впливати на їх результат. Наприклад, якщо обчислювати значення функції $\exp(x)$ за допомогою її розвинення в ряд

$$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots,$$

то виявляється, що для додатних значень аргументу обчислення функції у такий спосіб дає цілком прийнятні значення. Водночас, для отримання правильного результату при $x < 0$ необхідно скористатися тим, що $\exp(-x) = [\exp(x)]^{-1}$, тобто спочатку за допомогою ряду обчислити значення функції для модуля аргументу, а вже потім порахувати величину, обернену до отриманої в результаті обчислення суми.

5. Алгоритм обчислень має бути економічним, тобто з мінімізованою кількістю елементарних операцій. Для ілюстрації знайдемо суму

$$S = 1 + x + x^2 + x^3 + \dots + x^{1023} \quad (0 < x < 1).$$

Якщо обчислювати кожен із доданків як

$$x, x^2 = x \cdot x, x^3 = x^2 \cdot x, \dots, x^{1023} = x^{1022} \cdot x,$$

то потрібно буде виконати 1022 операції множення і 1023 операції додавання, причому кожна з них характеризуватиметься похибками, пов'язаними з машинним округленням результату. З іншого боку, якщо скористатись виразом, який описує суму прогресії

$$S = \frac{1 - x^{1024}}{1 - x},$$

та врахувати, що $x^2 = x \cdot x$, $x^4 = x^2 \cdot x^2$, \dots , $x^{1024} = x^{512} \cdot x^{512}$, то необхідно буде виконати всього 10 операцій множення, одну ділення та дві віднімання.

Завдання I.1. Написати програму пошуку машинного епсілон для чисел типу float та double.

Завдання I.2. Написати програму обчислення функції за допомогою її розвинення у степеневий ряд

$$f(x) = \sum_{k=0}^{\infty} a_k x^k \Rightarrow f(x) \approx \sum_{k=0}^{N-1} a_k x^k, \quad |a_N x^N| < \varepsilon,$$

із точністю $\varepsilon = 10^{-8}$ на прикладі $f(x) = \sin(x)$, $f(x) = \cos(x)$, $f(x) = \exp(x)$.

Примітка:

$$\begin{aligned}\exp(x) &= 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \\ \sin(x) &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \\ \cos(x) &= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots\end{aligned}$$

Завдання I.3. Написати функцію, яка читає дані з файла, що складається зі стовпців чисел, і заповнює ними задані масиви (масив).

Орієнтовні алгоритми

Запропоновані в цьому та наступному розділах алгоритми носять назву “орієнтовні” через те, що вони висвітлюють особливості реалізації саме числових методів і не завжди відображають усі деталі, які потрібно врахувати для того, щоб програма працювала стабільно. Наприклад, перед відкриттям файла бажано перевірити, чи він взагалі існує на диску; під час зчитування числових даних варто переконатися, чи дійсно цифри внесено у файл (набрано користувачем на клавіатурі); якщо очікується, що дані у файлі розташовано у декілька стовпців,

то слід перевірити, чи кожен рядок містить однакову кількість чисел тощо. Але всі ці моменти більше стосуються якості та рівня програмування і в наведених алгоритмах не будуть відображатися.

Завдання I.2

Розглянемо розв'язання завдання на прикладі функції $\sin(x)$. Як відомо з курсу математичного аналізу, її розвинення у ряд можна записати у вигляді

$$\sin(x) = \sum_{k=0}^{\infty} q_k, \quad \text{де } q_k = (-1)^k \frac{x^{2k+1}}{(2k+1)!}.$$

Легко побачити, що, зокрема,

$$q_2 = \frac{x^5}{5!} = \frac{x^3 \cdot x^2}{3! \cdot 4 \cdot 5} = -q_1 \frac{x^2}{4 \cdot 5},$$

і, загалом, два послідовні доданки пов'язані між собою співвідношенням

$$q_{k+1} = -\frac{x^2}{(2k+2)(2k+3)} \cdot q_k, \quad q_0 = x.$$

Тобто, для обчислення k -го доданка не потрібно безпосередньо шукати значення виразів x^{2k+1} і $(2k+1)!$, а можна скористатись отриманим вище співвідношенням, що суттєво зменшить кількість операцій.

Таким чином, сам алгоритм можна записати у вигляді такої послідовності операцій.

- i. У діалоговому режимі зчитати із клавіатури значення аргументу, для якого необхідно розрахувати значення функції.
- ii. Обчислити перший доданок q_0 і присвоїти його значення результату.

- iii. Спираючись на значення попереднього доданка, обчислити наступний.
- iv. Порівняти *модуль* отриманого доданка з ϵ . Якщо значення ϵ більше, то перейти до п. vi.
- v. Додати отриманий доданок до результату. Перейти до п. iii.
- vi. Вивести на екран результат.
- vii. Для порівняння вивести на екран значення функції для цього ж аргументу, обчислене за допомогою вбудованих функцій.

Завдання I.3

- i. У діалоговому режимі отримати з клавіатури назву файла та кількість стовпців.
- ii. Відкрити файл для читання.
- iii. Визначити кількість рядків (наприклад, читаючи дані з файла аж до його закінчення; після досягнення кінця файлу повернути вказівник на початок).
- iv. Створити масив необхідного розміру.
 - v. Зчитати дані з файла в масив.
 - vi. Закрити файл.
- vii. Вивести дані з масиву на екран.

Практичне заняття II. Алгебраїчні рівняння

Задачу розв'язання алгебраїчного рівняння можна звести до пошуку таких значень аргументу функції $f(x)$, при яких значення самої функції дорівнює нулеві,

$$f(x) = 0, \quad x = ?;$$

тобто знайти точки перетину графіка функції $f(x)$ із віссю x . При розв'язанні цієї задачі, насамперед, постає завдання знайти діапазон значень аргументу, де може бути розташований розв'язок. Розглянемо приклад, наведений на рис. 1. Значення функції у точці x_1 є додатним, а у точці x_2 — від'ємним, тому, очевидно, між цими точками знаходиться значення x таке, що $f(x) = 0$. Отже, відрізок $[a, b]$, на якому шукається розв'язок, слід вибрати з умови

$$f(a) \cdot f(b) < 0. \quad (\text{II.1})$$

Умова (II.1) не гарантує, що на відрізку є лише один корінь, наприклад $f(x_1) \cdot f(x_4) < 0$, а коренів між цими точками три. І навпаки, якщо умова (II.1) не виконується, то це може означати, що на вибраному проміжку коренів немає, як, наприклад

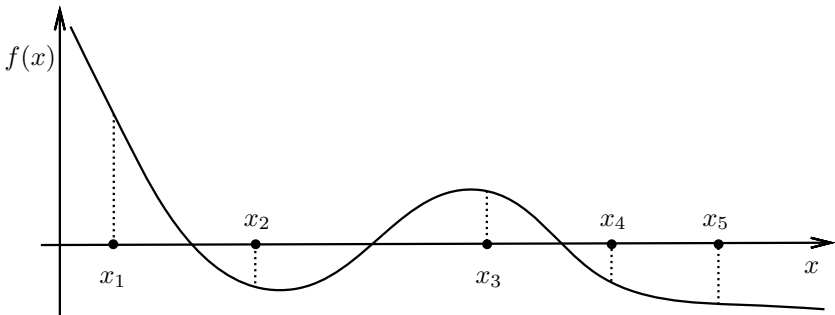


Рис. 1. Приклад можливого взаємного розташування коренів рівняння та границь відрізка (точки x_i), де шукається розв'язок

на відрізку $[x_4, x_5]$, або їх є декілька — відрізок $[x_1, x_3]$. Окремим є випадок $f(a) \cdot f(b) = 0$, коли кінець відрізка збігається із коренем рівняння.

Питання щодо локалізації кореня (тобто про визначення границь відрізка), як правило, алгоритмічно не розв'язується; це, швидше, питання майстерності того, хто проводить обчислення. Проте в багатьох випадках локалізувати корінь досить просто. Один із найпростіших підходів — вибрати початковий відрізок довільної ширини d таким чином, щоб одна з його границь збігалася з точкою $x = 0$, а потім зміщувати його на всю ширину праворуч або ліворуч доти, поки добуток значень функції у його крайніх точках не буде від'ємним або нульовим.

Розглянемо основні методи числового розв'язання алгебраїчних рівнянь.

Метод ділення навпіл (або дихотомія чи бісекція)

Цей метод полягає у тому, що вихідний відрізок ділиться навпіл точкою (рис. 2):

$$c = (a + b)/2. \quad (\text{II.2})$$

Якщо $f(a) \cdot f(c) > 0$, тоді точка c стає лівим кінцем нового відрізка ($a \rightarrow c$, відрізок $[a, b]$ замінюється на $[c, b]$, рис. 2, а); якщо $f(a) \cdot f(c) < 0$, то точка c стає правим кінцем відрізка

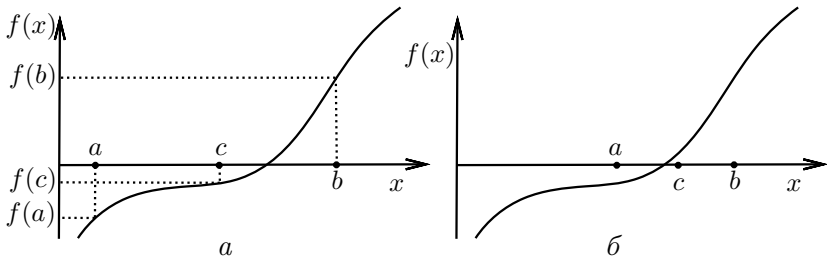


Рис. 2. Ілюстрація до методу ділення навпіл: *а* — перший крок, *б* — другий

($b \rightarrow c$, цей випадок зображено на рис. 2, б). Якщо ж $f(a) \cdot f(c) = 0$, то це означає, що $f(c) = 0$ і точка $c \in$ розв'язком.

Операція поділу повторюється, поки не починає виконуватись одна з таких умов:

$$|f(c)| < \delta; \quad (\text{II.3})$$

$$|b - a| < \delta_x; \quad (\text{II.4})$$

$$|b - a| < \varepsilon_x \cdot b, \quad (\text{II.5})$$

де δ — абсолютна точність визначення функції, а умови (II.4) та (II.5) описують порівняння відрізка з абсолютною та відносною точністю знаходження кореня відповідно. Розв'язком вважається координата однієї із границь кінцевого відрізка. Часто достатньо відслідковувати лише виконання умови (II.4).

За вдалого вибору початкових значень a та b метод збігається завжди. Якщо ж функція має вигляд, подібний до зображеного на рис. 3, то, незважаючи на те, що на відрізку $[a, b]$ умова (II.1) задовольняється, кореня там немає і метод ділення навпіл може дати хибний розв'язок. Аби уникнути подібної ситуації, необхідно контролювати зростання значення функції

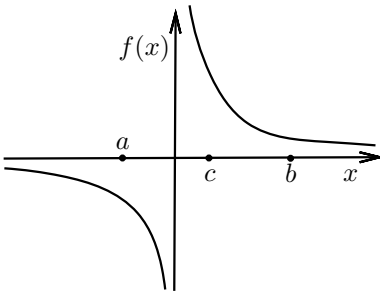


Рис. 3. Графік функції, для якої застосування методу дихотомії призведе до хибного розв'язку

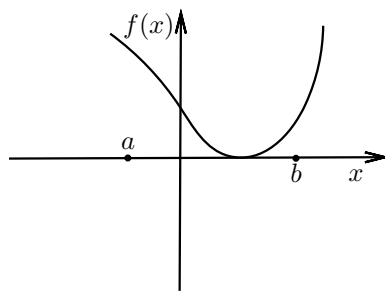


Рис. 4. Графік функції, корінь якої не можна визначити методом ділення навпіл

у точці c , порівняно з $f(a)$ і $f(b)$.

Методом ділення навпіл неможливо також знайти корінь, подібний до зображеного на рис. 4, адже не можна вибрати відрізок $[a, b]$, який містить корінь і при цьому функція має різні знаки на його границях.

Метод хорд

Цей метод відрізняється від методу дихотомії способом пошуку нової границі відрізка — точки c . Положення c визначається з умови

$$c = a - f(a) \cdot \frac{b - a}{f(b) - f(a)}. \quad (\text{II.6})$$

Як видно з рис. 5, точка c є перетином хорди AB з віссю x і у цьому випадку вона знаходиться ближче до кореня, ніж середина відрізка $[a, b]$. Для подібних трикутників Bbc і Aac можна записати відношення $f(b)/(b-c) = -f(a)/(c-a)$, звідки

$$c = \frac{af(b) - bf(a)}{f(b) - f(a)},$$

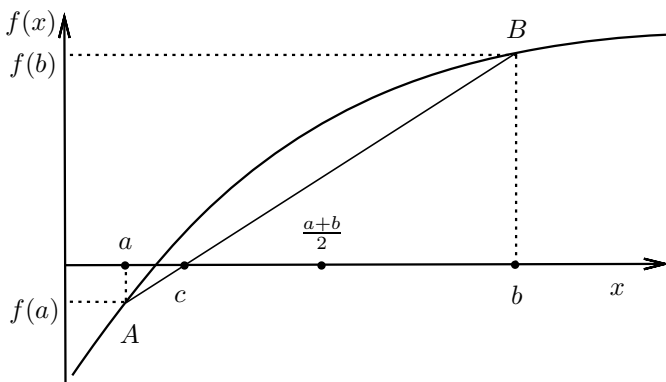


Рис. 5. Ілюстрація до пошуку наступного наближення у методі хорд

що дозволяє легко отримати вираз (II.6). Тобто, у цьому методі при визначенні положення нової границі, окрім знака функції на кінцях відрізка, враховується і її значення.

Після визначення c зміна границі відрізка відбувається так само, як у методі ділення навпіл.

У методі хорд критерій (II.4) незастосовний. Наприклад, у випадку, зображеному на рис. 6, положення точки b не змінюється, точка c із кожним кроком наближається до кореня, але відрізок $[c_k, b]$ не стає малим (c_k — нова границя відрізка, отримана під час k -го кроку). У цьому разі можна порівнювати взаємне розташування точок, отриманих під час послідовних наближень, тобто використовувати умову

$$|c_k - c_{k-1}| < \delta_x, \quad (\text{II.7})$$

або можна застосовувати умову (II.3).

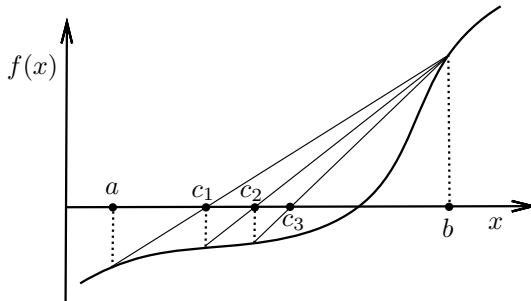


Рис. 6. Приклад функції, для якої у методі хорд довжина відрізка локалізації кореня не може бути меншою за будь-яку наперед задану малу величину

У випадках, проілюстрованих рисунками 3 та 4, при використанні методу хорд виникають ті самі труднощі із пошуком кореня, що і при застосуванні методу бісекції.

Метод Ньютона (або дотичних чи лінеаризації)

У цьому випадку розташування точки, більш наближеної до шуканого розв'язку, вибирається в місці перетину дотичної до графіка функції $f(x)$ при $x = b$ та осі абсцис (рис. 7). Ураховавши, що з одного боку, $\operatorname{tg} \alpha = f'(b)$, а з іншого, $\operatorname{tg} \alpha = f(b)/(b - c)$, отримуємо умову для визначення положення нової границі відрізка:

$$c = b - \frac{f(b)}{f'(b)}. \quad (\text{II.8})$$

Очевидно, функція повинна бути такою, щоб від неї можна було аналітично знайти похідну. Подальше звуження відрізка відбувається так само, як і у попередніх випадках. При цьому мають виконуватись умови

$$\begin{cases} f(b) \cdot f'(b) > 0; \\ b - f(b)/f'(b) > a. \end{cases} \quad (\text{II.9})$$

Перша з них означає, що функція у точці b є додатною і зростає або є від'ємною і спадає. Друга умова вимагає, щоб знайдена точка не виходила за межі відрізка. Якщо ці умови не виконуються, то замість точки $x = b$ можна використовувати іншу границю. Для функції, наведеної на рис. 8, умови (II.9) не виконуються у точках a та b , і тому для методу

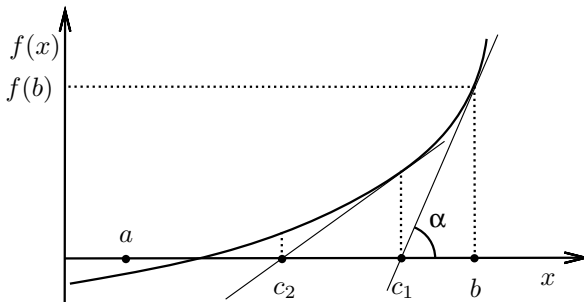


Рис. 7. Спосіб пошуку наступного наближення у методі Ньютона

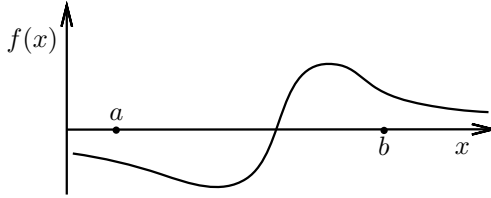


Рис. 8. Приклад невдалого вибору початкового відрізка для методу Ньютона

Ньютона потрібно підбирати інший відрізок. Загалом, використовуючи цей метод, не обов'язково задавати відрізок $[a, b]$, можна шукати розв'язок і навколо однієї точки ($x = b$), яка буде початковим наближенням. Проте збігатись цей метод при будь-якому початковому наближенні буде лише у випадку, коли у всіх точках виконується умова

$$\frac{|f \cdot f''|}{(f')^2} < 1 .$$

У протилежному випадку метод збігатиметься лише тоді, коли початкова точка знаходиться у певному околі розв'язку.

Критерієм припинення ітерацій може бути або умова (II.3), або (II.7), де c_k — наближення до розв'язку, отримане на k -му кроці. Фактично, згідно з (II.8), $c_{k+1} = c_k - \frac{f(c_k)}{f'(c_k)}$.

Метод січних

За змістом цей метод є симбіозом методів Ньютона та хорд. Спільними рисами з методом Ньютона є: можливість знаходити розв'язок в околі однієї точки; наступне наближення розраховується на основі попереднього. Проте, щоб аналітично не обчислювати похідну, використовують різницеву схему, фактично проводячи хорду через точки $f(c_k)$ та $f(c_{k-1})$ (рис. 9). Тобто, застосовується ітераційний процес, під час якого для отримання наступного наближення використовуються резуль-

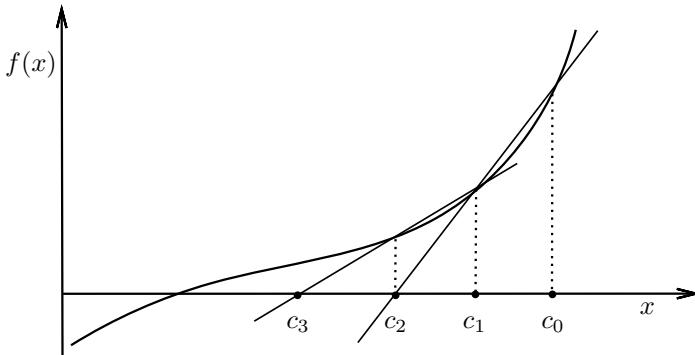


Рис. 9. Спосіб пошуку наступного наближення у методі січних

тати двох (а не одного, як у вищенаведених методах) попередніх кроків:

$$c_{k+1} = c_k - f(c_k) \cdot \frac{c_k - c_{k-1}}{f(c_k) - f(c_{k-1})}. \quad (\text{II.10})$$

Для здійснення першого кроку потрібні дві початкові точки c_0 та c_1 , проте не обов'язково, щоб шуканий корінь рівняння знаходився на проміжку між ними.

У знаменнику виразу (II.10) розміщена різниця $f(c_k) - f(c_{k-1})$. Далеко від кореня ця різниця буде великою, проте при наближенні до кореня вона зменшуватиметься. Як наслідок, при машинних розрахунках виникає втрата значущих цифр, так зване “розхитування” розрахунку. Аби уникнути цього використовують так званий *прийом Гарвіка*: вибирають не дуже мале значення δ_x і проводять ітерації до виконання умови (II.7); після чого ітерації не припиняють, а повторюють доти, поки різниця $|c_k - c_{k-1}|$ продовжує зменшуватись. Перше збільшення цієї різниці означає, як правило, початок “розхитування”, після чого розрахунки припиняють і результат останньої ітерації не використовують.

Метод простої ітерації

У цьому методі рівняння $f(x) = 0$ замінюється еквівалентним

$$x = \varphi(x). \quad (\text{II.11})$$

Наприклад, функція $\varphi(x)$ може мати вигляд

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

де $\psi(x)$ — довільна неперервна знакостала функція. Зрозуміло, що коли x буде коренем вихідного рівняння, то рівність (II.11) також виконуватиметься. У цьому випадку ітераційне співвідношення має вигляд

$$c_{k+1} = \varphi(c_k), \quad (\text{II.12})$$

і обчислення продовжуються, доки не буде виконуватись нерівність (II.7).

Для того, щоб метод збігався, необхідно аби виконувалась умова $|\varphi'(x)| < 1$ на всьому інтервалі, де шукається розв'язок. Один із простих варіантів — вибрати функцію $\psi(x)$ у вигляді константи: $\psi(x) = \tau = \text{const}$. Якщо на відрізку, де шукається розв'язок,

$$0 < m < f'(x) < M$$

(m та M — константи), то для забезпечення збіжності процесу можна покласти

$$\tau = \frac{2}{M + m}.$$

Для цього методу, окрім (II.7), існує ще один критерій припинення ітерацій: відповідь не буде відрізнятися від точного значення кореня більше ніж на δ_x , якщо

$$\frac{(c_k - c_{k-1})^2}{|2c_{k-1} - c_k - c_{k-2}|} < \delta_x. \quad (\text{II.13})$$

Якщо ж у методі простої ітерації покласти $\psi(x) = 1/f'(x)$, то він зведеться до методу Ньютонна.

Завдання II.1. Написати програму пошуку кореня рівняння $f(x) = 0$:

- 1) методом ділення навпіл;
- 2) методом Ньютона;
- 3) методом хорд;
- 4) методом січних;
- 5) методом простої ітерації.

Порівняти отримані результати.

Орієнтовні алгоритми

Завдання II.1

- i. У програмі означити функцію, корені якої шукаються, її похідну (для методу Ньютона) або ітераційну функцію (для методу простої ітерації).
- ii. Задати величини δ , δ_x та максимальну кількість ітерацій N_{iter} .
- iii. Визначити інтервал, на якому відбуватиметься пошук розв'язку.
- iv. Присвоїти нульове значення лічильнику ітерацій i_{iter} .
- v. Використовуючи один із виразів (II.2), (II.8), (II.6), (II.10) або (II.12) (залежно від методу), обчислити наступне наближення до розв'язку.
- vi. Збільшити лічильник ітерацій на одиницю.
- vii. Залежно від значень функції $f(x)$ у точках a , b та c змінити інтервал пошуку розв'язку (для методів дихотомії та хорд).

viii. Перевірити виконання:

- а) умови (II.7) (умови (II.4) для методу бісекції);
- б) умови $i_{\text{iter}} > N_{\text{iter}}$;
- в) умови (II.3).

За виконання будь-якої з умов для методів ділення навпіл, дотичних, хорд та простих ітерацій перейти до п. ix. У протилежному випадку перейти до п. v.

Для методу січних при виконанні умови (II.7) використати прийом Гарвіка. Для цього необхідно запам'ятати різницю наближень, отриманих у двох останніх ітераціях, перейти до п. v і в наступних циклах замість (II.7) перевіряти, чи не зростає ця різниця. У решті випадків виконувати дії, аналогічні діям за іншими методами.

ix. Вивести на екран результат.

Примітка. Із метою оцінювання обчислювальної складності методу на екран можна вивести також значення i_{iter} .

Практичне заняття III. Системи лінійних алгебраїчних рівнянь: методи Гаусса та LU-розвинення

У загальному випадку систему лінійних алгебраїчних рівнянь, у якій кількість рівнянь дорівнює кількості невідомих, можна записати у вигляді

$$\hat{A} \mathbf{x} = \mathbf{f}, \quad (\text{III.1})$$

де $\hat{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$ — матриця коефіцієнтів рівнянь,

$$\mathbf{f} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix} \text{ — вектор вільних членів,}$$

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \text{ — вектор невідомих.}$$

Система (III.1) має розв'язок, причому єдиний, за умови $\det \hat{A} \neq 0$, тобто, коли матриця \hat{A} невинроджена.

Розрізняють методи розв'язання таких систем:

- 1) прямі — отримується точний розв'язок (у припущенні відсутності помилок округлення) за скінченної кількості арифметичних дій;
- 2) ітераційні — розраховується послідовність \mathbf{x}_k (k — номер ітерації), яка збігається до точного розв'язку при $k \rightarrow \infty$.

Спочатку розглянемо декілька прямих методів.

Метод Гаусса (або метод виключення Гаусса)

У цьому методі використовується відома властивість, яка полягає у тому, що якщо в лінійній системі поміняти місцями два рядки, або до одного з рядків додати інший, помножений на якесь число, то розв'язок системи не зміниться. Метод Гаусса складається із прямого ходу і зворотного. Метою прямого ходу є зведення лінійної системи до верхньої трикутної (тобто такої, для якої всі $a_{ij} = 0$ при $i > j$). Необхідні для цього дії виконуються за декілька кроків.

Крок 1. Виключаємо x_1 з усіх рівнянь, починаючи з другого. Для цього до другого рівняння додаємо перше, домножене на $\left(-\frac{a_{21}}{a_{11}}\right)$; до третього — перше, домножене на $\left(-\frac{a_{31}}{a_{11}}\right)$, і т. д. У результаті отримаємо систему рівнянь

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = f_1; \\ \quad a_{22}^{(1)}x_2 + \dots + a_{2n}^{(1)}x_n = f_2^{(1)}; \\ \dots\dots\dots \\ \quad a_{n2}^{(1)}x_2 + \dots + a_{nn}^{(1)}x_n = f_n^{(1)}, \end{array} \right.$$

де $\{a_{ij}^{(1)}\}$ та $\{f_i^{(1)}\}$ — матриця коефіцієнтів рівнянь та вектор вільних членів після першого кроку:

$$a_{ij}^{(1)} = a_{ij} - \frac{a_{i1}}{a_{11}} \cdot a_{1j}, \quad f_i^{(1)} = f_i - \frac{a_{i1}}{a_{11}} \cdot f_1, \quad i, j = \overline{2, n}. \quad (\text{III.2})$$

Крок 2. Виключаємо x_2 з останніх $(n-2)$ рівнянь. Для цього до третього рівняння додаємо друге, домножене на $\left(-\frac{a_{32}^{(1)}}{a_{22}^{(1)}}\right)$; до четвертого — друге, домножене на $\left(-\frac{a_{42}^{(1)}}{a_{22}^{(1)}}\right)$, і т. д. Отри-

$$\left\{ \begin{array}{l} x_n = \frac{f_n^{(n-1)}}{a_{nn}^{(n-1)}}; \\ \dots\dots\dots \\ x_k = \frac{1}{a_{kk}^{(k-1)}} \cdot (f_k^{(k-1)} - a_{k,k+1}^{(k-1)}x_{k+1} - \dots - a_{kn}^{(k-1)}x_n); \\ \dots\dots\dots \\ x_2 = \frac{1}{a_{22}^{(1)}} \cdot (f_2^{(1)} - a_{23}^{(1)}x_3 - \dots - a_{2n}^{(1)}x_n); \\ x_1 = \frac{1}{a_{11}} \cdot (f_1 - a_{12}x_2 - \dots - a_{1n}x_n). \end{array} \right. \quad (\text{III.5})$$

Зазначимо, що визначник матриці \hat{A} можна обчислити як добуток діагональних елементів матриці (III.4)

$$\det \hat{A} = a_{11} \times a_{22}^{(1)} \times a_{33}^{(2)} \times \dots \times a_{nn}^{(n-1)}. \quad (\text{III.6})$$

При використанні методу Гаусса може виникнути ситуація ділення на нуль і подальші обчислення унеможливлуються. Ситуація ділення на дуже мале число також є небажаною, тому що у цьому випадку буде накопичуватись помилка, пов'язана з машинним округленням. Як наслідок, частіше на практиці використовують *метод Гаусса з вибором головного елемента*.

Метод Гаусса з вибором головного елемента

У цьому випадку проводять вибір по стовпцям, а саме: перед виключенням x_1 серед набору $\{a_{i1}\} \quad (i = \overline{1, n})$ шукають найбільший за модулем елемент. Якщо для нього $i = k \neq 1$, то міняють місцями перший та k -й рядки матриці \hat{A} і перший та k -й компоненти вектора \mathbf{f} і лише після цього виконують процедуру виключення першої невідомої (крок 1 прямого ходу). Далі, перед початком другого кроку, подібним чи-

ном знаходять максимальний за модулем елемент серед набору $\{a_{i2}^{(1)}\}$ ($i = \overline{2, n}$), міняють місцями другий та відповідний рядки і т. д. Якщо на одному із кроків не вдається знайти у стовпці відмінний від нуля елемент, то матриця системи є виродженою (визначник дорівнює нулю) і розв'язку не існує.

Схожим способом можна реалізувати вибір по рядках: перед виключенням x_j шукається максимальний за модулем елемент серед набору $\{a_{ji}^{(j-1)}\}$ ($i = \overline{j, n}$), потім за необхідності переставляються стовпці матриці \hat{A} разом із відповідними компонентами вектора \mathbf{x} і лише потім розпочинається домноження та додавання рівнянь (виключення невідомих). Звичайно, у цьому випадку необхідно запам'ятовувати, які невідомі з якими обмінювались місцями.

Найбільш ефективним буде метод Гаусса з одночасним вибором головного елемента по всій матриці (і по стовпцях, і по рядках). Однак потрібно відслідковувати перестановки стовпців, інакше невідомі x_i будуть отримані у неправильному порядку. Також, при обчисленні цим методом детермінанта, слід пам'ятати, що перестановка двох рядків чи двох стовпців призводить до зміни знака детермінанта на протилежний.

Отриманий числовими методами розв'язок $\mathbf{x}^{(1)}$ є наближеним за рахунок похибок заокруглення, але його можна покращити (уточнити). Похибка отриманого розв'язку $\boldsymbol{\varepsilon}^{(1)} = \mathbf{x} - \mathbf{x}^{(1)}$ (\mathbf{x} — точний розв'язок) задовольняє рівняння

$$\hat{A}\boldsymbol{\varepsilon}^{(1)} = \hat{A}\mathbf{x} - \hat{A}\mathbf{x}^{(1)} = \mathbf{f} - \hat{A}\mathbf{x}^{(1)}.$$

Обчисливши праву частину, а потім розв'язавши цю систему рівнянь, можна отримати $\boldsymbol{\varepsilon}^{(1)}$, а отже і уточнений розв'язок:

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \boldsymbol{\varepsilon}^{(1)}.$$

Через те, що при знаходженні $\boldsymbol{\varepsilon}^{(1)}$ знову мали місце похибки заокруглення, то $\mathbf{x}^{(2)}$ не збігається з точним розв'язком і його знову можна покращити, знайшовши $\boldsymbol{\varepsilon}^{(2)} = \mathbf{x} - \mathbf{x}^{(2)}$ і

т. д. Процес уточнення можна продовжувати, доки величина похибки $\varepsilon^{(i)}$ не стане меншою за бажану точність розв'язку.

Метод LU -розвинення

У цьому випадку матриця \hat{A} представляється у вигляді добутку двох трикутних матриць:

$$\hat{A} = \hat{L} \cdot \hat{U}, \quad (\text{III.7})$$

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & u_{nn} \end{pmatrix}. \quad (\text{III.8})$$

Коефіцієнти матриць $\{l_{ij}\}$ та $\{u_{ij}\}$ можна визначити послідовно. А саме, безпосередньо перемноживши перший рядок матриці \hat{L} на матрицю \hat{U} , отримаємо

$$u_{11} = a_{11}, \quad u_{12} = a_{12}, \quad \dots, \quad u_{1n} = a_{1n}.$$

Після множення другого рядка запишемо

$$\begin{aligned} l_{21} &= a_{21}/u_{11}, \\ u_{22} &= a_{22} - l_{21} \cdot u_{12}, \\ &\dots \dots \dots \\ u_{2n} &= a_{2n} - l_{21} \cdot u_{1n}. \end{aligned}$$

За результатом множення третього рядка матриці \hat{L} на матрицю \hat{U} дістанемо

$$l_{31} = \frac{a_{31}}{u_{11}},$$

Потім, підставивши отримане в (III.13), знайти і невідомі:

$$x_k = \frac{1}{u_{kk}} \left(v_k - \sum_{j=k+1}^n u_{kj} x_j \right), \quad k = \overline{(n-1), 1}. \quad (\text{III.15})$$

Задача про розв'язання системи лінійних рівнянь споріднена до задачі про пошук оберненої матриці. Так, щоб знайти матрицю, обернену до \hat{A} , можна розв'язати набір систем рівнянь типу

$$\hat{A} \mathbf{x}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \hat{A} \mathbf{x}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \dots, \quad \hat{A} \mathbf{x}_n = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}.$$

Тоді набір розв'язків $\{\mathbf{x}_i\}$ утворюють стовпці оберненої матриці \hat{A}^{-1} .

Завдання III.1. Написати програму розв'язання системи рівнянь методом Гаусса. За її допомогою розв'язати систему рівнянь, коефіцієнти якої записані у файлі у вигляді

$$\begin{array}{cccccc} a_{11} & a_{12} & \dots & a_{1n} & f_1 \\ a_{21} & a_{22} & \dots & a_{2n} & f_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & f_n \end{array}$$

Завдання III.2. Написати програму розв'язання системи рівнянь методом Гаусса з вибором головного елемента.

Завдання III.3. Написати програму пошуку визначника матриці методом Гаусса.

Завдання III.4. Написати програму пошуку оберненої матриці методом Гаусса.

Орієнтовні алгоритми

Завдання III.1–III.2

- i. Відкрити файл.
- ii. Визначити кількість рівнянь n у системі, порахувавши, наприклад, кількість рядків у файлі.
- iii. Залежно від n створити масив необхідного розміру.
- iv. Перевірити правильність даних у файлі (кількість чисел у кожному рядку), зчитати дані в масив.
- v. Закрити файл.
- vi. Провести прямий хід методу Гаусса. Для цього присвоїти цілій змінній i значення 0.
- vii. Збільшити значення $i \rightarrow i + 1$.
- viii. Серед множини елементів $\{a_{lk}\}$, $i \leq l \leq n$, $i \leq k \leq n$, знайти найбільший за модулем і визначити номер стовпця k_{\max} і номер рядка l_{\max} , де він знаходиться.
- ix. Якщо l_{\max} відрізняється від i , поміняти місцями у масиві даних i -й та l_{\max} -й рядки матриці коефіцієнтів рівняння та відповідні компоненти вектора вільних членів.
- x. Якщо k_{\max} відрізняється від i , поміняти місцями у масиві даних i -й та k_{\max} -й стовпці матриці коефіцієнтів рівняння. Запам'ятати, які стовпці були переставлені.
- xi. Якщо $a_{ii} = 0$, то вивести на екран напис про те, що розв'язку не існує та припинити роботу програми.
- xii. Для всіх коефіцієнтів, які знаходяться в рядку, номер якого більше i , та у стовпці, номер якого більше i , а також

компонентів вектора вільних членів із номером більшим за i , провести заміну:

$$a_{lk} \rightarrow a_{lk}^* = a_{lk} - \frac{a_{li}}{a_{ii}} \cdot a_{ik}, \quad f_l \rightarrow f_l^* = f_l - \frac{a_{li}}{a_{ii}} \cdot f_i, \quad a_{li} = 0, \\ l, k = \overline{(i+1), n}.$$

xiii. Якщо $i < (n - 1)$, перейти до п. vii.

Примітка. При виконанні Завдання III.1 пп. viii–x пропустити.

xiv. Провести зворотний хід методу Гаусса. Для цього надати i значення n .

xv. Розрахувати значення i -ї невідомої за формулою

$$x_i = \frac{1}{a_{ii}} \left(f_i - \sum_{k=i+1}^n a_{ik} x_k \right).$$

Врахувати, що при $i = n$ кількість доданків у сумі нульова.

xvi. Зменшити значення $i \rightarrow i - 1$.

xvii. Якщо $i > 0$, то перейти до п. xv.

xviii. Вивести на екран (або у файл) значення всіх невідомих, з урахуванням перестановок стовпців, які були виконані, при виборі головного елемента.

Завдання III.3

- i. Виконати пункти і–хiii алгоритму Завдання III.1–III.2 (окрім дій, які стосуються компонент вектора вільних членів). Порахувати всі перестановки рядків і стовпців, які були виконані при виборі головного елемента, і зберегти їх у змінній s .
- ii. Обчислити визначник матриці за формулою

$$\det \hat{A} = (-1)^s \prod_{i=1}^n a_{ii}.$$

- iii. Вивести на екран (або у файл) результат.

Завдання III.4

- i. Виконати пункти і–v алгоритму *Завдання III.1–III.2*.
- ii. Присвоїти цілій змінній значення $j = 1$.
- iii. Присвоїти компонентам вектора вільних членів значення

$$f_l = \begin{cases} 1, & l = j; \\ 0, & l \neq j. \end{cases}$$

- iv. Виконати пункти vi–xvii алгоритму *Завдання III.1–III.2*.
- v. Запам'ятати отриманий вектор розв'язків як j -й стовпець оберненої матриці.
- vi. Збільшити значення $j \rightarrow j + 1$.
- vii. Якщо $j < n$, то перейти до п. iii.
- viii. Вивести на екран (або у файл) результат.

Практичне заняття IV. Системи лінійних алгебраїчних рівнянь: метод прогонки, ітераційні методи

Метод прогонки

Це також прямий метод розв'язання системи лінійних алгебраїчних рівнянь. Він є модифікацією методу Гаусса для окремого випадку розріджених систем — систем із тридіагональною матрицею. Сама система рівнянь виглядає так:

$$\left\{ \begin{array}{rcl} b_1 x_1 + c_1 x_2 & = & d_1 ; \\ a_2 x_1 + b_2 x_2 + c_2 x_3 & = & d_2 ; \\ & a_3 x_2 + b_3 x_3 + c_3 x_4 & = d_3 ; \\ & \dots \dots \dots & \dots \dots \dots \\ & & a_{n-1} x_{n-2} + b_{n-1} x_{n-1} + c_{n-1} x_n = d_{n-1} ; \\ & & a_n x_{n-1} + b_n x_n = d_n . \end{array} \right. \quad (\text{IV.1a})$$

Або її можна записати у вигляді

$$\begin{aligned} a_i x_{i-1} + b_i x_i + c_i x_{i+1} &= d_i, \quad i = \overline{1, n}, & (\text{IV.16}) \\ a_1 = c_n &= 0. \end{aligned}$$

Такі системи часто виникають під час розв'язання різних задач математичного моделювання, наприклад, при інтерполяції базовими сплайнами чи розв'язанні диференціальних рівнянь у частинних похідних. Назва “тридіагональна” пов'язана з тим, що у матриці коефіцієнтів такої системи ненульові члени розміщені лише на центральній діагоналі і двох, розташованих поруч із нею.

Давайте спробуємо розв'язати цю систему методом Гаусса, тобто перетворити матрицю коефіцієнтів на верхню трикутну. Зазначимо, що в цьому випадку кожен із $n - 1$ кроків складатиметься лише з однієї дії, а саме: під час k -го кроку ($k = \overline{1, (n - 1)}$) виключається x_k з $(k + 1)$ -го рівняння і з цією метою до вказаного рівняння додається k -те, домножене на $\left(-\frac{a_{k+1}}{b_k^*}\right)$, де b_k^* — коефіцієнт перед змінною x_k у k -му,

У свою чергу, (IV.3) можна переписати у вигляді

$$\begin{aligned} x_n &= \frac{d_n - a_n B_{n-1}}{b_n + a_n A_{n-1}} = B_n. \\ x_i &= A_i x_{i+1} + B_i, \quad i = \overline{(n-1), 1} \end{aligned} \quad (\text{IV.5})$$

Таким чином, розв'язання тридіагональної системи рівнянь методом прогонки розбивається на два етапи:

- а) пряма прогонка, під час якої за формулами (IV.4) розраховуються прогоночні коефіцієнти A_i та B_i ;
- б) зворотна прогонка, коли за формулами (IV.5) розраховуються самі невідомі.

Для стійкості алгоритму прогонки необхідно, щоб, по-перше, $|A_1| = \frac{c_1}{b_1} < 1$, і, по-друге, виконувалась умова діагонального домінування по рядках: $|b_n| \geq |a_n| + |c_n|$, причому хоча б для одного рядка нерівність має бути строгою.

Детермінант матриці коефіцієнтів у цьому випадку, як і у методі Гаусса, можна знайти як добуток діагональних елементів тридіагональної матриці:

$$\det \hat{A} = \prod_{i=1}^n b_i^* = - \prod_{i=1}^n \frac{c_i}{A_i} = \prod_{i=1}^n (b_i + A_{i-1} a_i). \quad (\text{IV.6})$$

При використанні виразу (IV.6) вважають, що $A_0 = 0$.

Ітераційні методи

У цьому випадку, як і при розв'язанні звичайних алгебраїчних рівнянь, використовується заміна рівняння (III.1) на еквівалентне:

$$\mathbf{x} = \hat{B} \mathbf{x} + \mathbf{F}, \quad (\text{IV.7})$$

де $\hat{B} = \hat{E} - \tau \hat{A}$, $\mathbf{F} = \tau \mathbf{f}$ (\hat{E} — одинична матриця, τ — скалярний множник).

Тоді розв'язок шукають у вигляді послідовності наближень

$$\mathbf{x}^{(k+1)} = \hat{B} \mathbf{x}^{(k)} + \mathbf{F}, \quad (\text{IV.8})$$

причому початкове наближення до розв'язку $\mathbf{x}^{(0)}$ часто вибирають у вигляді нульового вектора. Процес збігатиметься, якщо всі власні значення матриці \hat{B} за абсолютною величиною менші одиниці. Критерієм закінчення ітераційного процесу може виступати виконання однієї із трьох нерівностей:

$$\left| \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)} \right| = \sqrt{\sum_{i=1}^n \left(x_i^{(k)} - x_i^{(k-1)} \right)^2} < \varepsilon, \quad (\text{IV.9a})$$

$$\max_{1 \leq i \leq n} \left| x_i^{(k)} - x_i^{(k-1)} \right| < \varepsilon, \quad (\text{IV.9б})$$

$$\max_{1 \leq i \leq n} \left| \frac{x_i^{(k)} - x_i^{(k-1)}}{x_i^{(k)}} \right| < \varepsilon. \quad (\text{IV.9в})$$

При $|x_i| \gg 1$ варто застосовувати саме критерій (IV.9в) .

Підібрати скалярний коефіцієнт τ можна, спираючись на вигляд матриці \hat{A} . А саме, матрицю коефіцієнтів можна представити у вигляді

$$\hat{A} = \hat{L} + \hat{D} + \hat{U},$$

де \hat{L} та \hat{U} — нижня та верхня трикутні матриці з нульовими елементами на головній діагоналі, \hat{D} — діагональна матриця. Тоді для запису \hat{B} та \mathbf{f} можна використати або вираз

$$\hat{B} = -\hat{D}^{-1}(\hat{L} + \hat{U}), \quad \mathbf{F} = \hat{D}^{-1} \mathbf{f}, \quad (\text{IV.10})$$

що відповідатиме *методу Якобі* (інша назва — *метод простої ітерації*), або

$$\hat{B} = -(\hat{L} + \hat{D})^{-1} \hat{U}, \quad \mathbf{F} = (\hat{L} + \hat{D})^{-1} \mathbf{f}, \quad (\text{IV.11})$$

як це робиться у *методі Зайделя (Гаусса—Зайделя)*. Якщо підставити (IV.10) та (IV.11) у (IV.8) і записати рівності для

окремих компонент вектора невідомих, то вони виглядатимуть більш наочно. А саме, у методі Якобі зв'язок між компонентами вектора невідомих на різних кроках ітерації описується рівностями

$$\begin{cases} x_1^{(k+1)} = -\frac{1}{a_{11}}(a_{12}x_2^{(k)} + a_{13}x_3^{(k)} + \dots + a_{1n}x_n^{(k)} - f_1); \\ x_2^{(k+1)} = -\frac{1}{a_{22}}(a_{21}x_1^{(k)} + a_{23}x_3^{(k)} + \dots + a_{2n}x_n^{(k)} - f_2); \\ \dots \\ x_n^{(k+1)} = -\frac{1}{a_{nn}}(a_{n1}x_1^{(k)} + a_{n2}x_2^{(k)} + \dots + a_{nn-1}x_{n-1}^{(k)} - f_n), \end{cases} \quad (\text{IV.12})$$

а при використанні методу Зайделя рівностями

$$\begin{cases} x_1^{(k+1)} = -\frac{1}{a_{11}}(a_{12}x_2^{(k)} + a_{13}x_3^{(k)} + \dots + a_{1n}x_n^{(k)} - f_1); \\ x_2^{(k+1)} = -\frac{1}{a_{22}}(a_{21}x_1^{(k+1)} + a_{23}x_3^{(k)} + \dots + a_{2n}x_n^{(k)} - f_2); \\ \dots \\ x_n^{(k+1)} = -\frac{1}{a_{nn}}(a_{n1}x_1^{(k+1)} + a_{n2}x_2^{(k+1)} + \dots + a_{nn-1}x_{n-1}^{(k+1)} - f_n). \end{cases} \quad (\text{IV.13})$$

Тобто, у методі Зайделя при обчисленні чергового наближення i -ї компоненти вектора невідомих використовуються значення інших компонент із номером, меншим за i , отримані на цьому ж кроці, а також значення компонент із номером, більшим за i , отримані під час попереднього наближення. Водночас у методі Якобі використовуються лише значення, отримані під час попереднього кроку ітерації.

Таким чином, розв'язання системи рівнянь ітераційним шляхом має здійснюватись таким чином:

1) вибирається початкове наближення до розв'язку, наприклад у вигляді нульового вектора;

- 2) спираючись на попереднє наближення за допомогою формул (IV.12) або (IV.13), розраховується наступне наближення;
- 3) повторюється попередній пункт доти, поки не буде виконуватись умова (IV.9).

Для збіжності достатньо щоб

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|, \quad i = \overline{1, n}, \quad (\text{IV.14})$$

і щоб хоча б для одного з рівнянь системи рівність була строга.

Необхідною умовою виконання процесу є: $a_{ii} \neq 0, i = \overline{1, n}$. За умови, що $\det \hat{A} \neq 0$, цього завжди можна досягти перестановкою рядків та стовпців.

Завдання IV.1. Написати програму розв'язання тридіагональної системи рівнянь методом прогонки.

Завдання IV.2. Написати програму розв'язання системи рівнянь методом Зайделя.

Орієнтовні алгоритми

У запропонованих алгоритмах вважається, що коефіцієнти систем рівнянь знаходяться у файлі на диску комп'ютера.

Завдання IV.1

- i. Відкрити файл.
- ii. Визначити кількість рівнянь n у системі, підрахувавши, наприклад, кількість рядків у файлі.
- iii. Залежно від n створити масив дійсних чисел необхідного розміру.

Примітка. Найпростіший варіант — створити двовимірний масив розміром $n \times (n + 1)$. Проте більшість комірок масиву будуть містити нулі, тому для раціонального використання оперативної пам'яті, особливо при великих n ,

краще створити чотири одновимірні масиви з n дійсних чисел.

- iv. Перевірити правильність даних у файлі, зчитати дані в масив.
- v. Закрити файл.
- vi. Користуючись виразами (IV.4), обчислити коефіцієнти A_i та B_i , $i = \overline{1, (n-1)}$.
- vii. Користуючись виразами (IV.5), обчислити невідомі.
- viii. Вивести на екран (або у файл) отримані результати.

Завдання IV.2

- i. Відкрити файл.
- ii. Визначити кількість рівнянь n у системі, підрахувавши, наприклад, кількість рядків у файлі.
- iii. Залежно від n створити масив дійсних чисел необхідного розміру.
- iv. Перевірити правильність даних у файлі, зчитати дані в масив.
- v. Закрити файл.
- vi. Упорядкувати матрицю коефіцієнтів (провести вибір головного елемента для кожного мінора). Для цього присвоїти цілій змінній i значення 0.
- vii. Збільшити значення $i \rightarrow i + 1$.
- viii. Серед множини елементів $\{a_{lk}\}$, $i \leq l \leq n$, $i \leq k \leq n$, знайти найбільший за модулем і визначити номер стовпця k_{\max} і номер рядка l_{\max} , у яких він знаходиться.

- ix. Якщо l_{\max} відрізняється від i , поміняти місцями у масиві даних i -й та l_{\max} -й рядки матриці коефіцієнтів рівняння та відповідні компоненти вектора вільних членів.
- x. Якщо k_{\max} відрізняється від i , поміняти місцями у масиві даних i -й та k_{\max} -й стовпці матриці коефіцієнтів рівняння.
- xi. Якщо $a_{ii} = 0$, то вивести на екран напис про те, що розв'язок не існує, та припинити роботу програми.
- xii. Якщо $i < (n - 1)$, перейти до п. vii.
- xiii. Занулити лічильник ітерацій $i_{\text{iter}} = 0$.
- xiv. Задати максимальну кількість ітерацій N_{iter} та мінімальне відхилення двох послідовних ітерацій ϵ .
- xv. Присвоїти вектору невідомих початкове наближення: $x_i = 0, \quad i = \overline{1, n}$.
- xvi. Присвоїти змінній i значення 1.
- xvii. Розрахувати наступне наближення i -ї змінної:

$$x_i \rightarrow x_i = -\frac{1}{a_{ii}} \left(\sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j - f_i \right).$$

- xviii. Збільшити $i \rightarrow i + 1$.
- xix. Якщо $i \leq n$, то перейти до п. xvii.
- xx. Збільшити $i_{\text{iter}} \rightarrow i_{\text{iter}} + 1$.
- xxi. Якщо $i_{\text{iter}} > N_{\text{iter}}$, то вивести на екран напис про те, що розв'язок не збігається і припинити роботу програми.
- xxii. Перевірити виконання умови (IV.9в). Якщо вона виконується, то вивести на екран (або у файл) отримані результати; якщо — ні, то перейти до п. xvi.

Практичне заняття V. Інтерполяція за допомогою поліномів Лагранжа та Ньютона

Задачу інтерполяції можна сформулювати так: для заданої таблиці значень $\{x_i, y_i = f(x_i)\}$, $i = \overline{0, n}$, побудувати інтерполяційну функцію $\Phi(x)$, щоб

$$\Phi(x_i) = y_i, \quad i = \overline{0, n}. \quad (\text{V.1})$$

Точки з координатами (x_i, y_i) називаються вузлами інтерполяції, вони можуть бути, наприклад, результатом числового розрахунку функції $f(x)$. Графік функції $\Phi(x)$ проходить через вузли (рис. 10). Побудова функції $\Phi(x)$ дає можливість оцінити значення функції $f(x)$ між вузлами, тобто там, де її точне значення невідоме або його безпосереднє обчислення є ресурсомістким чи займає багато часу. Очевидно, що розв'язок задачі інтерполяції не є єдиним. Часто функцію $\Phi(x)$ шукають у вигляді полінома.

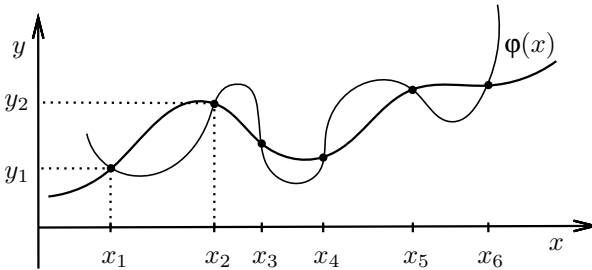


Рис. 10. Варіанти інтерполяції заданих (крапки на графіку) вузлів

Метод Лагранжа

Для спрощення припустимо, що нам потрібно побудувати інтерполяційний поліном для таблиці із трьох пар значень —

$(x_0, y_0), (x_1, y_1), (x_2, y_2)$. Розглянемо функцію

$$L_0(x) = y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}.$$

Ця функція задовольняє умову (V.1) у точці x_0 , а в інших вузлах дорівнює нулю, тобто $L_0(x_0) = y_0$, $L_0(x_1) = 0$, $L_0(x_2) = 0$. Аналогічно запишемо функції

$$L_1(x) = y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)},$$

$$L_2(x) = y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)},$$

які задовольняють умову (V.1) у двох інших вузлах. Очевидно, що графік функції $L(x)$, яка є сумою трьох попередніх

$$\begin{aligned} L(x) &= L_0(x) + L_1(x) + L_2(x) = \\ &= y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + \\ &\quad + y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}, \end{aligned}$$

буде проходити через усі три вузли. Узагальнимо отриману формулу на випадок $(n + 1)$ вузла інтерполяції

$$L(x) = \sum_{i=0}^n y_i \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)}. \quad (\text{V.2})$$

Вираз (V.2) є інтерполяційним поліномом Лагранжа, який використовується для інтерполяції на відрізку $[x_0, x_n]$. Зрозуміло, що це поліном n -го порядку.

Метод Ньютона

У методі Ньютона нам знадобиться поняття *скінченних різниць*. Воно використовується у випадку, коли точки x_i рівновіддалені, тобто $x_i - x_{i-1} = h = \text{const}$ ($i = \overline{1, n}$). Набір вузлів

із такою властивістю ще називається рівномірною мережею, а величина h — кроком мережі. Для нерівномірної мережі розглядаються розділені різниці, але про них ми говорити не будемо.

Розрізняють скінченні різниці вперед і назад.

Скінченні різниці вперед можуть мати різні порядки, можуть обчислюватись для різних вузлів і описуються такими виразами.

Різниці першого порядку (або перші різниці):

$$\begin{aligned} \Delta^{(1)} y_0 &= y_1 - y_0 = f(x_0 + h) - f(x_0), \\ \Delta^{(1)} y_1 &= y_2 - y_1 = f(x_0 + 2h) - f(x_0 + h), \\ &\dots\dots\dots \\ \Delta^{(1)} y_{n-1} &= y_n - y_{n-1} = f(x_0 + nh) - f(x_0 + (n-1)h). \end{aligned} \tag{V.3}$$

Із виразу (V.3) видно, що скінченні різниці вперед першого порядку можна визначити для всіх вузлів, окрім останнього.

Для різниць вперед другого порядку справедливо:

$$\begin{aligned} \Delta^{(2)} y_0 &= \Delta^{(1)} y_1 - \Delta^{(1)} y_0, \\ \Delta^{(2)} y_1 &= \Delta^{(1)} y_2 - \Delta^{(1)} y_1, \\ &\dots\dots\dots \\ \Delta^{(2)} y_{n-2} &= \Delta^{(1)} y_{n-1} - \Delta^{(1)} y_{n-2}. \end{aligned} \tag{V.4}$$

Різниці вперед другого порядку можна обчислити для всіх точок, окрім двох останніх.

У загальному випадку різниця вперед порядку k виражається формулою

$$\Delta^{(k)} y_i = \Delta^{(k-1)} y_{i+1} - \Delta^{(k-1)} y_i, \quad i = \overline{0, n-k}, \tag{V.5}$$

причому $\Delta^{(0)} y_i = y_i$ і цю різницю можна записати для $(n+1-k)$ перших вузлів. Найстарший порядок різниці вперед, який можна визначити для набору $\{x_i, y_i\}$, $i = \overline{0, n}$ — це

n -й, причому таку різницю можна обчислити лише для вузла (x_0, y_0) .

У виразах (V.3)–(V.5) використовується рекурсія: для обчислення різниці будь-якого порядку необхідно розрахувати різниці всіх попередніх порядків. Без рекурсії скінченні різниці вперед обчислюються за формулою

$$\Delta^{(k)} y_i = \sum_{s=0}^k (-1)^{s+k} \frac{k!}{s!(k-s)!} y_{i+s}. \quad (\text{V.6})$$

Аналогічно обчислюються *скінченні різниці назад*:

$$\begin{aligned} \nabla^{(1)} y_i &= \nabla^{(0)} y_i - \nabla^{(0)} y_{i-1} = y_i - y_{i-1}, \\ \nabla^{(2)} y_i &= \nabla^{(1)} y_i - \nabla^{(1)} y_{i-1} = y_i - 2y_{i-1} + y_{i-2}, \\ &\dots\dots\dots \\ \nabla^{(k)} y_i &= \nabla^{(n-1)} y_i - \nabla^{(n-1)} y_{i-1}. \end{aligned} \quad (\text{V.7})$$

Подібно до попередніх виразів, для набору вузлів $\{x_i, y_i\}$, $i = \overline{0, n}$, скінченну різницю назад k -го порядку можливо знайти лише для $(n+1-k)$ останніх точок; найстарший порядок дорівнює n і таку різницю можливо обчислити лише для точки (x_n, y_n) .

При побудові інтерполяційного полінома Ньютона використовуються саме скінченні різниці. Так, поліном Ньютона для інтерполяції вперед має вигляд

$$\begin{aligned} N(x) &= y_0 + \frac{\Delta^{(1)} y_0}{h} (x - x_0) + \frac{\Delta^{(2)} y_0}{2! h^2} (x - x_0)(x - x_1) + \\ &+ \dots + \frac{\Delta^{(n)} y_0}{n! h^n} (x - x_0)(x - x_1) \dots (x - x_{n-1}), \end{aligned} \quad (\text{V.8a})$$

тобто при його побудові використовуються скінченні різниці вперед від 0-го до n -го порядку, обчислені для початкової точки $(\Delta^{(i)} y_0, i = \overline{0, n})$.

Якщо ввести величину $t = \frac{x - x_0}{h}$, тоді

$$\begin{aligned} x &= x_0 + th, \\ \frac{x - x_1}{h} &= \frac{(x_0 + th) - (x_0 + h)}{h} = \frac{th - h}{h} = t - 1, \\ \frac{x - x_2}{h} &= t - 2, \\ &\dots\dots\dots \\ \frac{x - x_{n-1}}{h} &= t - n + 1, \end{aligned}$$

і вираз (V.8a) можна записати у вигляді

$$\begin{aligned} N(x) &= y_0 + t \cdot \Delta^{(1)}y_0 + \frac{t(t-1)}{2!} \cdot \Delta^{(2)}y_0 + \\ &+ \dots + \frac{t(t-1)\dots(t-n+1)}{n!} \cdot \Delta^{(n)}y_0. \end{aligned} \quad (\text{V.8б})$$

Поліном Ньютона також є поліномом n -го порядку. Використовуючи формулу (V.8a) або (V.8б) можна інтерполювати функцію $y = f(x)$ на всьому відрізку зміни аргументу $[x_0, x_n]$, але з погляду підвищення точності розрахунків більш доцільно її використовувати для обчислення значення функції у точках лівої половини відрізка. Для правої половини краще використовувати інтерполяційний поліном Ньютона для інтерполяції назад:

$$\begin{aligned} N(x) &= y_n + t' \cdot \nabla^{(1)}y_n + \frac{t'(t'+1)}{2!} \cdot \nabla^{(2)}y_n + \\ &+ \dots + \frac{t'(t'+1)\dots(t'+n-1)}{n!} \cdot \nabla^{(n)}y_n, \end{aligned} \quad (\text{V.9})$$

де

$$t' = \frac{x - x_n}{h} < 0.$$

Різні способи побудови поліномів Лагранжа і Ньютона дають тотожні інтерполяційні формули. Але поліном Ньютона

зручніше застосовувати у випадку, коли кількість вузлів інтерполяції не є сталою, а може збільшуватись.

Завдання V.1. Написати програму інтерполяції заданого набору даних за методом Лагранжа.

Завдання V.2. Написати процедуру обчислення скінченних різниць вперед для заданого набору даних із використанням рекурсії.

Завдання V.3. Написати процедуру обчислення скінченних різниць вперед без використання рекурсії.

Завдання V.4. Написати програму інтерполяції заданого набору даних за методом Ньютона (формула інтерполяції вперед).

Орієнтовні алгоритми

У запропонованих алгоритмах вважається, що набір даних знаходиться у текстовому файлі на диску комп'ютера, а результатом інтерполяції є набір точок $\{x_j, \varphi(x_j)\}$, де $j = \overline{1, N}$, $N \gg n$, $x_j \in [x_0, x_n]$, тобто кількість точок $\{x_j\}$, в яких обраховується значення інтерполяційної функції $\{\varphi(x_j)\}$, має значно перевищувати кількість точок вихідного набору.

Завдання V.1

- i. Відкрити файл.
- ii. Визначити кількість точок, по яких буде проводитись інтерполяція.

Примітка. У формулах, наведених у теоретичній частині, вважається, що точки нумеруються з нуля, тому кількість точок буде на одиницю більша ніж n .

- iii. Створити двовимірний масив потрібного розміру.

- iv. Зчитати дані у масив.
- v. Закрити файл.
- vi. Визначити мінімальне x_{\min} та максимальне x_{\max} значення аргументу.
- vii. Визначити крок інтерполяції $\delta x = (x_{\max} - x_{\min})/N$, де N — кількість точок, у яких буде розраховане значення інтерполяційної функції (рекомендовано вибирати $N \geq 100$).
- viii. Присвоїти цілій змінній k значення 0.
- ix. Відкрити файл для внесення результатів.
- x. Обчислити абсцису x_k^{int} інтерполяційної функції

$$x_k^{\text{int}} = x_{\min} + \delta x \cdot k.$$
- xi. Якщо $x_k^{\text{int}} > x_{\max}$, то перейти до п. xvi.
- xii. Користуючись формулою (V.2), обчислити ординату інтерполяційної функції $y_k^{\text{int}} = L(x_k^{\text{int}})$.
- xiii. Записати значення $(x_k^{\text{int}}, y_k^{\text{int}})$ у файл.
- xiv. Збільшити значення $k \rightarrow k + 1$.
- xv. Перейти до п. x.
- xvi. Закрити файл із результатами.

Отриманий файл зі значеннями інтерполяційної функції можна використати для побудови графіка, наприклад за допомогою програми *OriginPro*.

Завдання V.2

Назва функції:

$FD()$

— повертає дійсну величину.

Аргументи функції:

$\{x_i, y_i\}, i = \overline{0, n}$

— набір точок (двовірний масив дійсних чисел) для інтерполяції або вказівник на них;

k

— порядок різниці;

J

— номер точки, для якої визначається різниця.

Назва функції є абрєвіатурою від слів finite difference (скінченна різниця). Тоді для $FD(\{x_i, y_i\}, k, J)$ послїдовнїсть виконання описується дуже коротко.

- i. Перевїрити коректнїсть отриманих даних, тобто виконання умов $k \geq 0$, $J \geq 0$ та $(k + J) \leq n$. У протилежному випадку вивести вїдповїдне повїдомлення, присвоїти FD значення 0 і припинити виконання функції.
- ii.

$$FD(k, J) = \begin{cases} y_J, & \text{якщо } k = 0; \\ FD(k - 1, J + 1) - FD(k - 1, J), & \text{якщо } k \geq 1. \end{cases}$$

Завдання V.4

- i. Вїдкрити файл.
- ii. Визначити кїлькїсть точок, по яких буде проводитись їнтерполяція.
Примїтка. У формулах, наведених у теоретичнїй частинї, вважається, що точки нумеруються з нуля, тому кїлькїсть точок буде на одиницю бїльша нїж n .
- iii. Створити двомїрний масив потрїбного розмїру.
- iv. Зчитати данї у масив.
- v. Закрити файл.
- vi. Перевїрити рївномїрнїсть мережї, порївнюючи рїзницю $(x_1 - x_0)$ з їншими рїзницями $(x_{i+1} - x_i)$ для $i = \overline{1, (n - 1)}$. Якщо рївномїрнїсть не виконується, то вивести вїдповїдне повїдомлення і завершити виконання програми.

Примїтка. При порївняннї дїйсних чисел необїднo враховувати похибку машинного округлення.

- vii. Знайти крок мережі h .
- viii. Визначити мінімальне x_{\min} та максимальне x_{\max} значення аргументу.
- ix. Визначити крок інтерполяції $\delta x = (x_{\max} - x_{\min})/N$, де N — кількість точок, у яких буде розраховане значення інтерполяційної функції (рекомендовано вибирати $N \geq 100$).
- x. Присвоїти цілій змінній k значення 0.
- xi. Відкрити файл для внесення результатів.
- xii. Обчислити абсцису x_k^{int} інтерполяційної функції

$$x_k^{\text{int}} = x_{\min} + \delta x \cdot k.$$
- xiii. Якщо $x_k^{\text{int}} > x_{\max}$, то перейти до п. xviii.
- xiv. Користуючись формулою (V.8a) (або (V.8b)), обчислити ординату інтерполяційної функції y_k^{int} . У цьому випадку доцільно створити окрему функцію для розрахунку кінцевих різниць, а також обчислювати наступний доданок у сумі, використовуючи значення попереднього.
- xv. Записати значення $(x_k^{\text{int}}, y_k^{\text{int}})$ у файл.
- xvi. Збільшити значення $k \rightarrow k + 1$.
- xvii. Перейти до п. xii.
- xviii. Закрити файл із результатами.

Отриманий файл зі значеннями інтерполяційної функції можна використати для побудови графіка, наприклад за допомогою програми *OriginPro*.

Практичне заняття VI. Інтерполяція сплайнами

У випадку, коли кількість вузлових точок $\{x_i, y_i = f(x_i)\}$, $i = \overline{0, n}$, є досить великою, то при інтерполяції методом Лагранжа або Ньютона буде отримано поліном високого порядку (ступінь полінома на одиницю менший за кількість точок). У такому випадку часто спостерігаються значні розходження між значеннями функції $f(x)$ та значеннями інтерполяційних поліномів. Щоб уникнути цього можна використовувати не один поліном високого порядку, а набір поліномів низького (часто третього) порядку, кожен із яких використовується для інтерполяції $f(x)$ в різних частинах $[x_0, x_n]$. Набір таких поліномів називається сплайном.

Кубічні сплайн-функції

Це спеціальним чином побудовані поліноми третього порядку. Вони є моделлю тонкого стрижня з пружного матеріалу. Якщо цей стрижень закріпити у двох сусідніх вузлах інтерполяції і задати в цих точках кути нахилу, то він набуде певної форми для мінімізації своєї потенціальної енергії.

У цьому випадку кожен проміжок між сусідніми вузлами інтерполюється своїм поліномом третього порядку, тобто цих поліномів на одиницю менше ніж точок, по яких відбувається інтерполяція. Аналітичний вигляд для кожного з них є таким:

$$S_i(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3, \quad (\text{VI.1})$$

$$x_{i-1} \leq x \leq x_i, \quad i = \overline{1, n}.$$

Для того, щоб записати їх явний вигляд, необхідно знати $4n$ величин: $\{a_i, b_i, c_i, d_i\}$. Рівняння, для визначення цих невідомих, можна отримати, виходячи з таких умов. По-перше, повинні задовольнятися співвідношення (V.1), тобто

$$S_i(x_{i-1}) = y_{i-1}, \quad S_i(x_i) = y_i,$$

а отже,

$$a_i = y_{i-1}, \quad (\text{VI.2})$$

$$a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = y_i, \quad (\text{VI.3})$$

де $h_i = x_i - x_{i-1}$ і мережа не обов'язково рівномірна, $i = \overline{1, n}$. Записавши вирази (VI.2) та (VI.3), маємо $2n$ рівнянь. По-друге, щоб сплайн був гладким, перші та другі похідні повинні бути неперервними для внутрішніх вузлів інтерполяції:

$$S'_i(x_i) = S'_{i+1}(x_i), \quad S''_i(x_i) = S''_{i+1}(x_i), \quad i = \overline{1, (n-1)},$$

тобто для $i = \overline{1, (n-1)}$

$$b_{i+1} = 2h_i c_i + 3h_i^2 d_i + b_i, \quad (\text{VI.4})$$

$$c_{i+1} = c_i + 3h_i d_i. \quad (\text{VI.5})$$

Вирази (VI.4) та (VI.5) — це ще $(2n - 2)$ рівняння. Дві умови (рівняння), яких не вистачає, визначаються значеннями першої та другої похідних у точках x_0 та x_n . Найчастіше використовують умову вільних кінців сплайну

$$\begin{aligned} S''_1(x_0) &= 0, \\ S''_n(x_n) &= 0, \end{aligned}$$

тобто

$$c_1 = 0, \quad (\text{VI.6a})$$

$$c_n + 3h_n b_n = 0. \quad (\text{VI.7a})$$

Але можуть задаватися й інші граничні умови. Наприклад, кути нахилу сплайну в крайніх точках:

$$\begin{aligned} S'_1(x_0) &= k_1, \\ S'_n(x_n) &= k_2, \end{aligned}$$

тоді

$$b_1 = k_1, \quad (\text{VI.6b})$$

$$b_n + 2h_n c_n + 3h_n^2 d_n = k_2. \quad (\text{VI.7b})$$

Інший варіант

$$\begin{aligned}S_1''(x_0) &= m_1, \\S_n''(x_n) &= m_2,\end{aligned}$$

і тоді

$$2c_1 = m_1, \quad (\text{VI.6в})$$

$$2c_n + 6h_n d_n = m_2, \quad (\text{VI.7в})$$

де k_1, k_2, m_1, m_2 — задані константи.

Систему рівнянь (VI.2)–(VI.7) можна розв'язувати, наприклад, методом Гаусса, але зручніше та економічніше звести її спочатку до тридіагональної і застосувати метод прогонки. Для цього із (VI.2) знаходимо коефіцієнти

$$a_i = y_{i-1}, \quad i = \overline{1, n}. \quad (\text{VI.8})$$

Із (VI.5) запишемо

$$d_i = \frac{c_{i+1} - c_i}{3h_i}, \quad i = \overline{1, n}, \quad (\text{VI.9})$$

і підставивши у (VI.3), отримаємо

$$b_i = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{3} (c_{i+1} + 2c_i), \quad (\text{VI.10})$$

після чого, підставляючи в (VI.4), маємо

$$\begin{aligned}h_{i-1} c_{i-1} + 2(h_{i-1} + h_i) c_i + h_i c_{i+1} &= \\= 3 \left(\frac{y_i - y_{i-1}}{h_i} - \frac{y_{i-1} - y_{i-2}}{h_{i-1}} \right), \quad i = \overline{2, n}.\end{aligned} \quad (\text{VI.11})$$

Доповнивши останню систему якоюсь парою умов із набору (VI.6)–(VI.7), отримаємо тридіагональну систему з $(n + 1)$ рівняння.

Таким чином, для обчислення інтерполяційної функції в точці x ($x \in [x_0, x_n]$) необхідно, спираючись на вузлові точки, за допомогою виразів (VI.8)–(VI.11) обчислити набір коефіцієнтів $\{a_i, b_i, c_i, d_i\}$, визначити на якому інтервалі знаходиться шукана точка x (відшукати номер k такий, щоб $x_k \leq x \leq x_{k+1}$) і тоді скористатися виразом (VI.1) із відомими значеннями a_k, b_k, c_k, d_k та x_{k+1} .

Базові сплайни

B -сплайном, або базовим сплайном порядку $(N - 1)$ відносно вузлів $\{x_i\}_{i=k}^{k+N}$ називається функція

$$B_{N-1,k}(x) = B_{N-1}(x_k, x_{k+1}, \dots, x_{k+N}, x) = N \cdot \sum_{i=k}^{k+N} \frac{(x_i - x)_{\max}^{N-1}}{\prod_{\substack{j=k \\ j \neq i}}^{k+N} (x_i - x_j)}, \quad (\text{VI.12})$$

де

$$(x_i - x)_{\max}^{N-1} = \begin{cases} (x_i - x)^{N-1}, & x \leq x_i; \\ 0, & x > x_i. \end{cases}$$

Якщо мережа рівномірна, тобто

$$x_{k+i} = x_k + i h,$$

де h — крок, то при $N = 4$ формулу для кубічного B -сплайну

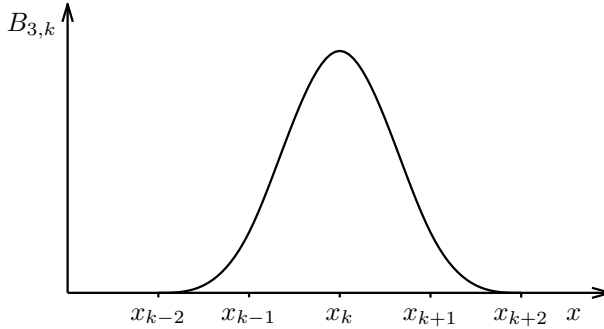


Рис. 11. Вигляд кубічного базового сплайну

з максимумом у точці $x = x_k$, можна записати у вигляді

$$\begin{aligned}
 B_{3,k}(x) = & \\
 & \begin{cases} 0, & x \leq x_{k-2}; \\ \frac{(x - x_{k-2})^3}{6h^4}, & x_{k-2} \leq x \leq x_{k-1}; \\ \frac{1}{6h} + \frac{x - x_{k-1}}{2h^2} + \frac{(x - x_{k-1})^2}{2h^3} - \frac{(x - x_{k-1})^3}{2h^4}, & x_{k-1} \leq x \leq x_k; \\ \frac{1}{6h} + \frac{x_{k+1} - x}{2h^2} + \frac{(x_{k+1} - x)^2}{2h^3} - \frac{(x_{k+1} - x)^3}{2h^4}, & x_k \leq x \leq x_{k+1}; \\ \frac{(x_{k+2} - x)^3}{6h^4}, & x_{k+1} \leq x \leq x_{k+2}; \\ 0, & x \geq x_{k+2}. \end{cases}
 \end{aligned}
 \tag{VI.13}$$

Тобто, якщо звичайні сплайни проводять лише на інтервалі між двома сусідніми точками, то базові — на відрізьку, який охоплює декілька точок із сукупності тих, по яких проводиться інтерполяція. У випадку кубічного базового сплайну він буде відмінним від нуля на інтервалі між п'ятьма послідовними точками, тобто шириною чотири кроки; причому на кожному із чотирьох підінтервалів (між сусідніми вузлами) він

описується своєю функцією. Вигляд функції (VI.13) наведено на рис. 11.

Іншою відмінністю базових сплайнів від звичайних є те, що значення інтерполяційної функції визначається не одним поліномом, а сумою декількох, а саме, лінійною комбінацією всіх B -сплайнів, проведених на сукупностях вузлів, які знаходяться в околі шуканої точки. Відповідно, для виконання умови (V.1) необхідно, щоб

$$\beta_{i-1} B_{i-1}(x_i) + \beta_i B_i(x_i) + \beta_{i+1} B_{i+1}(x_i) = y_i, \quad i = \overline{1, (n-1)}, \quad (\text{VI.14})$$

тут β_i — коефіцієнти інтерполяції, а індекс біля кубічного B -сплайну вказує на точку x_i , де він досягає свого максимального значення (порівняно з (VI.13) індекс “3” опущений задля скорочення запису). Тобто, необхідно підібрати такі значення коефіцієнтів інтерполяції, щоб значення інтерполяційної функції $f(x_i) = y_i$ збігалося із сумою значень трьох базових сплайнів, центрованих на точці, що є аргументом функції x_i , та двох сусідніх, домножених на ці коефіцієнти. Для крайніх точок діапазону розглядаються лише два доданки:

$$\begin{aligned} \beta_0 B_0(x_0) + \beta_1 B_1(x_0) &= y_0, \\ \beta_{n-1} B_{n-1}(x_n) + \beta_n B_n(x_n) &= y_n. \end{aligned} \quad (\text{VI.15})$$

Фактично, (VI.14) та (VI.15) — це тридіагональна система рівнянь, розв’язком якої є коефіцієнти $\{\beta_i\}$. Після цього значення інтерполяційної функції у довільній точці розраховується, як сума значень базових сплайнів у цій точці:

$$B(x) = \sum_{i=0}^n \beta_i B_i(x). \quad (\text{VI.16})$$

Кількість ненульових доданків у цій сумі для будь-якої точки не перевищуватиме чотири.

Завдання VI.1. Написати програму кубічної сплайн-інтерполяції для заданого набору рівновіддалених вузлів (використовувати кубічні базові сплайни).

Орієнтовні алгоритми

Завдання VI.1

У цій програмі доцільно створити окрему функцію, яка б обчислювала значення кубічного B -сплайну $B_i(x)$, центрованого у точці x_i відповідно до виразу (VI.13). Аргументами цієї функції можуть бути вказівник на набір вузлів, номер вузла i , відносно якого будуватиметься сплайн, та саме значення x .

- i. Зчитати дані з файлу у двовимірний масив та визначити кількість точок, по яких відбуватиметься інтерполяція, враховуючи, що вони нумеруються, починаючи з нуля.
- ii. Перевірити рівномірність отриманої мережі, порівнюючи різницю $(x_1 - x_0)$ з іншими різницями $(x_{i+1} - x_i)$ для $i = 1, (n - 1)$.
- iii. Визначити коефіцієнти інтерполяції $\{\beta_i\}$. Для цього знайти коефіцієнти тридіагональної матриці:

$$\begin{aligned} d_i &= y_i, & i &= \overline{0, n}; \\ a_i &= B_{i-1}(x_i), & i &= \overline{1, n}; \\ b_i &= B_i(x_i), & i &= \overline{0, n}; \\ c_i &= B_{i+1}(x_i), & i &= \overline{0, (n - 1)}; \\ a_0 &= 0, c_n = 0, \end{aligned}$$

де зміст величин a_i, b_i, c_i та d_i такий самий як і у формулі (IV.16) на с. 32.

- iv. Розв'язати отриману систему рівнянь відносно $\{\beta_i\}$ (див. алгоритм до *Завдання IV.1* на с. 37).

- v. Визначити мінімальне x_{\min} та максимальне x_{\max} значення аргументу.
- vi. Визначити крок інтерполяції $\delta x = (x_{\max} - x_{\min})/N$, де N — кількість точок, у яких буде розраховане значення інтерполяційної функції (рекомендовано вибирати $N \geq 100$).
- vii. Присвоїти цілій змінній k значення 0.
- viii. Відкрити файл для внесення результатів.
- ix. Обчислити абсцису x_k^{int} інтерполяційної функції
$$x_k^{\text{int}} = x_{\min} + \delta x \cdot k.$$
- x. Якщо $x_k^{\text{int}} > x_{\max}$, то перейти до п. xv.
- xi. Користуючись формулою (VI.16), обчислити ординату інтерполяційної функції y_k^{int} .
- xii. Записати значення $(x_k^{\text{int}}, y_k^{\text{int}})$ у файл.
- xiii. Збільшити значення $k \rightarrow k + 1$.
- xiv. Перейти до п. ix.
- xv. Закрити файл із результатами.

Практичне заняття VII. Апроксимація

При інтерполяції використовувалася умова (V.1) рівності значень інтерполяційної та вихідної функцій у вузлах інтерполяції. Проте у випадку, коли вихідні дані є результатом експерименту, слід ураховувати можливість наявності похибок (випадкових чи систематичних). Тому використовувати умову (V.1) не завжди доцільно.

Задачу апроксимації можна сформулювати трохи інакше: за набором значень $\{x_i, y_i\}$, $i = \overline{0, n}$, знайти залежність $y = \varphi(x)$, значення якої у точках x_i мало відрізняються від y_i . Фактично, йдеться про знаходження емпіричної формули.

Розв'язання цієї задачі складається із двох етапів:

- 1) підбір загального вигляду функції;
- 2) визначення оптимальних значень параметрів, які містить ця функція.

Перший етап можна виконати, спираючись на фізичні міркування (якщо відома залежність, якою мають описуватись отримані значення) або на геометричні — за виглядом отриманої кривої. Один із найпростіших варіантів — апроксимація набору значень лінійною функцією

$$\varphi = a_1 x + a_0, \quad (\text{VII.1})$$

де a_1 та a_0 — сталі, значення яких залежать від $\{x_i, y_i\}$. Зазначимо, що до лінійної залежності отримані експериментально дані можна звести, використовуючи, наприклад, не декартові координати. Зокрема, температурна залежність провідності σ власного напівпровідника описується формулою $\sigma = \sigma_0 \exp\left(\frac{E_g}{2kT}\right)$ (де E_g — ширина забороненої зони, k — стала Больцмана, T — температура, σ_0 — стала). Якщо експериментально виміряти залежність $\sigma(T)$, то лінійну апроксимацію отриманих даних можна застосувати у випадку, коли

по осі ординат відкладати натуральний логарифм провідності, а по осі абсцис — величину, обернену до температури:

$$\ln \sigma = \ln \sigma_0 + \frac{E_g}{2k} \frac{1}{T}.$$

У загальному випадку

$$y = \Phi(x, a_0, a_1, \dots, a_m), \quad (\text{VII.2})$$

де a_i — невідомі сталі параметри, а функція Φ далеко не обов'язково є поліноміальною. Кількість параметрів не має перевищувати кількість точок, по яких відбувається апроксимація; як правило, кількість параметрів набагато менша.

Другий етап задачі — підбір параметрів — виконується кількома методами.

Метод середніх

У цьому випадку параметри знаходять з умови, що середнє відхилення від точок, по яких проводять апроксимацію, має бути рівним нулеві:

$$\sum_{i=0}^n \varepsilon_i = \sum_{i=0}^n [\Phi(x_i, a_0, a_1, \dots, a_m) - y_i] = 0. \quad (\text{VII.3})$$

Звичайно, одного цього рівняння недостатньо для визначення $(m + 1)$ величин і тому різні ε_i групують у $(m + 1)$ рівнянь. Наприклад:

$$\left\{ \begin{array}{l} \varepsilon_0 + \varepsilon_1 + \varepsilon_2 = 0; \\ \varepsilon_3 + \varepsilon_4 + \varepsilon_5 + \varepsilon_6 = 0; \\ \dots\dots\dots \\ \varepsilon_{n-1} + \varepsilon_n = 0. \end{array} \right. \quad (\text{VII.4})$$

Метод найменших квадратів

У цьому методі записується сума квадратів відхилень

$$S = \sum_{i=0}^n \varepsilon_i^2 = \sum_{i=0}^n [\Phi(x_i, a_0, a_1, \dots, a_m) - y_i]^2 \quad (\text{VII.5})$$

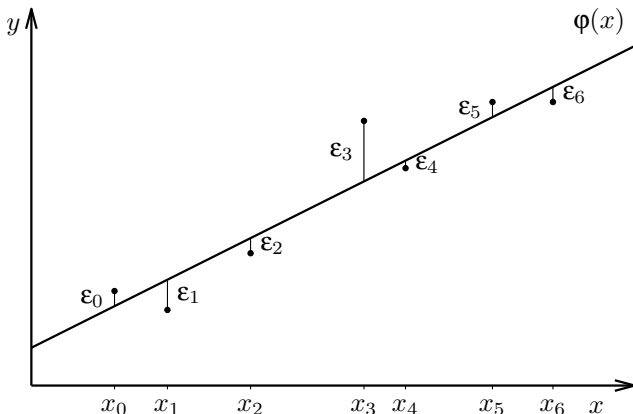


Рис. 12. Приклад лінійної апроксимації

і параметри знаходяться з умови мінімізації цієї суми:

$$\frac{\partial S}{\partial a_0} = 0, \quad \frac{\partial S}{\partial a_1} = 0, \dots, \quad \frac{\partial S}{\partial a_m} = 0. \quad (\text{VII.6})$$

Наприклад, при лінійній апроксимації (рис. 12) за формулою (VII.1) сума квадратів відхилень матиме вигляд

$$S = \sum_{i=0}^n [a_0 + a_1 x_i - y_i]^2 \quad (\text{VII.7})$$

і, прирівнявши до нуля частинні похідні виразу (VII.7) по параметрах a_0 та a_1 , можна отримати вирази для їх визначення:

$$a_0 = \frac{\sum_i x_i^2 \sum_i y_i - \sum_i x_i \sum_i x_i y_i}{n \sum_i x_i^2 - \left(\sum_i x_i \right)^2}, \quad (\text{VII.8})$$

$$a_1 = \frac{n \sum_i x_i y_i - \sum_i x_i \sum_i y_i}{n \sum_i x_i^2 - \left(\sum_i x_i \right)^2}.$$

У випадку апроксимації параболою

$$\varphi(x) = a_0 + a_1x + a_2x^2 \quad (\text{VII.9})$$

сума квадратів відхилень матиме вигляд

$$S = \sum_{i=0}^n [a_0 + a_1x_i + a_2x_i^2 - y_i]^2, \quad (\text{VII.10})$$

і після знаходження частинних похідних отримується система рівнянь

$$\begin{cases} na_0 + a_1 \sum_i x_i + a_2 \sum_i x_i^2 - \sum_i y_i = 0; \\ a_0 \sum_i x_i + a_1 \sum_i x_i^2 + a_2 \sum_i x_i^3 - \sum_i x_i y_i = 0; \\ a_0 \sum_i x_i^2 + a_1 \sum_i x_i^3 + a_2 \sum_i x_i^4 - \sum_i x_i^2 y_i = 0. \end{cases} \quad (\text{VII.11})$$

Її розв'язок дозволяє отримати явні вирази для параметрів:

$$a_2 = D^{-1} \cdot \left\{ \sum_i y_i x_i^2 \cdot \left[n \sum_i x_i^2 - \left(\sum_i x_i \right)^2 \right] - \right. \\ \left. - \sum_i x_i^3 \cdot \left[n \sum_i x_i y_i - \sum_i x_i \cdot \sum_i y_i \right] + \right. \\ \left. + \sum_i x_i^2 \cdot \left[\sum_i x_i y_i \cdot \sum_i x_i - \sum_i x_i^2 \cdot \sum_i y_i \right] \right\},$$

$$\begin{aligned}
a_1 = D^{-1} \cdot & \left\{ \sum_i x_i^4 \cdot \left[n \sum_i x_i y_i - \sum_i x_i \cdot \sum_i y_i \right] - \right. \\
& - \sum_i x_i^2 y_i \cdot \left[n \sum_i x_i^3 - \sum_i x_i \cdot \sum_i x_i^2 \right] + \\
& \left. + \sum_i x_i^2 \cdot \left[\sum_i x_i^3 \cdot \sum_i y_i - \sum_i x_i^2 \cdot \sum_i x_i y_i \right] \right\}, \\
a_0 = D^{-1} \cdot & \left\{ \sum_i x_i^4 \cdot \left[\sum_i y_i \cdot \sum_i x_i^2 - \sum_i x_i \cdot \sum_i x_i y_i \right] - \right. \\
& - \sum_i x_i^3 \cdot \left[\sum_i y_i \cdot \sum_i x_i^3 - \sum_i x_i y_i \cdot \sum_i x_i^2 \right] + \\
& \left. + \sum_i y_i x_i^2 \cdot \left[\sum_i x_i^3 \cdot \sum_i x_i - \left(\sum_i x_i^2 \right)^2 \right] \right\}, \quad (\text{VII.12})
\end{aligned}$$

де величина D описується виразом

$$\begin{aligned}
D = & \left\{ \sum_i x_i^4 \cdot \left[n \sum_i x_i^2 - \left(\sum_i x_i \right)^2 \right] - \sum_i x_i^3 \cdot \left[n \sum_i x_i^3 - \right. \right. \\
& \left. \left. - \sum_i x_i \cdot \sum_i x_i^2 \right] + \sum_i x_i^2 \cdot \left[\sum_i x_i^3 \cdot \sum_i x_i - \left(\sum_i x_i^2 \right)^2 \right] \right\}.
\end{aligned}$$

Існує також модифікований метод найменших квадратів, у якому враховується те, що різні точки, на основі яких проходить апроксимація, можуть мати різну важливість. У цьому випадку сума квадратів відхилень записується у вигляді

$$S = \sum_{i=0}^n \left\{ \omega_i [\varphi(x_i, a_0, a_1, \dots, a_m) - y_i]^2 \right\}, \quad (\text{VII.13})$$

де ω_i — вагові коефіцієнти, а дії для знаходження параметрів залишаються такими самими. Чим важливіша точка, тим більше значення вагового коефіцієнта для неї. Наприклад, якщо

помилка визначення функції у точці (x_i, y_i) дорівнює Δ_i , то можна прийняти, що $\omega_i = 1/\Delta_i$. У випадку, коли всі дані мають однакову відносну похибку, то вдалим вибором вагових коефіцієнтів є $\omega_i = 1/y_i$ — так звана статистична вага.

Завдання VII.1. Написати програму апроксимації поліномом першого порядку за методом найменших квадратів. За її допомогою обчислити коефіцієнти полінома для даних, що знаходяться у файлі.

Завдання VII.2. Написати програму апроксимації поліномом другого порядку за методом найменших квадратів. За її допомогою обчислити коефіцієнти полінома для даних, що знаходяться у файлі.

Орієнтовні алгоритми

Завдання VII.1

- i. Зчитати дані з файла у двовимірний масив та визначити кількість точок, по яких буде відбуватися апроксимація, ураховуючи, що вони нумеруються, починаючи з нуля.
- ii. Користуючись виразами (VII.8) (або (VII.12)), обчислити коефіцієнти апроксимуючого полінома. У цьому випадку доцільно спочатку розрахувати потрібні суми, а потім вже знайти параметри прямої (параболи).
- iii. Вивести на екран (або у файл) отримані результати.

Файл зі вхідними даними можна використати для побудови точок, а отримані параметри для побудови прямої (параболи), наприклад за допомогою програми *OriginPro*. Також у цій програмі можна безпосередньо виконати апроксимацію за допомогою вбудованих функцій і порівняти знайдені коефіцієнти.

Практичне заняття VIII. Числове інтегрування

Формули числового інтегрування функції однієї змінної називаються *квадратурними формулами*. Якщо функцію задано таблично (тобто набором $\{x_i, y_i = f(x_i)\}$, $i = \overline{0, n}$), то найбільш ефективними будуть квадратурні формули інтерполяційного типу. І навпаки, коли відомо аналітичний вигляд підінтегральної функції, але неможливо здійснити саме інтегрування, то найбільш ефективними будуть квадратурні формули типу Гаусса (у цьому випадку вибирає розташування вузлових точок той, хто проводить обчислення, і це дозволяє визначити їх таким чином, щоб мінімізувати кількість обчислень). Розглянемо ці методи детальніше.

Якщо необхідно знайти інтеграл

$$I = \int_a^b f(x) dx, \quad (\text{VIII.1})$$

то найбільш загальною постановкою задачі числового інтегрування є заміна виразу (VIII.1) сумою

$$I = \sum_{i=0}^n C_i f(x_i) + R \approx \sum_{i=0}^n C_i f(x_i), \quad (\text{VIII.2})$$

де C_i — вагові коефіцієнти, $f(x_i)$ — значення функції у вузлових точках x_i , R — похибка.

Один із варіантів розв'язання цієї задачі — замінити $f(x)$ інтерполяційним поліномом:

$$f(x) = \sum_{i=0}^n f(x_i) \chi_i(x) + r_n(x), \quad (\text{VIII.3})$$

де поліноми $\chi_i(x_i) = 1$ та $\chi_i(x_j) = 0$ при $j \neq i$, а $r_n(x)$ —

залишковий член інтерполяції (похибка заміни). Тоді

$$C_i = \int_a^b \chi_i(x) dx ;$$

$$R = \int_a^b r_n(x) dx .$$
(VIII.4)

Формули Ньютона—Котеса

Якщо для інтерполяції функції $f(x)$ використовуються поліноми Лагранжа, тобто

$$f(x) = \sum_{i=0}^n f(x_i) \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)},$$
(VIII.5)

то отримані в результаті квадратурні формули називаються формулами Ньютона—Котеса. Залежно від порядку використовуваних поліномів виділяють декілька методів.

Метод прямокутників. У цьому випадку на кожному відрізку, на які розбивається область інтегрування вузловими

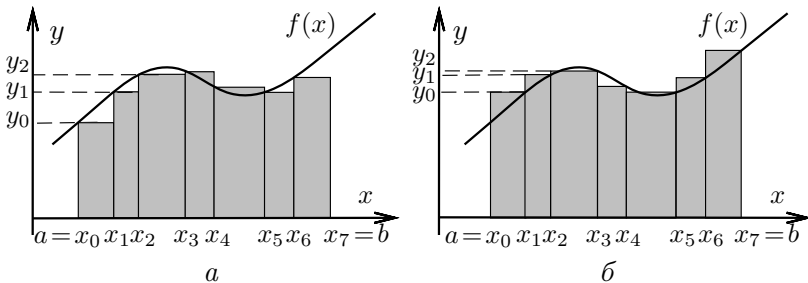


Рис. 13. Ілюстрація методу прямокутників із centruванням на лівій (а) і правій (б) границях відрізка (випадок нерівномірної мережі)

точками, функція $f(x)$ інтерполюється поліномом нульового степеня:

$$f(x_i) = y_i, \quad i = \overline{0, n}. \quad (\text{VIII.6})$$

Якщо позначити відстань між сусідніми точками $h_i = x_i - x_{i-1}$, $i = \overline{1, n}$, то інтеграл (VIII.1) можна обчислити, використовуючи центрування на лівій границі відрізка (рис. 13, а)

$$I \approx h_1 y_0 + h_2 y_1 + \dots + h_n y_{n-1} = \sum_{i=1}^n h_i y_{i-1}, \quad (\text{VIII.7a})$$

на правій (рис. 13, б)

$$I \approx h_1 y_1 + h_2 y_2 + \dots + h_n y_n = \sum_{i=1}^n h_i y_i \quad (\text{VIII.7б})$$

або у середніх точках

$$I \approx \sum_{i=1}^n h_i f(x_{i-1/2}), \quad (\text{VIII.7в})$$

де $x_{i-1/2} = \frac{1}{2}(x_{i-1} + x_i)$, $i = \overline{1, n}$.

Останній варіант найпоширеніший і дає найточніший результат серед усіх трьох. Якщо крок між вузловими точками однаковий ($h_i = h = \text{const}$) то (VIII.7в) перетворюється на

$$I \approx h \cdot \sum_{i=1}^n f(x_{i-1/2}). \quad (\text{VIII.7в}')$$

Найпростіший варіант застосування методу прямокутників для обчислення інтеграла — це використання значення функції лише в одній вузловій точці:

$$I = \int_a^b f(x) dx \approx (b - a) f\left(\frac{a + b}{2}\right). \quad (\text{VIII.8})$$

Метод трапецій. У цьому наближенні на відріжку $[x_k, x_{k+1}]$ функція $f(x)$ замінюється інтерполяційним поліномом Лагранжа першого порядку (рис. 14):

$$f(x) = f(x_k) + \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k} (x - x_k). \quad (\text{VIII.9})$$

Тоді

$$I \approx \frac{1}{2} \sum_{i=1}^n h_i (y_{i-1} + y_i). \quad (\text{VIII.10})$$

У випадку сталого кроку цей вираз перетворюється на

$$I \approx h \cdot \left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i \right). \quad (\text{VIII.10}')$$

У найпростішому випадку використовуються лише значення функції у точках на границях інтервалу, де шукається інтеграл:

$$I \approx \frac{b-a}{2} [f(a) + f(b)]. \quad (\text{VIII.11})$$

Метод парабол (або Симпсона)

Якщо інтервал $[a, b]$ розбити на парну кількість відрізків і на кожному з інтервалів $[x_0, x_2]$, $[x_2, x_4]$, \dots , $[x_{n-2}, x_n]$ інтерполювати функцію поліномом Лагранжа другого порядку, то у випадку рівномірної мережі для інтеграла отримаємо

$$I \approx \frac{h}{3} [y_0 + 4(y_1 + y_3 + \dots + y_{n-1}) + 2(y_2 + y_4 + \dots + y_{n-2}) + y_n]. \quad (\text{VIII.12})$$

Інший варіант — використовувати точки посередині відрізків (аргументи з напівцілими індексами); тоді кількість відрізків, на які розбивається інтервал інтегрування, може бути не лише парною:

$$I \approx \frac{h}{6} [y_0 + 4(y_{1/2} + y_{3/2} + \dots + y_{n-1/2}) + 2(y_1 + y_2 + \dots + y_{n-1}) + y_n]. \quad (\text{VIII.13})$$

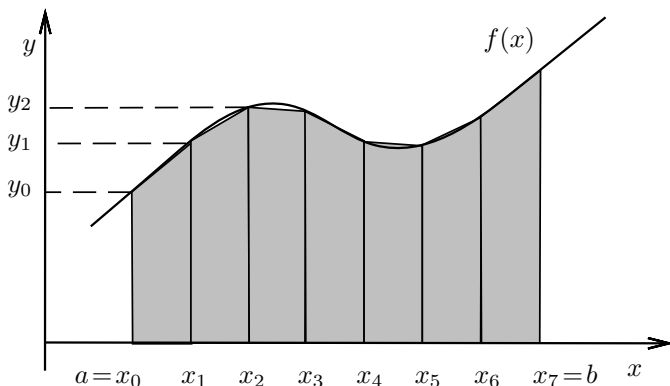


Рис. 14. Застосування методу трапецій до обчислення інтеграла (випадок рівномірної мережі)

Найпростіший випадок числового інтегрування методом Симпсона описується виразом

$$I \approx \frac{h}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right]. \quad (\text{VIII.14})$$

Квадратури Гаусса

Розв'язуючи задачу пошуку такого розташування вузлів x_i і такого значення вагових коефіцієнтів C_i , щоб $r_n(x) = 0$ (див. вирази (VIII.2)–(VIII.3)) для поліномів щонайбільшого степеня, К. Ф. Гаусс показав, що коли в ролі n вузлів використати нулі поліномів Лежандра $P_{n+1}(x)$ степеня $(n + 1)$, а вагові коефіцієнти обчислювати за виразом

$$C_i = \int_{-1}^1 \prod_{\substack{k=0 \\ k \neq i}}^n \frac{x - x_k}{x_i - x_k} dx,$$

то квадратурна формула

$$\int_{-1}^1 f(x) dx = \sum_{i=0}^n C_i f(x_i) \quad (\text{VIII.15})$$

буде точною для поліномів степеня $(2n + 1)$.

Іншими словами, порядок числового інтегрування з використанням квадратур Гаусса може бути таким.

1. Вибирати кількість вузлів (припустимо, їх буде $n + 1$).
2. Обчислити розташування вузлів $\{\tilde{x}_i\}$, $i = \overline{0, n}$, на відрізку $[-1, 1]$ як нулів полінома Лежандра степеня $n + 1$. Рекурентна формула для поліномів Лежандра має вигляд

$$\begin{aligned} P_{-1} &= 0, \\ P_0 &= 1, \\ (k + 1) P_{k+1}(x) &= (2k + 1) \cdot x \cdot P_k(x) - k \cdot P_{k-1}(x). \end{aligned} \quad (\text{VIII.16})$$

Для знаходження нулів можна використати метод Ньютона або будь-який інший метод розв'язання алгебраїчних рівнянь.

3. Обчислити вагові коефіцієнти за виразом

$$C_i = \frac{2}{(1 - \tilde{x}_i)^2 \cdot \left[\frac{d}{dx} P_{n+1}(\tilde{x}_i) \right]^2}. \quad (\text{VIII.17})$$

Загалом задача, сформульована у пунктах 2 та 3, є стандартною і тому можна просто використати розраховані таблиці вузлів та вагових коефіцієнтів для квадратур Гаусса (таблиця).

4. Урахувати, що інтервал інтегрування може відрізнятись від $[-1, 1]$, і тому остаточна формула матиме вигляд

$$\int_a^b f(x) dx = \frac{(b - a)}{2} \sum_{i=0}^n C_i f \left(\frac{a + b}{2} + \frac{b - a}{2} \cdot \tilde{x}_i \right). \quad (\text{VIII.18})$$

Таблиця. Вузли та вагові коефіцієнти для квадратур Гаусса

Кількість вузлів n	Положення вузлів $\tilde{x}_i \in [-1, 1]$	Вагові коефіцієнти C_i
2	$\pm 0,577350$	1,0
3	$\pm 0,774597$ 0,0	0,555556 0,888889
4	$\pm 0,861136$ $\pm 0,339981$	0,347855 0,652145
5	$\pm 0,906180$ $\pm 0,538469$ 0,0	0,236927 0,478629 0,568889
6	$\pm 0,932470$ $\pm 0,661209$ $\pm 0,238620$	0,171325 0,360762 0,467914

Завдання VIII.1. Написати процедуру пошуку інтеграла за допомогою квадратур Ньютона—Котеса, використовуючи

- а) метод прямокутників із різним положенням центра;
- б) метод трапецій;
- в) метод парабол (Симпсона).

Завдання VIII.2. Написати процедуру пошуку інтегралів за допомогою квадратур Гаусса з 3 та з 5 вузлами.

Орієнтовні алгоритми

Завдання VIII.1

- i. Зчитати дані з файлу у двовимірний масив та визначити кількість точок, по яких проводитиметься інтегрування.
- ii. Для методу парабол перевірити:
 - а) що точок більше двох;

- б) рівномірність мережі.
- iii. Обчислити значення інтеграла, користуючись
 - а) виразом (VIII.7a) для методу прямокутників із центруванням на лівій границі;
 - б) виразом (VIII.10) для методу трапецій;
 - в) виразом (VIII.12) для методу Симпсона.
- iv. Вивести на екран (або у файл) отримані результати.

Завдання VIII.2

- i. Створити функцію, яка обчислює значення підінтегрального виразу.
- ii. Використовуючи \tilde{x}_i та C_i із таблиці (третій та п'ятий рядки) за виразом (VIII.18) обчислити значення інтеграла.
- iii. Вивести на екран (або у файл) отримані результати.

Список літератури

1. *Форсайт, Дж.* Машинные методы математических вычислений / Дж. Форсайт, М. Малькольм, К. Моулер. — М. : Мир, 1980. — 280 с.
2. *Калиткин, Н. Н.* Численные методы / Н. Н. Калиткин. — М. : Наука, 1978. — 512 с.
3. *Турчак, Л. И.* Основы численных методов / Л. И. Турчак, П. В. Плотников. — М. : Физматлит, 2003. — 304 с.
4. *Буслов, В. А.* Введение в численный анализ / В. А. Буслов, С. Л. Яковлев. — СПб. : Изд-во Санкт-Петербургского гос. ун-та, 1999. — 99 с.
5. *Петров, И. Б.* Лекции по вычислительной математике / И. Б. Петров, А. И. Лобанов. — М. : Бином. Лаборатория знаний, 2006. — 529 с.
6. *Бахвалов, Н. С.* Численные методы / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельников. — М. : Физматлит, 2002. — 630 с.
7. *Каханер, Д.* Численные методы и программное обеспечение / Д. Каханер, К. Моулер, С. Нэш. — М. : Мир, 1998. — 575 с.
8. *Самарский, А. А.* Введение в численные методы / А. А. Самарский. — М. : Лань, 2009. — 288 с.; М. : Наука, 1997. — 234 с.; М. : Наука, 1987. — 269 с.
9. *Єжов, С. М.* Методи обчислень / С. М. Єжов. — К. : Київ. нац. ун-т імені Тараса Шевченка, 2001. — 174 с.

Навчальне видання

МОМОТ Андрій Іванович
ОЛІХ Олег Ярославович

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ

**Методичні вказівки
до практичних робіт**

Редактор *Л. В. Магда*

Оригінал-макет виготовлено Видавничо-поліграфічним центром "Київський університет"



Підписано до друку 06.09.11. Формат 60x84^{1/16}. Вид. № Фз7. Гарнітура Roman. Папір офсетний.
Друк офсетний. Наклад 100. Ум. друк. арк. 4,19. Обл.-вид. арк. 4,5. Зам. № 211-5770

Видавничо-поліграфічний центр "Київський університет"
01601, Київ, б-р Т. Шевченка, 14, кімн. 43

☎ (38044) 239 32 22; (38044) 239 31 72; тел./факс (38044) 239 31 28
Свідоцтво внесено до Державного реєстру ДК № 1103 від 31.10.02