

**МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ім. Богдана Хмельницького
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ТА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ**

Супруненко О.О.

ЧИСЕЛЬНІ МЕТОДИ В ІНФОРМАТИЦІ

Курс лекцій

*для студентів, які навчаються за напрямками підготовки
050101 „Комп’ютерні науки” та 050103 „Програмна інженерія”*

Черкаси 2009

УДК 519.6, 004.423
ББК 22.19

Рецензенти: Онищенко Б.О., кандидат фізико-математичних наук,
доцент кафедри математичного і програмного
забезпечення автоматизованих систем Черкаського
національного університету імені Богдана
Хмельницького;

Данченко О.Б., кандидат технічних наук, доцент кафедри
адміністрування бізнесу Університету економіки і
права «Крок».

Супруненко О.О.

Чисельні методи в інформатиці. Курс лекцій: для студентів, які
навчаються за напрямами підготовки 050101 „Комп’ютерні науки”,
050103 „Програмна інженерія”. – Черкаси: ЧНУ, 2009. – 130 с.

ISBN 978-966-353-143-4

У курсі лекцій з дисципліни „Чисельні методи в інформатиці” розглядаються особливості отримання чисельних розв’язків задач найбільш розповсюджених у моделюванні обчислювальних систем та систем керування технологічними процесами, оцінка похибок чисельного розв’язку, поняття стійкості та коректності постановки задачі.

У першій частині курсу лекцій розглядаються точні та наближені методи розв’язання лінійних рівнянь, ітераційні методи розв’язання систем нелінійних та трансцендентних рівнянь, методи інтерполяції поліномами і сплайнами, методи обробки експериментальних даних.

У другій частині курсу лекцій викладені методи чисельного диференціювання та інтегрування, одно- й багатокрокові методи розв’язання задачі Коші, сіткові методи розв’язання крайових задач для звичайних диференціальних рівнянь.

Кожна з тем має контрольні питання. У додатку наведена інформація по застосуванню внутрішньої мови програмування середовища автоматизації математичних розрахунків MatLab для створення програм, що реалізують розглянуті чисельні методи.

Матеріали курсу лекцій спрямовані на освоєння прикладних аспектів застосування чисельних методів в інженерній практиці. Дані лекції використовуються у навчальному процесі – при вивченні дисципліни “Чисельні методи в інформатиці”, що викладається автором для студентів бакалаврату напрямів підготовки “Програмна інженерія” та “Комп’ютерні науки” факультету інформаційних технологій та біомедичної кібернетики Черкаського національного університету імені Богдана Хмельницького.

Зміст

Передмова.....	5
----------------	---

Частина 1. Чисельні методи розв'язання систем лінійних та нелінійних рівнянь. Інтерполяція та наближення функцій..... 6

Розділ 1. Чисельні методи розв'язання задач. Похибки чисельного розв'язку....	6
1.1. Основні поняття.....	7
1.2. Поняття стійкості та коректності задачі.....	8
1.3. Похибки результату чисельного розв'язання задачі.....	10
Контрольні питання.....	13
Розділ 2. Розв'язання функціональних рівнянь з однією змінною	14
2.1. Етап 1: відокремлення коренів.....	16
2.2. Етап 2: уточнення коренів.....	18
2.2.1. Метод половинного ділення.....	18
2.2.2. Метод хорд.....	19
2.2.3. Метод січних.....	20
2.2.4. Метод дотичних (Ньютона).....	21
2.2.5. Метод простої ітерації.....	23
Контрольні питання.....	24
Розділ 3. Прямі та непрямі методи розв'язання систем лінійних алгебраїчних рівнянь. Методи Гауса та LU-розкладу	25
3.1. Основні поняття.....	25
3.2. Прямі методи розв'язання систем лінійних алгебраїчних рівнянь.....	26
3.2.1. Метод Гауса	26
3.2.2. Метод LU-розкладу	28
3.2.3. Зв'язок методу Гауса з методом LU - розкладу.....	29
3.3. Непрямі методи розв'язання систем лінійних алгебраїчних рівнянь	32
3.3.1. Забезпечення збіжності ітераційного процесу	32
3.3.2. Метод простої ітерації та метод Зейделя для розв'язання систем лінійних алгебраїчних рівнянь.....	33
3.3.3. Метод релаксації для розв'язання систем лінійних алгебраїчних рівнянь	34
Контрольні питання.....	35
Розділ 4. Розв'язання систем нелінійних рівнянь. Метод Ньютона	36
Контрольні питання.....	40
Розділ 5. Інтерполяція функцій. Інтерполяційні поліноми Лагранжа. Сплайн-інтерполяція.....	41
5.1. Кусково-лінійна інтерполяція. Інтерполяційні поліноми вищих порядків. Інтерполяційний поліном Лагранжа	43
5.2. Сплайн-інтерполяція	46
Контрольні питання.....	51
Розділ 6. Апроксимація експериментальних даних. Метод найменших квадратів. Інтерполяція функцій за допомогою ортогональних поліномів.....	52
6.1. Апроксимація експериментальних даних. Метод найменших квадратів.....	52
6.2. Інтерполяція функцій ортогональними поліномами.....	54

6.3. Базисні сплайни (В-сплайни).....	57
Контрольні питання.....	61
Частина 2. Чисельне диференціювання та інтегрування. Розв'язання диференційних рівнянь.....	62
Розділ 7. Чисельне диференціювання та чисельне інтегрування.....	62
7.1. Чисельне диференціювання.....	62
7.1.1. Використання скінченних різниць.....	62
7.1.2. Чисельне диференціювання на основі інтерполяційної формули Лагранжа.....	65
7.1.3. Застосування похідних.....	66
7.2. Чисельне інтегрування.....	67
7.2.1. Квадратурні формули інтерполяційного типу.....	69
7.2.2. Замкнуті формули квадратури Ньютона-Котеса.....	70
Контрольні питання.....	71
Розділ 8. Чисельні методи розв'язання задачі Коші для звичайних диференційних рівнянь.....	73
8.1. Основні поняття.....	73
8.2. Явний метод Ейлера.....	76
8.3. Неявний метод Ейлера.....	78
8.4. Модифікації методу Ейлера.....	80
8.5. Метод Гюна.....	83
8.6. Метод рядів Тейлора.....	84
8.7. Методи Рунге-Кутта.....	85
Контрольні питання.....	88
Розділ 9. Багатокрокові методи розв'язання диференційних рівнянь в частинних похідних.....	90
9.1. Метод Адамса-Бешфорса-Маултона.....	90
9.2. Метод Мілна-Сімпсона.....	93
9.3. Метод Хеммінга.....	95
9.4. Обмеження при застосуванні методів прогнозу-корекції.....	96
Контрольні питання.....	98
Розділ 10. Крайові задачі для звичайних диференційних рівнянь. Методи сіток.....	99
10.1. Основні поняття.....	99
10.2. Методи сіток.....	101
10.2.1. Метод кінцевих різниць.....	102
10.2.2. Метод кінцевих елементів.....	109
10.2.3. Алгоритм методу кінцевих елементів.....	112
Контрольні питання.....	119
Додаток А. Короткі теоретичні відомості по програмуванню в інтегрованій системі автоматизації математичних розрахунків MatLab.....	120
Додаток Б. Пакети прикладних програм інтегрованої системи автоматизації математичних розрахунків MatLab.....	126
Список літературних джерел.....	130

Передмова

Підготовка інженерних кадрів передбачає набуття практичних навичок у вирішенні прикладних задач, які пов'язані з конкретної областю техніки відповідно спеціальності. При підготовці інженерів-програмістів основна увага приділяється вивченню теоретичного матеріалу та його реалізації у практичних задачах при моделюванні та створенні програмних продуктів для автоматизації різноманітних технологічних процесів, при створенні інформаційно-аналітичних комплексів для управління організаційними структурами.

Підпорядковуючись основним задачам навчання програмістів, даний лекційний курс передбачає вивчення чисельних методів, які використовуються при створенні моделей динамічних систем, обробці результатів експериментальних даних, управлінні обчислювальними процесами. Курс лекцій вміщує чисельні методи, які необхідно знати і вміти застосовувати кожному інженеру та науковцю-досліднику, що є фахівцем в області інформаційних технологій.

Знайомлячись із матеріалом даної дисципліни та вивчаючи особливості конкретних чисельних методів студент не може повною мірою зрозуміти особливості функціонування алгоритмів методів та накопичення похибок результатів без самостійної програмної реалізації даних алгоритмів. Тому реалізацію чисельних методів у даному курсі пропонується здійснювати в системі Matlab (корпорації The MathWorks, Inc. (США)), що має внутрішню мову програмування та найбільшу бібліотеку чисельних методів NAG Foundation Toolbox, яка створена групою The Numerical Algorithms Group Ltd. і на даний момент нараховується більше 400 функцій у вигляді *m*-файлів.

Система автоматизації математичних розрахунків Matlab дозволяє використовувати сотні пакетів прикладних програм, найвідоміші з яких Symbolic Math Toolbox, Spline Toolbox, Statistics Toolbox, Optimization Toolbox, Fuzzy Logic Toolbox, Neural Networks Toolbox, Partial Differential Equations Toolbox та ін. Система Matlab має підсистему імітаційного моделювання Simulink for Windows, яка дозволяє виконувати блочне, імітаційне та ситуаційне моделювання різних систем та пристроїв, має спеціальний інструментарій для візуалізації результатів розрахунків.

Даний курс має за мету вивчення основних чисельних методів, що застосовуються в галузі інформаційних технологій, та їх практичну реалізацію і перевірку якості отримання результатів, що є важливим результатом роботи програміста.

Частина 1. Чисельні методи розв'язання систем лінійних та нелінійних рівнянь. Інтерполяція та наближення функцій.

Розділ 1. Чисельні методи розв'язання задач. Похибки чисельного розв'язку

У сучасних математичних дослідженнях надзвичайно важливим завданням є отримання числових розв'язків задач, які використовуються в моделях реальних фізичних об'єктів та систем.

В більшості галузей сучасної науки і техніки часто зустрічаються математичні задачі, які не можливо точно розв'язати класичними методами або такий розв'язок отримується у складному вигляді, абсолютно не прийнятному для практичного використання. Наприклад, досить часто необхідно розв'язувати системи лінійних алгебраїчних рівнянь з десятками і сотнями невідомих. Розповсюдженим є і знаходження коренів алгебраїчних рівнянь високого порядку та коренів трансцендентних рівнянь. Нерідко приходить мати справу з розв'язанням систем диференціальних рівнянь, які не інтегруються в елементарних функціях.

Кількість таких задач значно зросла у зв'язку з бурхливим розвитком науки і техніки з одного боку, і можливістю застосуванням розроблених теорій в машинних моделях, які формуються та функціонують у комп'ютерному варіанті.

Комп'ютерну техніку використовують не тільки для побудови математичних моделей, а і для ефективного розв'язання задач, що пов'язані з функціонуванням даних моделей. У зв'язку з цими потребами виділилась область математики, яка покликана розробляти методи отримання числових результатів основних задач математичного аналізу, алгебри та геометрії, а також шляхи використання з цією метою сучасних комп'ютерних засобів. Ця область математики отримала назву *чисельні методи*.

З початку застосування комп'ютерної техніки швидкість розрахунків зросла з $0,1 = 10^{-1}$ операцій в секунду при ручному розрахунку до $10^9 \div 10^{11}$ операцій в секунду на сучасній комп'ютерній техніці [1].

У зв'язку з бурхливим розвитком комп'ютерної техніки постали задачі математизації різних розділів науки: хімії, біології, економіки, геології, географії, психології, медицини, численних галузей техніки.

Процес математизації полягає у побудові математичних моделей процесів і явищ та в розробці методів їх дослідження.

Застосування комп'ютерної техніки та розширення технічної і математичної освіти різко збільшило можливості побудови й дослідження математичних моделей. Все частіше результати розрахунків дозволяють виявити і передбачити явища, що раніше не спостерігалися, що можливо завдяки виникненню і розвитку машинного (комп'ютерного) експерименту.

Вимоги до чисельних розв'язків нових задач привели до появи великої кількості нових чисельних методів, систематизації та переосмислення давніх методів. Застосування комп'ютерної техніки і технологій привело до:

- збільшення швидкості отримання розв'язків, розширення пам'яті, вдосконалення структури та технічних засобів обчислювальної техніки;
- розробки програмних засобів спілкування є комп'ютером (операційні системи, мови програмування, бібліотеки і пакети прикладних програм);
- поглиблення розуміння процесів і явищ науки й техніки, природи і суспільства, створення їх математичних моделей;
- вдосконалення методів розв'язання традиційних математичних і прикладних задач, створення методів розв'язання нових задач;
- росту розуміння можливостей застосування комп'ютерної техніки, розширення комп'ютерної грамотності.

Ефект, що досягається вдосконаленням чисельних методів, за порядком дорівнює ефекту, що досягається за рахунок ефективності комп'ютерної техніки. В наш час стрімко розвиваються багатопроцесорні та багатомашинні обчислювальні системи, що обумовлює необхідність розпаралелювання задач, в тому числі й задач, що реалізують чисельні методи. Тому перспективними є дослідження в області чисельних методів, що дозволять проводити ефективно розбиття задачі на підзадачі та будувати оцінки їх ефективності по часу та іншим критеріям.

1.1. Основні поняття

Розв'язати задачу в класичній математиці означає довести існування розв'язку і знайти його точне значення.

При **чисельному розв'язку задачі** потрібно також довести існування розв'язку і запропонувати *ітераційний процес*, який збігається до розв'язку зі заданою точністю.

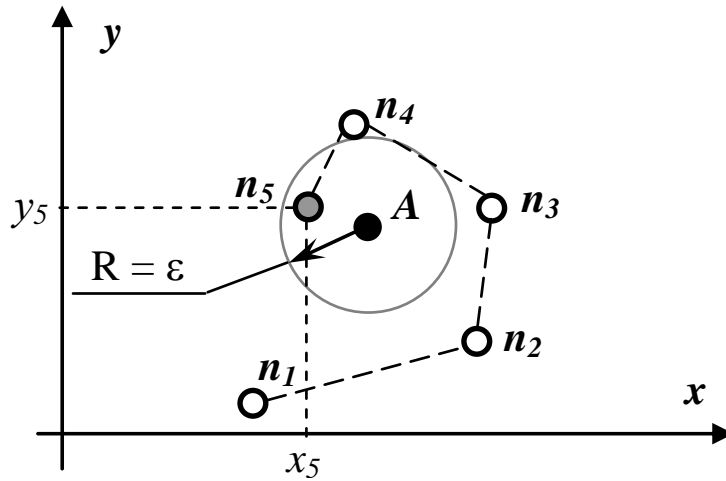


Рис. 1.1. Точність розв'язання задачі чисельними методами визначається ε -околом.

Поняття заданої точності в чисельних методах продемонструємо графічно. Навколо точки A , що відображає точний розв'язок задачі, окреслимо коло радіусом ε , тобто ε -оکیل. При обчисленні чергового значення змінних x_5 та y_5 , що визначають положення точки розв'язку n_5 , яка першою з проміжних результатів обчислень n_i входить в ε -оکیل, отримуємо результат чисельного розрахунку задачі з точністю ε .

Для прикладного математика і програміста, що працює у певній галузі науки чи техніки, важливо, щоб процес розв'язку не потребував великих витрат часу і комп'ютерної пам'яті. Для нього важливі також питання, пов'язані зі стійкістю результатів відносно збурень початкових даних і округленнями при обчисленнях.

1.2. Поняття стійкості та коректності задачі

Похибки у вхідних даних задачі є неусувними. Причому в одних випадках похибки вхідних даних і результатів задачі мають однаковий порядок, а в інших вони можуть відрізнитися на кілька порядків. Чутливість задачі до неточностей у вхідних даних характеризується *стійкістю*.

Задача називається **стійкою за вхідними даними**, якщо її розв'язок неперервно залежить від вхідних даних, тобто малому приросту вхідної величини Δx відповідає малий приріст розв'язку Δy .

При невиконання даної умови задача вважається не стійкою, що означає – зміни, навіть незначні, у вхідних даних можуть привести до як завгодно великих похибок розв'язку. Застосування до розв'язання таких задач чисельних методів є недоцільним, оскільки поточні похибки будуть накопичуватися при виконанні ітерацій і не дозволять отримати розв'язок із заданою (чи обмеженою) точністю.

Нестійким також може бути чисельний метод розв'язання задачі. Для усунення такої нестійкості вносять зміни до алгоритму методу, чи пред'являють жорсткі вимоги до обчислювальної похибки (похибки округлення). Наприклад, обчислимо інтеграл:

$$I_n = \int_0^1 x^n e^{x-1} dx; \quad n = 1, 2, \dots, 9.$$

При інтегруванні за частинами отримаємо:

$$\int_0^1 x^n e^{x-1} dx = x^n e^{x-1} \Big|_0^1 - \int_0^1 n x^{n-1} e^{x-1} dx = 1 - n \int_0^1 x^{n-1} e^{x-1} dx \Rightarrow$$

$$\Rightarrow \quad I_1 = \frac{1}{e}; \quad I_2 = 1 - 2 \cdot I_1; \quad \dots \quad ; \quad I_n = 1 - n \cdot I_{n-1}$$

Обчислимо значення інтегралу для $n = 1, 2, \dots, 9$:

$$\begin{array}{lll} I_1 = 0,367879; & I_4 = 0,170904; & I_7 = 0,110160; \\ I_2 = 0,264242; & I_5 = 0,145480; & I_8 = 0,108720; \\ I_3 = 0,207274; & I_6 = 0,127120; & I_9 = -0,068480. \end{array}$$

Значення I_9 є помилковим, оскільки підінтегральна функція $x^9 e^{x-1}$ у всіх точках відрізка $[0;1]$ є невід'ємною. Дана помилка викликана похибкою округлення: для I_1 $\Delta I_1 = 4,4 \cdot 10^{-7}$; ... ; для I_9 $\Delta I_9 = 9! \cdot \Delta I_1 = 0,1601$. Істинне значення I_9 з трьома значущими цифрами після коми дорівнює 0,0916.

Ще одним важливим поняттям є коректність постановки задачі.

Задача називається коректно поставленою, якщо для будь-яких вхідних даних з деякого класу існує єдиний і стійкий за вхідними даними розв'язок.

Для розв'язку некоректно поставлених задач застосовувати звичайні чисельні методи не варто, оскільки похибки округлень при розрахунках можуть катастрофічно зростати, спотворюючи

результати розв'язання задачі. Для розв'язання таких задач використовують так звані *методи регуляризації* [2], які замінюють дану задачу коректно поставленою.

1.3. Похибки результату чисельного розв'язання задачі

Похибки результату чисельного розв'язання задачі поділяються на три групи [1, 3], розподіл по яких обумовлений наступними причинами:

- математичний опис задачі є неточним (неточно задані початкові дані, не всі характеристики об'єкта моделювання відображені в моделі, та ін.) – з цієї причини виникає *незнищенна похибка*,
- застосований для розв'язання задачі метод не є точним – виникає *похибка методу*,
- при введенні даних, виконанні арифметичних операцій, при виведенні результатів проводяться округлення, що викликає *обчислювальну похибку*.

Наприклад, розглянемо модель математичного маятника (рис. 2), який починає рух в момент часу $t = t_0$. Знайти кут відхилення φ від вертикального положення в момент часу t_1 .

Диференціальне рівняння, що описує коливання маятника записується у вигляді:

$$l \frac{d^2 \varphi}{dt^2} + \mu \frac{d\varphi}{dt} - g \cdot \sin \varphi = 0 \quad (1.1)$$

де l – довжина маятника; μ – коефіцієнт тертя; g – прискорення сили тяжіння. В описі задачі є незнищенна похибка – реальне тертя залежить від швидкості і не є лінійним (лише наближається до лінійного); визначення параметрів $l, \mu, g, t_0, \varphi(t_0), \varphi'(t_0)$ не є точним.

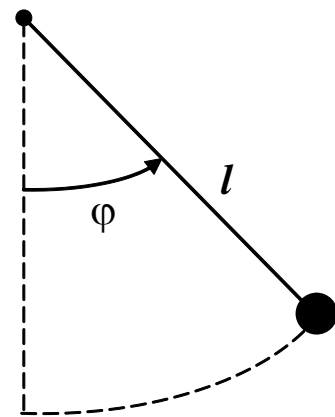


Рис. 1.2. Математичний маятник.

Незнищенна похибка $\rho_1 = \bar{I} - I$, де I – точне значення невідомої.

Диференційне рівняння (1.1) не розв'язується в явному вигляді, тому для розв'язку застосовуємо чисельний метод, що має похибку методу:

$$\rho_2 = \tilde{I}_k - \bar{I}$$

Обчислення проводяться з кінцевою кількістю розрядів чисел, що викликає обчислювальну похибку:

$$\rho_3 = I^* - \tilde{I}_k$$

Повну похибку розв'язку запишемо у вигляді:

$$\rho = I^* - I = \rho_1 + \rho_2 + \rho_3.$$

Абсолютна і відносна похибки. Якщо a – точне значення деякої величини, й a^* – відоме наближення до нього, то абсолютна похибка наближеного значення a^* є $\Delta(a^*) \geq |a^* - a|$. Відносна похибка наближеного значення розраховується як $\delta(a^*) \geq \left| \frac{a^* - a}{a} \right|$.

Відносну похибку часто виражають в процентах. Наведемо приклади з розрахунку абсолютної і відносної похибок:

$$\begin{array}{l} x = 3,141592 \\ 1) \quad x^* = 3,14 \end{array} \quad \begin{array}{l} \Delta(x^*) = |3,141592 - 3,14| = 0,001592 \\ \delta(x^*) = \left| \frac{3,141592 - 3,14}{3,141592} \right| = 0,00507 \end{array}$$

$$\begin{array}{l} y = 1\,000\,000 \\ 2) \quad y^* = 999\,996 \end{array} \quad \begin{array}{l} \Delta(y^*) = |1\,000\,000 - 999\,996| = 4 \\ \delta(y^*) = \left| \frac{1\,000\,000 - 999\,996}{1\,000\,000} \right| = 0,000004 \end{array}$$

$$\begin{array}{l} z = 0,000012 \\ 3) \quad z^* = 0,000009 \end{array} \quad \begin{array}{l} \Delta(z^*) = |0,0000122 - 0,000009| = 0,000003 \\ \delta(z^*) = \left| \frac{0,000012 - 0,000009}{0,000012} \right| = 0,25 \end{array}$$

У першому прикладі різниця між значеннями абсолютної і відносної похибки невелика, тому можемо використовувати будь-яку з них.

У другому та третьому прикладах, в яких значення параметру значно відрізняється від *оддиниці* в більший чи менший бік, відносна похибка δ є кращим індикатором точності наближень, аніж абсолютна похибка Δ . Тому в таких випадках для оцінки точності наближень значень рекомендується використовувати відносну похибку [4].

Число p^* є наближенням p з d знаками (цифрами) після коми, якщо d є найбільшим позитивним цілим числом, для якого виконується умова [5]:

$$\left| \frac{p - p^*}{p} \right| < \frac{10^{-d}}{2}.$$

Визначимо число значущих цифр для наближень у прикладах 1÷3:

$$x = 3,141592 \quad \delta(x^*) = \left| \frac{3,141592 - 3,14}{3,141592} \right| = 0,00507$$

1)

$$x^* = 3,14 \quad \delta(x^*) = 0,00507 < \frac{10^{-2}}{2}$$

Наближення x^* до точного значення x виконано з двома значущими цифрами: $d = 2$.

$$y = 1\,000\,000 \quad \delta(y^*) = \left| \frac{1\,000\,000 - 999\,996}{1\,000\,000} \right| = 0,000004$$

2)

$$y^* = 999\,996 \quad \delta(y^*) = 0,000004 < \frac{10^{-5}}{2}$$

Наближення y^* до точного значення y виконано з п'ятьма значущими цифрами: $d = 5$.

$$z = 0,000012 \quad \delta(z^*) = \left| \frac{0,000012 - 0,000009}{0,000012} \right| = 0,25$$

3)

$$z^* = 0,000009 \quad \delta(z^*) = 0,25 < \frac{10^0}{2}$$

Наближення z^* до точного значення z виконано без значущих цифр: $d = 0$.

Контрольні питання

1. Яка причина виділення окремої області математики – чисельних методів?
2. В чому полягає відмінність розв'язання задач класичними методами математики та чисельними методами?
3. Які чинники пов'язують розвиток чисельних методів та комп'ютерної техніки й технологій?
4. Що значить розв'язати задачу з точністю ε ?
5. Яка задача називається стійкою за вхідними даними?
6. Яка задача є коректно поставленою?
7. На які групи поділяються похибки чисельних розв'язків і які причини викликають їх виникнення?
8. В яких випадках і чому оцінка похибки проводиться за абсолютною похибкою?
9. В яких випадках оцінка похибки проводиться за відносною похибкою?
10. Що значить знайти розв'язок задачі з n значущими цифрами?

Розділ 2. Розв'язання функціональних рівнянь з однією змінною

Важливу задачу прикладного аналізу представляє розв'язок функціональних рівнянь з однією змінною, які застосовуються при описі математичних моделей у фізиці, механіці, економіці та інших науках та прикладних галузях.

У загальному вигляді функціональне рівняння записується у вигляді:

$$F(x) = 0 \quad (2.1)$$

де $F(x)$ визначена і неперервна на скінченному чи нескінченному інтервалі $[a, b]$.

Число ξ називається коренем r кратності, якщо воно задовольняє рівнянням:

$$F(\xi) = 0, F'(\xi) = 0, \dots F^{r-1}(\xi) = 0 \quad (2.2)$$

Однократний корінь називається простим.

Два рівняння $F(x)$ та $G(x)$ називаються рівносильними, якщо будь-який розв'язок кожного з них є розв'язком і для другого.

Функціональні рівняння поділяються на алгебраїчні, якщо функція є алгебраїчною (наприклад: $x^3 - 3x + 10 = 0$), та трансцендентні, якщо функція вміщує логарифмічні, тригонометричні та інші неалгебраїчні функції (наприклад: $\sin(2 \cdot x) - \lg(x/3) = 0$). Алгебраїчні рівняння поділяються на лінійні й нелінійні. У лінійних рівняннях змінна має лише перший ступінь, а у нелінійних – змінна представлена у першому, другому чи вищих ступенях.

Шляхом алгебраїчних перетворень із будь-якого алгебраїчного рівняння можна отримати рівняння в канонічній формі:

$$P_n(x) = a_0 x^n + a_1 x^{n-1} + \dots + a_n = 0 \quad (2.3)$$

де n – ступінь алгебраїчного рівняння.

Кожне алгебраїчне рівняння має хоча б один дійсний корінь чи пару комплексних.

При зведенні алгебраїчного рівняння до канонічної форми отримаємо ті ж корені, що і для початкового рівняння, але при цьому можуть з'явитися сторонні корені. Наприклад:

$$\sqrt{2x^2 - 1} + x = \sqrt{2x^2 + 1} - 1 \Rightarrow 7x^4 + 12x^3 + 2x^2 - 4x - 5 \quad (2.4)$$

Якщо функція $F(x)$ не є алгебраїчною, то рівняння (2.1) називається трансцендентним. Наприклад:

$$\lg(2x) = \cos(0.5 \cdot x) \quad (2.5)$$

В деяких випадках розв'язування трансцендентних рівнянь зводиться до розв'язання алгебраїчних.

Більшість функціональних рівнянь з однією змінною не розв'язуються точними методами шляхом аналітичних перетворень. До того ж, більшість функціональних рівнянь є математичними моделями певних об'єктів чи явищ, вказати точні параметри (коефіцієнти) яких неможливо. Тому на практиці для їх розв'язання застосовують чисельні методи, більшість з яких є наближеними і дозволяє знайти розв'язок з певною наперед заданою точністю.

У теорії чисельних методів **розв'язати рівняння** – значить встановити, чи має воно корені на вказаному проміжку, з'ясувати кількість коренів, відшукати значення всіх виявлених коренів із заданою точністю, тобто корені мають знаходитися у ε -околі точного розв'язку.

Задача чисельного знаходження дійсних і комплексних коренів рівняння зазвичай складається з *двох етапів*:

- **відокремлення коренів** – знаходження достатньо малих околів у заданій області, в кожному з яких знаходиться тільки один корінь,
- **уточнення коренів** – обчислення коренів із заданим ступенем точності ε в деякому околі (ε -околі, п. 1.1.).

Для знаходження дійсних коренів рівняння (2.1) застосовується ряд методів, серед яких частіше використовують такі чисельні методи:

- метод половинного ділення (бісекції);
- метод хорд;
- метод січних;
- метод дотичних (Ньютона);
- комбінований метод;
- метод Рібакова, знаходження всіх дійсних коренів;
- метод Ейткена-Стеффенсона;
- метод зворотної квадратичної інтерполяції-екстраполяції;
- метод порозрядного наближення;
- метод простої ітерації.

При застосуванні вищенаведених методів важливо забезпечити збіжність їх до розв'язку. По швидкості збіжності до розв'язку розрізняють:

- методи поступового обмеження інтервалу пошуку кореня (методи половинного ділення), ці методи повільно сходяться до розв'язку;
- методи апроксимації функції, які передбачають заміну початкової функції більш простою, та проведення обчислень з використанням спрощеного опису функції (метод хорд, метод Ньютона), дані методи швидше сходяться до розв'язку.

Швидкість збіжності метода характеризує відношення e_{k+1}/e_k , яке менше одиниці. У відношенні $e_k = |x_k - x|$, де x – точне значення розв'язку, x_k – чергове наближення до кореня. Чим менша величина даного відношення, тим швидше метод збігається до розв'язку.

2.1. Етап 1: відокремлення коренів

Відокремлення коренів полягає у встановленні „тісних” проміжків, кожен з яких вміщує тільки один корінь. Найбільш наочно про демонструвати цей етап можливо графічно. Враховуючи, що дійсні корені рівняння (2.1) – це точки перетину графіка функції $F(x)$ з віссю абсцис, достатньо побудувати графік $F(x)$ (рис.2.1) і відмітити на осі Ox відрізки, кожен з яких вміщує лише один корінь.

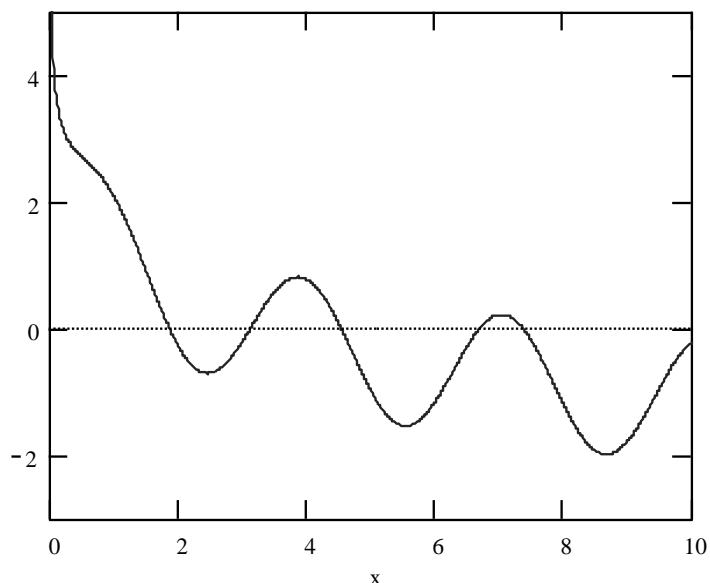


Рис. 2.1. Графічний спосіб відокремлення коренів.

Побудову графіка вдається значно спростити, якщо замінити рівняння (2.1) рівносильним йому рівнянням $F_1(x) = F_2(x)$. В цьому випадку будують графіки функцій $F_1(x)$ та $F_2(x)$, потім на вісі Ox

відмічають відрізки, що вміщують абсциси точок перетину цих графіків (рис. 2.2.).

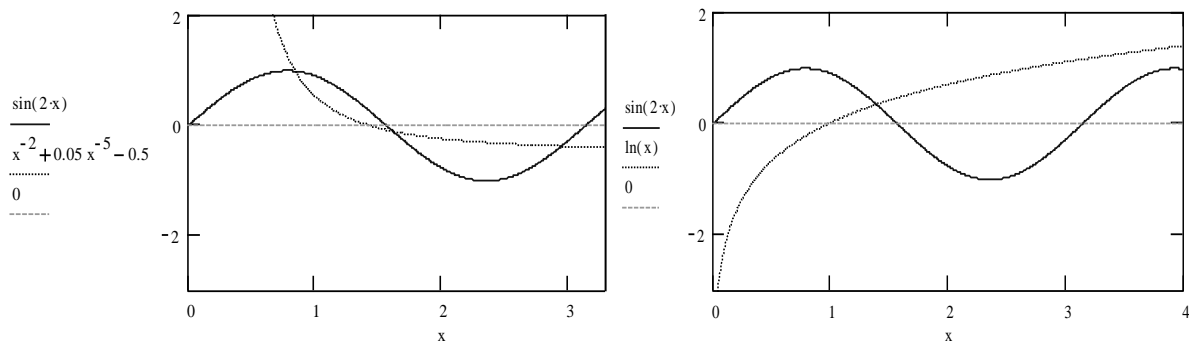


Рис. 2.2. Варіанти розв'язання рівняння при заміні функції рівносильними функціями $F_1(x)$ та $F_2(x)$ на визначеному проміжку.

Зустрічаються випадки, коли не тільки на першому етапі – при виділенні проміжків, що вміщують корені рівняння, а і при подальшому розв'язанні рівняння розрахунки проводять не за основною функцією $F(x)$, а за наближеною до неї функцією $F_k(x)$ (рис. 2.3).

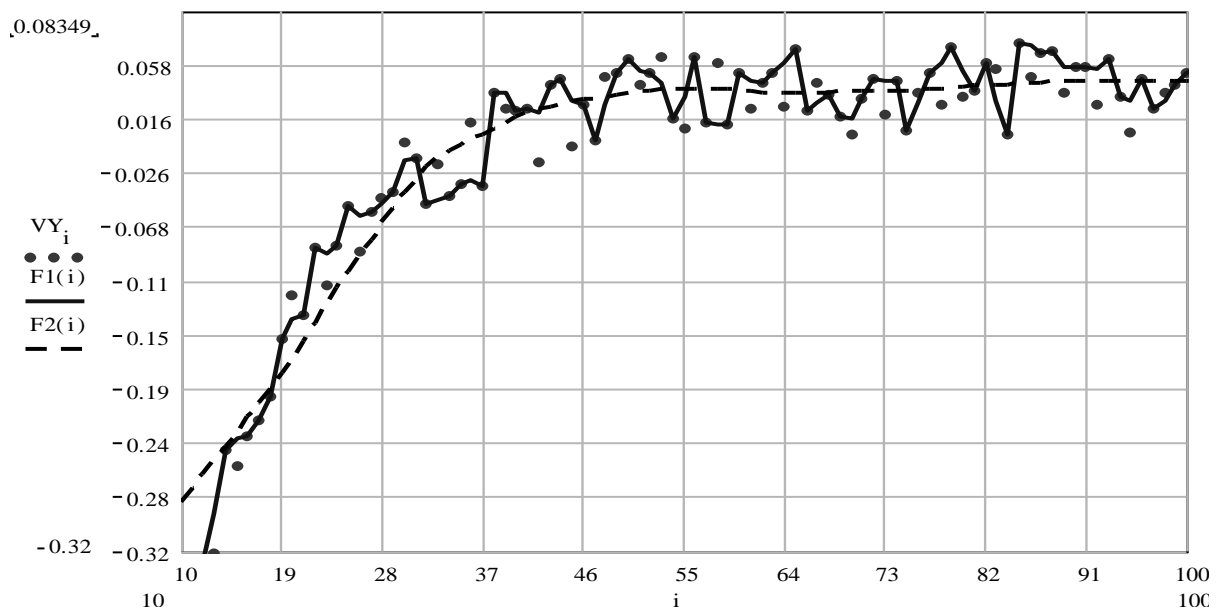


Рис.2.3. Реальна функція $F1(i)$ та її наближений аналог-функція $F2(i)$.

При написанні програми, яка виконує відокремлення коренів графічний спосіб не є зручним, тому застосовують найпростіший алгоритм відокремлення коренів:

проміжок, на якому відшуковуються розв'язки, ділять рівномірно на невеликі відрізки, на кінцях кожного відрізка розраховують значення функції, якщо функція на кінцях відрізка змінює знак – відрізок містить корінь (значення початку і кінця відрізка записується для подальшої обробки), інакше – кореня на відріжку немає і відрізок ігнорується.

Величина відрізків поділу залежить від коефіцієнта гладкості функції. Для визначення відрізка, що вміщує один корінь, можемо скористатися наступною **теоремою** [5].

Теорема 2.1. Якщо функція $y = F(x)$ неперервна на інтервалі $[a, b]$ і якщо $F(a)$ та $F(b)$ мають протилежні знаки, тобто $F(a) \cdot F(b) < 0$, то $F(x)$ має хоча б один дійсний корінь на інтервалі $[a, b]$. Якщо при цьому $F(x)$ має першу похідну, що не змінює знак на інтервалі $[a, b]$, то корінь єдиний.

2.2. Етап 2: уточнення коренів

2.2.1. Метод половинного ділення

Припустимо, що рівняння $F(x) = 0$ має на відріжку $[a, b]$ єдиний корінь. Функція $F(x)$ на цьому відріжку неперервна. Поділимо відрізок $[a, b]$ точкою $c = \frac{a+b}{2}$ навпіл (рис.2.4). Якщо $F(c) = 0$, то розв'язок рівняння $x = c$. Якщо $F(c) \neq 0$, то вибираємо відрізок $[a, c]$ чи $[c, b]$, на якому функція змінює знак, тобто $F(a) \cdot F(c) < 0$ чи $F(c) \cdot F(b) < 0$. Для подальших поділів використовуємо цей відрізок. На кожному кроці процесу половинного ділення відрізок зменшується вдвічі.

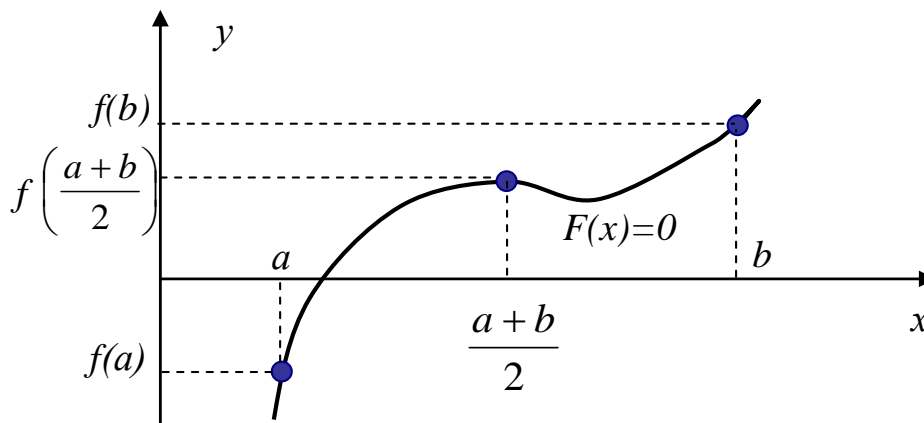


Рис. 2.4. Знаходження кореня рівняння методом половинного ділення.

Ітераційний процес продовжуємо, поки довжина відрізка не стане менше заданої точності ε :

$$|a - b| < \varepsilon \quad (2.6)$$

При оцінюванні похибки методу, можемо помітити, що вибираючи в якості розв'язку ліву (a) чи праву (b) границю остаточно розрахованого проміжку, допускаємо, що розв'язок може знаходитись на відстані, що не перевищує розмір проміжку, тобто меншій чи рівній ε . Якщо ж в якості відповіді обрати точку в середині проміжку, то точний результат не може відхилитися від неї більш ніж на $\varepsilon/2$, тому в даному випадку при реалізації алгоритму для досягнення заданої точності ε потрібно вийти з ітераційного процесу, коли виконається умова $|a - b| < 2 \cdot \varepsilon$.

Метод половинного ділення є абсолютно збіжним до розв'язку, оскільки його алгоритм не залежить від виду функції $F(x)$.

2.2.2. Метод хорд

Даний метод ґрунтується на лінійній інтерполяції функції $F(x)=0$ по двох точках, що мають протилежні знаки у значеннях функції $F(a)$ та $F(b)$ (рис. 2.5). Метод хорд швидше попереднього збігається до розв'язку навіть при досить малих значеннях ε .

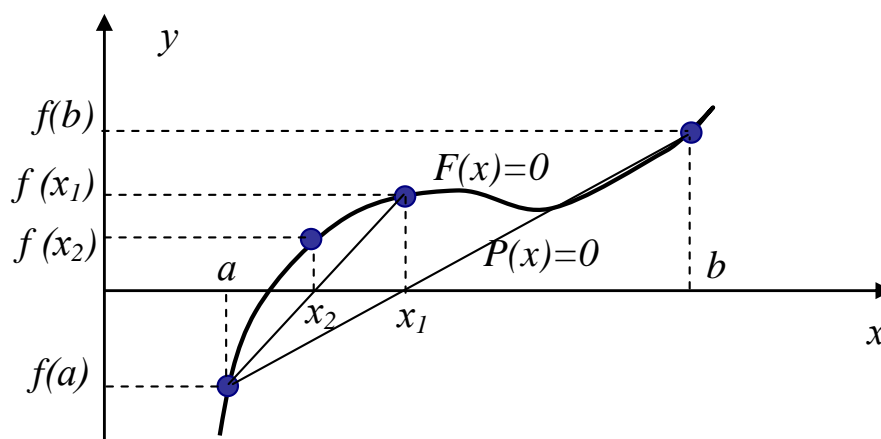


Рис. 2.5. Знаходження кореня за методом хорд.

Наприклад, потрібно знайти корінь рівняння $F(x)=0$ на проміжку $[a; b]$, і відомо, що $F(x)$ неперервна на $[a; b]$ та $F(a) \cdot F(b) < 0$. Крім того, перша $F'(x)$ і друга $F''(x)$ похідні функції $F(x)$ зберігають на

проміжку $[a; b]$ свій знак. Замінімо функцію $F(x)$ лінійною функцією, яка проходить через вузлові точки $(a, F(a))$ і $(b, F(b))$:

$$\frac{y - f(a)}{f(b) - f(a)} = \frac{x - a}{b - a} \Rightarrow P(x) = y = f(a) + \frac{x - a}{b - a} (f(b) - f(a)) \quad (2.7)$$

Лінійна функція $P(x)$ на кінцях відрізка $[a; b]$ приймає такі ж значення, як і функція $F(x)=0$.

В якості першого наближення при знаходженні кореня функції $F(x)=0$ візьмемо точне значення кореня функції $P(x)=0$, тобто x_1 , яке розрахуємо з рівняння:

$$f(a) + \frac{x_1 - a}{b - a} (f(b) - f(a)) = 0 \Rightarrow x_1 = a - f(a) \frac{b - a}{f(b) - f(a)} \quad (2.8)$$

При подальшому дослідженні відрізків $[a; x_1]$ та $[x_1; b]$, виберемо той, на якому функція змінює знак. На рисунку 2.5 бачимо, що таким відрізком є $[a; x_1]$. Для вибраного відрізка побудуємо лінійне наближення функції за формулою 2.7, виконаємо розрахунки кореня для лінійного наближення за формулою 2.8, в результаті розрахунків отримуємо координати точки x_2 . Ітераційний процес продовжуємо до отримання істинності нерівності: $|F(x_i)| < \varepsilon$.

Метод хорд має високу збіжність до розв'язку навіть для таких функцій $F(x)=0$, які мають низькі показники гладкості, але в окремих випадках швидкість збіжності може знижуватись, наприклад, на тих ділянках, де значення функції змінюється дуже повільно.

2.2.3. Метод січних

Метод січних подібний до методу хорд, тільки точки з координатами $(x_0, F(x_0))$ та $(x_1, F(x_1))$ взяті з одного боку від кореня рівняння $F(x)=0$. Геометрична інтерпретація методу представлена на рис. 2.6.

В якості початкового наближення обираємо точки $(x_0, f(x_0))$ та $(x_1, f(x_1))$. Через точки $(x_0, f(x_0))$ та $(x_1, f(x_1))$ проводимо січну до графіку функції, яка перетинає вісь x в точці $(x_2, 0)$. Перевіряємо виконання умови $|f(x_2)| < \varepsilon$, якщо вона не виконується, проводимо січну через точки $(x_1, f(x_1))$ та $(x_2, f(x_2))$, знаходимо точку перетину січної з віссю x (точка $(x_3, 0)$) і перевіряємо виконання чергової умови

$|f(x_3)| < \varepsilon$. Ітерації продовжуємо до виконання умови виходу з ітераційного процесу, тобто $|f(x_n)| < \varepsilon$.

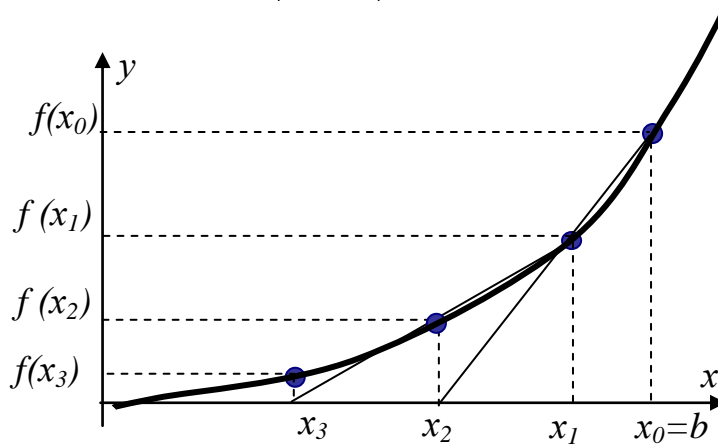


Рис. 2.6. Знаходження кореня за методом січних.

Для математичного опису методу січних отримаємо формулу прямої, що проходить через дві точки $(x_{n-1}, f(x_{n-1}))$ та $(x_n, f(x_n))$:

$$\frac{x_{n+1} - x_n}{f(x_{n+1}) - f(x_n)} = \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} \quad (2.9)$$

Враховуючи, що $f(x_{n+1}) = 0$ отримаємо загальну формулу для методу січних:

$$x_{n+1} = x_n - \frac{f(x_n) \cdot (x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} \quad (2.10)$$

Метод січних має досить високу збіжність до розв'язку у випадку, коли $F(x)=0$ гладка функція, але в процесі розв'язання деяких рівнянь швидкість збіжності може знижуватись, наприклад, на ділянках функції, близьких до функції $x = const$.

2.2.4. Метод дотичних (Ньютона)

Метод дотичних базується на заміні функції $F(x)=0$ у точці початкового наближення x_0 дотичною, яка при перетині з віссю x в точці $(x_1, 0)$ дає перше наближення. У цьому методі Ньютон замість інтерполяції використав екстраполяцію, що знаходиться за допомогою дотичної у визначеній точці. Геометрична інтерпретація методу дотичних (Ньютона) показана на рисунку 2.7. В якості початкового

наближення обираємо абсцису $x_0 = b$. У точці $(x_0, f(x_0))$ проводимо дотичну до графіку функції, яка перетинає вісь x в точці $(x_1, 0)$.

Перевіряємо виконання умови $|f(x_1)| < \varepsilon$, якщо вона не виконується, проводимо дотичну в точці $(x_1, f(x_1))$, знаходимо точку перетину дотичної з віссю x (точка $(x_2, 0)$) і перевіряємо виконання чергової умови $|f(x_2)| < \varepsilon$.

Ітераційний процес продовжуємо до виконання умови виходу, тобто $|f(x_n)| < \varepsilon$.

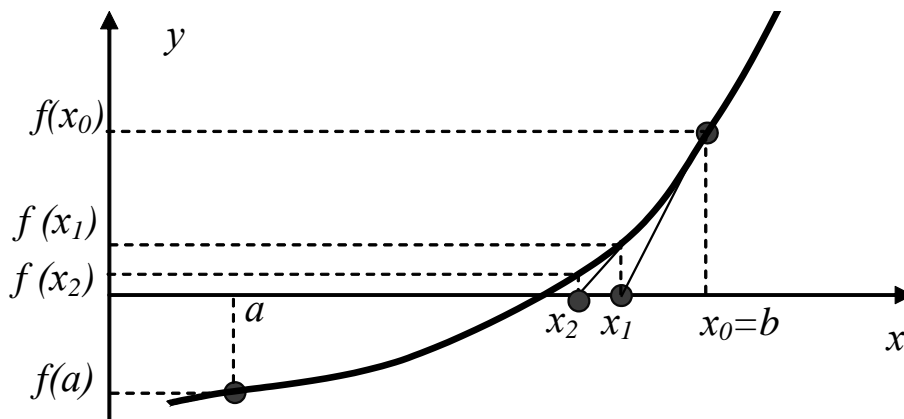


Рис. 2.7. Знаходження кореня за методом дотичних.

Для математичного опису методу дотичних отримаємо формулу дотичної, що проведена в точці $(x_0, f(x_0))$ і має кутовий коефіцієнт $f'(x_0)$:

$$y - f(x_0) = f'(x_0) \cdot (x_1 - x_0) \quad (2.11)$$

У точці перетину цієї дотичної з віссю x (точка $(x_1, 0)$) $y = 0$, тому формулу 2.11 можемо записати у вигляді:

$$-f(x_0) = f'(x_0) \cdot (x_1 - x_0) \Rightarrow x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} \quad (2.12)$$

Для будь якого кроку ітераційного процесу формулу Ньютона можемо записати у вигляді:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (2.13)$$

Метод дотичних (Ньютона) має високу збіжність до розв'язку, але час виконання ітерації дещо збільшується за рахунок необхідності обчислення похідної $f'(x_k)$.

2.2.5. Метод простої ітерації

Замінімо рівняння $F(x) = 0$ рівносильним рівнянням:

$$x = f(x) \quad (2.14)$$

Припустимо w – корінь рівняння, а x_0 – отримане будь-яким способом нульове наближення до кореня w . Підставимо x_0 в праву частину рівняння (2.14), отримаємо ітераційну послідовність: $x_1 = f(x_0)$; $x_2 = f(x_1)$; ... ; $x_n = f(x_{n-1})$. Таку числову послідовність називають послідовністю наближень. Послідовність наближень може бути збіжною чи розбіжною, і цей факт можемо визначити за наступною теоремою [6].

Теорема 2. *Ітераційна послідовність буде збіжною при будь-якому початковому значенні з інтервалу $x_0 \in [a; b]$, якщо для рівняння $x=f(x)$, що має єдиний розв'язок на проміжку $[a, b]$, виконуються умови:*

- 1) $f(x)$ визначена для всіх $x \in [a, b]$;
- 2) $f(x)$ диференційована на відрізку $[a, b]$;
- 3) існує таке дійсне q , яке задовольняє нерівності $|f'(x)| \leq q < 1$ для всіх $x \in [a, b]$.

Умови теореми є достатніми, але не являються необхідними, тому при їх невиконанні, іноді, ітераційна послідовність може виявитися збіжною до розв'язку.

Швидкість збіжності характеризується нерівністю:

$$|x_k - w| \leq q^k |x_0 - w| \quad (2.15)$$

де x_0 – початкове значення змінної x ; x_k – поточне значення змінної x , тобто значення на k -й ітерації; w – точний розв'язок рівняння.

Умовою припинення ітерацій є нерівність (умова Ліпшиця):

$$|f(x_n) - f(x_{n-1})| \leq \frac{q}{1-q} |x_n - x_{n-1}| \quad (2.16)$$

Для забезпечення умови збіжності ітераційного процесу проведемо перетворення рівняння $F(x) = 0$ до ітераційного вигляду:

$$F(x) = 0 \Rightarrow x = x - m \cdot F(x) \quad (2.17)$$

З умов виконання третього пункту теореми 2 припустимо, що $|f'(x)| = q$, тоді отримаємо рівняння:

$$f'(x) = 1 - m \cdot F'(x) < 1 \quad (2.18)$$

Достатньо підібрати значення m так, щоб для всіх $x \in [a, b]$ значення добутку $m \cdot F'(x) < 1$, тобто при рівності правої і лівої частин $m = 1/(\max(F'(x)))$ для $x \in [a, b]$.

Контрольні питання

1. Назвіть види функціональних рівнянь з однією змінною.
2. Що значить розв'язати задачу в теорії чисельних методів?
3. З яких етапів складається розв'язання задачі знаходження коренів функціонального рівняння з однією змінною?
4. Які методи реалізації другого етапу розв'язання функціональних рівнянь мають абсолютну збіжність до розв'язку?
5. В чому полягає суть алгоритму реалізації методу відокремлення коренів?
6. Сутність методу половинного ділення. Особливості обрання остаточного розв'язку.
7. Сутність методу хорд. Умова виходу з ітераційного процесу.
8. Сутність методу січних. Від яких факторів і чому залежить швидкість збіжності методу?
9. Сутність методу дотичних (Ньютона). З чим пов'язана швидкість обчислень за алгоритмом методу дотичних?
10. Сутність методу простої ітерації. Які умови повинні виконуватися для забезпечення збіжності методу?
11. Як потрібно перетворити рівняння за умовою збіжності методу простої ітерації?

Розділ 3. Прямі та непрямі методи розв'язання систем лінійних алгебраїчних рівнянь. Методи Гауса та LU-розкладу

3.1. Основні поняття

Математичні моделі багатьох технічних задач представлені системами лінійних рівнянь. Багато методів розв'язання нелінійних задач також зводяться до розв'язання деякої послідовності систем лінійних алгебраїчних рівнянь (СЛАР). Для багатьох методів розроблений математичний апарат, що дозволяє оцінити точність отриманого розв'язку. Чисельні методи розв'язку СЛАР поділяються на прямі (точні) та непрямі (ітераційні, наближені).

Прямі (точні) методи дозволяють розв'язати систему рівнянь за скінчене число арифметичних операцій. Якщо всі операції виконуються точно (без помилок округлення), то розв'язок заданої системи також отримуємо точним. До прямих методів належать: метод послідовного виключення невідомих (метод Гауса та його модифікації: метод головного елемента, метод квадратного кореня, метод відображень та ін.), метод ортогоналізації, метод LU-розкладу. Прямі методи застосовують на практиці для розв'язання СЛАР за допомогою обчислювальної техніки, як правило, з числами порядку не вище 10^3 [5].

Ітераційні методи є наближеними. Вони дозволяють знайти розв'язок системи, як межу послідовних наближень, що обчислюються по однаково алгоритму. Для застосування ітераційних методів у початкових умовах необхідно задати точність обчислень ε і початкове наближення x_0 чи $(x_0^1, x_0^2, x_0^3, \dots)$. До ітераційних методів належать: метод Зейделя, метод простої ітерації, метод релаксації, градієнтні методи та їх модифікації. На практиці ітераційні методи застосовують для розв'язання СЛАР з числами порядку 10^6 і вище.

Розглянемо систему m лінійних алгебраїчних рівнянь з n невідомими:

Для зменшення обчислювальної похибки застосовують перестановки рівнянь у системі в залежності від головного елемента. *Головним елементом* називають максимальний коефіцієнт рівняння. Сукупність рівнянь в системі потрібно переставити таким чином, щоб на головній діагоналі матриці коефіцієнтів стояли максимальні по модулю елементи. Метод Гауса у такій модифікації називається **методом Гауса з вибором головного елемента**.

3.2.2. Метод LU-розкладу

При розв'язанні системи лінійних алгебраїчних рівнянь даним методом матрицю коефіцієнтів A розкладають на добуток двох матриць (формула 3.5) нижньої трикутної матриці L , на головній діагоналі якої стоять одиниці, та верхньої трикутної матриці U , елементи головної діагоналі якої не дорівнюють нулю. У матричному вигляді при $n = 4$ це можна записати таким чином:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ m_{21} & 1 & 0 & 0 \\ m_{31} & m_{32} & 1 & 0 \\ m_{41} & m_{42} & m_{43} & 1 \end{pmatrix} \cdot \begin{pmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{pmatrix} \quad (3.5)$$

Для розкладання матриці коефіцієнтів A на трикутні матриці використаємо метод виключення Гауса. Отримаємо матрицю L з допомогою одиничної матриці. Для цього помножимо зліва матрицю A на одиничну матрицю. Наприклад:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 4 & 3 & -1 \\ -2 & -4 & 5 \\ 1 & 2 & 6 \end{pmatrix} \quad (3.6)$$

Перший рядок матриці A використовуємо для обнуління елементів першого стовпчика цієї ж матриці. Помножимо перший рядок на $-0,5$ і віднімемо від другого, помножимо перший рядок на $0,25$ і віднімемо від третього рядка. Використані коефіцієнти для перетворень запишемо на місця елементів одиничної матриці, що відповідають індексам обнулених елементів матриці A . В результаті отримаємо:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ -0,5 & 1 & 0 \\ 0,25 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 4 & 3 & -1 \\ 0 & -2,5 & 4,5 \\ 0 & 1,25 & 6,25 \end{pmatrix} \quad (3.7)$$

Другий рядок матриці A використовуємо для обнуління елементів другого стовпчика цієї ж матриці, що знаходяться нижче другого рядка (у матриці, що утворена з одиничної, запишемо використаний для обнулення коефіцієнт на місце елемента, що відповідає індексам обнуленого елемента матриці A):

$$A = \begin{pmatrix} 1 & 0 & 0 \\ -0,5 & 1 & 0 \\ 0,25 & -0,5 & 1 \end{pmatrix} \cdot \begin{pmatrix} 4 & 3 & -1 \\ 0 & -2,5 & 4,5 \\ 0 & 0 & 8,5 \end{pmatrix} \quad (3.8)$$

В результаті отримаємо LU-розклад матриці A і розв'яжемо систему рівнянь у два етапи:

$$\begin{aligned} AX = B &\Rightarrow LUX = B \\ \text{1 етап } LY = B &\Rightarrow Y \\ \text{2 етап } UX = Y &\Rightarrow X \end{aligned} \quad (3.9)$$

На першому етапі знаходимо проміжний вектор Y , використовуючи пряму підстановку, на другому етапі знаходимо безпосередньо вектор розв'язків X , застосовуючи зворотню підстановку. Зокрема, приймаючи у наведеному прикладі $B = (1 \ 6 \ 14)^T$ з проміжної системи $LY = B$ матимемо $Y = (1 \ 6,5 \ 17)^T$. Здійснюючи другий етап розв'язання системи, тобто розв'язуючи рівняння $UX = Y$, отримаємо $X = (0 \ 1 \ 2)^T$.

3.2.3. Зв'язок методу Гауса з методом LU - розкладу

Алгоритм методу Гауса можна записати у матричному вигляді. Він полягає у розкладанні матриці коефіцієнтів A на добуток більш простих матриць. Для наочності розглянемо систему лінійних алгебраїчних рівнянь, що складається з 3-х рівнянь:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases} \quad (3.10)$$

Для виключення невідомої x_1 з другого і третього рівнянь системи (3.10) виконаємо наступні операції:

- ділення першого рівняння на $a_{11} = 0$,
- віднімання перетвореного першого рівняння, помноженого на a_{21} та a_{31} відповідно від другого і третього рівнянь.

Перша з цих операцій еквівалентна множенню системи рівнянь зліва на діагональну матрицю:

$$D_1 = \begin{pmatrix} 1/a_{11} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Друга операція еквівалентна множенню зліва на матрицю:

$$Q_1 = \begin{pmatrix} 1 & 0 & 0 \\ -a_{21} & 1 & 0 \\ -a_{31} & 0 & 1 \end{pmatrix}$$

Таким чином, для виключення невідомої x_1 з другого і третього рівнянь системи (3.10) потрібно виконати наступне перетворення:

$$Q_1 D_1 A X = Q_1 D_1 B \quad \Rightarrow \quad L_1 A X = L_1 B$$

$$\text{де } L_1 = Q_1 D_1 = \begin{bmatrix} 1/a_{11} & 0 & 0 \\ -a_{21}/a_{11} & 1 & 0 \\ -a_{31}/a_{11} & 0 & 1 \end{bmatrix} \quad (3.11)$$

В результаті отримаємо систему рівнянь:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 = b_2^{(1)} \\ a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 = b_3^{(1)} \end{cases} \quad (3.12)$$

Другий крок прямого ходу методу Гауса, який полягає у виключенні невідомої x_2 з третього рівняння, можемо виконати, помноживши систему рівнянь на матрицю:

$$L_2 = Q_2 D_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/a_{22}^{(1)} & 0 \\ 0 & -a_{32}^{(1)}/a_{22}^{(1)} & 1 \end{bmatrix}$$

В результаті отримаємо наступну систему рівнянь:

$$\begin{bmatrix} 1 & a_{12}^{(1)} & a_{22}^{(1)} \\ 0 & 1 & \frac{a_{23}^{(1)}}{a_{22}^{(1)}} \\ 0 & 0 & a_{33}^{(1)} - \frac{a_{23}^{(1)}a_{32}^{(1)}}{a_{22}^{(1)}} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ \frac{b_2^{(1)}}{a_{22}^{(1)}} \\ b_3^{(1)} - \frac{b_2^{(1)}a_{32}^{(1)}}{a_{22}^{(1)}} \end{bmatrix} \quad (3.13)$$

Для останнього кроку прямого ходу методу Гауса помножимо рівняння з раніше виконаними перетвореннями $L_2L_1AX = L_2L_1B$ на матрицю:

$$L_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/a_{33}^{(2)} \end{bmatrix}$$

Таким чином, сукупність виконаних перетворень можемо записати у вигляді:

$$L_3L_2L_1AX = L_3L_2L_1B \Rightarrow L_3L_2L_1A = U \Rightarrow UX = B.$$

Їх результатом є верхня трикутна матриця U з одиничною головною діагоналлю:

$$U = \begin{bmatrix} 1 & a_{12}^{(1)} & a_{22}^{(1)} \\ 0 & 1 & a_{23}^{(2)} \\ 0 & 0 & 1 \end{bmatrix}$$

Система рівнянь (3.12) після перетворень матиме вигляд:

$$\begin{bmatrix} 1 & a_{12}^{(1)} & a_{22}^{(1)} \\ 0 & 1 & a_{23}^{(2)} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ b_3^{(2)}/a_{33}^{(2)} \end{bmatrix} \quad (3.14)$$

Оскільки $A = LU$, то потрібно обчислити L . Матрицю L – нижню трикутну матрицю – отримаємо за формулою: $L = L_1^{-1}L_2^{-1}L_3^{-1}$. Обернені матриці L_i^{-k} мають наступний вигляд:

$$L_1^{-1} = \begin{bmatrix} a_{11} & 0 & 0 \\ a_{21} & 1 & 0 \\ a_{31} & 0 & 1 \end{bmatrix}, L_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & a_{22}^{(1)} & 0 \\ 0 & a_{32}^{(1)} & 1 \end{bmatrix}, L_3^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & a_{33}^{(2)} \end{bmatrix}$$

а нижня трикутна матриця L запишеться так: $L = \begin{bmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22}^{(1)} & 0 \\ a_{31} & a_{32}^{(1)} & a_{33}^{(2)} \end{bmatrix}$.

Таким чином, отримані матриці L і U є дзеркальним відображенням одноіменних матриць, які використовувалися в класичному варіанті методу LU -розкладу (п.3.2.2). Отримані матриці L і U потрібно використовувати при розв'язанні СЛАР в два етапи: 1) $UY = B$; 2) $LX = Y$.

3.3 Непрямі методи розв'язання систем лінійних алгебраїчних рівнянь

3.3.1. Забезпечення збіжності ітераційного процесу

Непрямі (ітераційні, неточні) методи дозволяють наближено знайти розв'язок системи лінійних алгебраїчних рівнянь, як границю послідовних наближень, що обчислюються за певним алгоритмом.

До ітераційних методів належать: метод Зейделя, метод простої ітерації, метод релаксації, градієнтні методи та їх модифікації. Застосовуються ітераційні методи для розв'язання СЛАР з коефіцієнтами порядку 10^6 .

Ознакою ітераційного методу розв'язання системи ЛАР є наявність початкового наближення x_0 чи $(x_0^1, x_0^2, x_0^3, \dots)$ і потрібної точності отримання розв'язків ε .

Для успішного застосування ітераційних методів до початку ітерацій потрібно напевне знати, що ітераційний процес буде збігатися до конкретного розв'язку. Для цього початкову систему приводять до *нормалізованого вигляду* таким чином:

$$\begin{aligned} \text{Початкова система} \quad A \cdot X &= B \\ \text{Нормалізована система} \quad A^T \cdot A \cdot X &= A^T \cdot B \end{aligned} \quad (3.15)$$

Після заміни початкової системи лінійних рівнянь еквівалентною їй – нормалізованою системою – застосуємо ітераційні чисельні методи розв'язання СЛАР, забезпечивши, таким чином, збіжність ітераційного процесу до розв'язку.

3.3.2. Метод простої ітерації та метод Зейделя для розв'язання систем лінійних алгебраїчних рівнянь

Першим необхідним корком при застосуванні методів *простої ітерації* та *Зейделя* є приведення системи лінійних алгебраїчних рівнянь до нормалізованого вигляду, представленого формулою 3.15.

Для здійснення ітераційного процесу нормалізовану систему приводимо до ітераційного вигляду, тобто виражаємо з кожного i -го рівняння відповідну змінну x_i :

$$\begin{aligned}x_1 &= a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n \\x_2 &= a_{21}x_1 + a_{23}x_3 + \dots + a_{2n}x_n \\&\dots \\x_n &= a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn-1}x_{n-1}\end{aligned}\tag{3.16}$$

Для початку ітерацій задаємо початкові значення всіх невідомих $x_1, x_2, x_3, \dots, x_{n-1}, x_n$.

Метод простих ітерацій полягає у підстановці значень вектору початкових наближень $(x_1^0, x_2^0, x_3^0, \dots, x_n^0)$ в праву частину системи (3.16) і отримання в лівій частині чергового вектора наближень $(x_1^1, x_2^1, x_3^1, \dots, x_n^1)$. Збіжність методу простих ітерацій досить повільна, для її підвищення Зейдель запропонував підстановку, яка дозволяє в $(n-1)$ раз зменшити кількість кроків і, таким чином, підвищити збіжність методу.

При здійсненні ітерацій **підстановка Зейделя** полягає в тому, що знайшовши з першого рівняння x_1^1 по значеннях вектору початкових наближень інших змінних, ми використовуємо його при отриманні значення x_2^1 у складі вектора чергових наближень $(x_1^1, x_2^0, x_3^0, \dots, x_n^0)$. Тобто всі попередньо знайдені значення поточної ітерації використовуються для знаходження наступних значень цієї ж ітерації. Ітераційний процес можна записати наступним чином:

$$\begin{aligned}x_1^1 &= a_{12}x_2^0 + a_{13}x_3^0 + a_{14}x_4^0 \dots + a_{1n}x_n^0 \\x_2^1 &= a_{21}x_1^1 + a_{23}x_3^0 + a_{24}x_4^0 \dots + a_{2n}x_n^0 \\x_3^1 &= a_{31}x_1^1 + a_{32}x_2^1 + a_{34}x_4^0 \dots + a_{3n}x_n^0 \\&\dots \\x_n^1 &= a_{n1}x_1^1 + a_{n2}x_2^1 + a_{n3}x_3^1 \dots + a_{nn-1}x_{n-1}^1\end{aligned}\tag{3.17}$$

Умовою припинення ітерацій є виконання нерівностей $|x_i^{n-1} - x_i^n| < \varepsilon$ для всіх змінних x_i . При виході з ітераційного процесу можемо констатувати, що розв'язок знайдено з точністю ε .

3.3.3. Метод релаксації для розв'язання систем лінійних алгебраїчних рівнянь

Даний метод є ітераційним і для його успішного застосування потрібно забезпечити збіжність ітераційного процесу до розв'язку, для чого приведемо початкову систему лінійних алгебраїчних рівнянь до нормалізованого вигляду, представленого формулою 3.15.

Перед початком першої ітерації перетворюємо нормалізовану систему до канонічного вигляду (формула 2.3). Для початку обчислень вибираємо початкове наближення $(x_1^0, x_2^0, \dots, x_n^0)$ і точність обчислень ε . На першому етапі обчислюємо, так звані, нев'язки, тобто відхилення від нуля параметрів δ_i :

$$\begin{cases} \delta_1 = a_{11}x_1^0 + a_{12}x_2^0 + \dots + a_{1n}x_n^0 - b_1 \\ \delta_2 = a_{21}x_1^0 + a_{22}x_2^0 + \dots + a_{2n}x_n^0 - b_2 \\ \dots \\ \delta_i = a_{i1}x_1^0 + a_{i2}x_2^0 + \dots + a_{in}x_n^0 - b_i, \\ \dots \\ \delta_n = a_{n1}x_1^0 + a_{n2}x_2^0 + \dots + a_{nn}x_n^0 - b_n \end{cases} \quad (3.18)$$

які при канонічному представленні системи рівнянь повинні дорівнювати нулю.

В системі знаходимо рівняння з максимальною по модулю нев'язкою: $\delta_i = a_{i1}x_1^0 + a_{i2}x_2^0 + \dots + a_{in}x_n^0 - b_i$, i , обнулюючи δ_i ($\delta_i = 0$), обчислюємо значення x_1^1 за формулою:

$$x_1^1 = \frac{1}{a_{i1}}(b_i - a_{i2}x_2^0 - \dots - a_{in}x_n^0) \quad (3.19)$$

де i – номер рівняння з максимальною по модулю нев'язкою. На наступному кроці підраховуємо нев'язки за оновленим вектором наближень $(x_1^1, x_2^0, \dots, x_n^0)$ і вибираємо рівняння з максимальною по модулю нев'язкою δ_j :

$$\delta_j = a_{j1}x_1^1 + a_{j2}x_2^0 + \dots + a_{jn}x_n^0 - b_j, \quad (3.20)$$

після чого розраховуємо x_2^1 , що задовольняє рівності:

$$a_{j1}x_1^1 + a_{j2}x_2^1 + a_{j3}x_3^0 + \dots + a_{jn}x_n^0 = b_j \quad (3.21)$$

При знаходженні чергового значення x_j^n не забувайте обнулювати δ_j !

Розрахунки продовжуємо до знаходження всіх значень x_j^1 , після цього перевіряємо умову припинення ітерацій, яка записується у вигляді: $\delta_{\max} \leq \varepsilon$.

Якщо умова не виконується розпочинаємо наступний цикл ітерацій, який проводиться так, як і попередній, але в якості початкових значень невідомих виступають значення, обчислені у попередньому циклі ітерацій, наприклад: $(x_3^1, x_2^1, \dots, x_n^1)$.

Цикли ітерацій продовжуємо до отримання необхідної точності ε , яка визначається за простою умовою $\delta_{\max} \leq \varepsilon$, саме тому даний метод називається методом *релаксації*.

При реалізації методу релаксації рівняння намагаються вибирати в такому порядку, щоб за найменшу кількість кроків знайти розв'язок із заданою точністю. Оскільки такий вибір є невизначеним даний метод називають нестационарним [7].

Контрольні питання

1. Якими методами розв'язують системи лінійних алгебраїчних рівнянь?
2. В яких випадках варто застосовувати точні методи розв'язку СЛАР?
3. В чому полягає сутність методу LU-розкладу?
4. В чому сутність зв'язку методів Гауса і LU-розкладу?
5. Який обов'язковий етап потрібно виконати перед розв'язанням СЛАР неточними (ітераційними) методами? Для чого потрібен даний етап?
6. Сутність методу простої ітерації для розв'язання СЛАР.
7. Чим метод Зейделя відрізняється від методу простої ітерації при розв'язанні СЛАР?
8. Сутність методу релаксації для розв'язання СЛАР. Як регулюється точність розв'язків в даному методі?
9. Чому при розв'язанні СЛАР методом релаксації виникають нев'язки?
10. Чому метод релаксації є нестационарним?

Розділ 4. Розв'язання систем нелінійних рівнянь. Метод Ньютона

Не існує прямих методів розв'язання нелінійних та трансцендентних рівнянь. Розв'язання таких рівнянь складається з двох етапів: відокремлення коренів та їх уточнення. Процедура розв'язання починається з вибору початкової точки x_0 і обчислення нев'язки рівняння $f(x_0)$. Якщо $\varepsilon < f(x_0)$ вибирається алгоритм, за яким проводиться уточнення розв'язків (при цьому використовується інформація про знак нев'язки, її значення, або про швидкість її зміни $f(x_0)/dx$).

Для розв'язання нелінійних та трансцендентних рівнянь можуть застосовуватися звичайний ітераційний метод. Але при знаходженні розв'язків збіжність ітераційного методу до конкретного розв'язку залежить від початкових значень змінних [7]. Також складно вибрати функції f_i , які б задовольняли умову збіжності $\|J(x)\| \leq M < 1$, де J - матриця Якобі, що визначається формулою 4.1. Тому для розв'язання систем нелінійних і трансцендентних рівнянь застосовують *узагальнений метод Ньютона* (далі – метод Ньютона).

Метод Ньютона оснований на знаходженні послідовності $\{[x_1^k, x_2^k, \dots, x_n^k]\}$, що збігається до розв'язку (x_1, x_2, \dots, x_n) . Цей метод називають ітерацією нерухомої точки. Величина похідної в нерухомій точці визначає, чи буде ітераційний процес збіжним (теорема 4.1). Коли це правило застосовується для функції декількох змінних – похідні повинні бути частинними. Узагальненням похідної для системи функцій є матриця Якобі (Якобіан). Наприклад, для функцій трьох незалежних змінних $f_1(x, y, z)$, $f_2(x, y, z)$, $f_3(x, y, z)$ матриця Якобі має вигляд:

$$J(x, y, z) = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} \end{bmatrix} \quad (4.1)$$

Для функцій декількох змінних диференціал використовується, щоб показати, як змінення незалежних змінних (x, y, z) вплине на залежні змінні (u, v, ω) . Наприклад, задані функції:

$$u = f_1(x, y, z), \quad v = f_2(x, y, z), \quad \omega = f_3(x, y, z) \quad (4.2)$$

Допустимо, що значення цих функцій відомі в точці (x_0, y_0, z_0) і необхідно визначити їх значення в точці (x, y, z) віддаленій на $(\Delta x, \Delta y, \Delta z)$.

$$\begin{aligned} du &= \frac{\partial f_1}{\partial x}(x_0, y_0, z_0)dx + \frac{\partial f_1}{\partial y}(x_0, y_0, z_0)dy + \frac{\partial f_1}{\partial z}(x_0, y_0, z_0)dz, \\ dv &= \frac{\partial f_2}{\partial x}(x_0, y_0, z_0)dx + \frac{\partial f_2}{\partial y}(x_0, y_0, z_0)dy + \frac{\partial f_2}{\partial z}(x_0, y_0, z_0)dz, \\ d\omega &= \frac{\partial f_3}{\partial x}(x_0, y_0, z_0)dx + \frac{\partial f_3}{\partial y}(x_0, y_0, z_0)dy + \frac{\partial f_3}{\partial z}(x_0, y_0, z_0)dz, \end{aligned} \quad (4.3)$$

де $du, dv, d\omega$ – диференціали залежних змінних, dx, dy, dz – диференціали незалежних змінних. Якщо змінення функції позначити dF , а змінення змінних dX , використовуючи векторне позначення можемо записати:

$$dF = \begin{bmatrix} \partial u \\ \partial v \\ \partial \omega \end{bmatrix} = J(x_0, y_0, z_0) \cdot \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = J(x_0, y_0, z_0) \cdot dX \quad (4.4)$$

Збіжність поблизу нерухомої точки. Ітерацію нерухомої точки визначаємо наступним чином [4]:

$$\begin{aligned} x_{k+1} &= f_1(x_k, y_k, z_k) \\ y_{k+1} &= f_2(x_k, y_k, z_k) \\ z_{k+1} &= f_3(x_k, y_k, z_k) \end{aligned} \quad (4.5)$$

Теорема 4.1. Припустимо, що функції (4.2) та їх перші частинні похідні неперервні в області, в якій знаходиться нерухома точка (x, y, z) . Якщо початкова точка достатньо близько розташована до точки (x, y, z) і виконуються умови:

$$\begin{aligned} \left| \frac{\partial f_1}{\partial x}(x, y, z) \right| + \left| \frac{\partial f_1}{\partial y}(x, y, z) \right| + \left| \frac{\partial f_1}{\partial z}(x, y, z) \right| &< 1, \\ \left| \frac{\partial f_2}{\partial x}(x, y, z) \right| + \left| \frac{\partial f_2}{\partial y}(x, y, z) \right| + \left| \frac{\partial f_2}{\partial z}(x, y, z) \right| &< 1, \\ \left| \frac{\partial f_3}{\partial x}(x, y, z) \right| + \left| \frac{\partial f_3}{\partial y}(x, y, z) \right| + \left| \frac{\partial f_3}{\partial z}(x, y, z) \right| &< 1 \end{aligned} \quad (4.6)$$

то ітерація збігається до нерухомої точки (x, y, z) .

Дана теорема визначає достатню умову збіжності, але вона не є необхідною. Так деякі системи нелінійних алгебраїчних рівнянь можуть сходитися до розв'язку при невиконанні умови (4.6). Таку систему нелінійних рівнянь розглянемо у прикладі 4.1., що наведений нижче.

Метод Ньютона виконується за наступними етапами:

1 етап: для здійснення обчислень сформуємо функцію:

$$F(P_k) = \begin{bmatrix} f_1(x_k, y_k, z_k) \\ f_2(x_k, y_k, z_k) \\ f_3(x_k, y_k, z_k) \end{bmatrix} \quad (4.7)$$

2 етап: обчислимо Якобіан:

$$J(P_k) = \begin{bmatrix} \frac{\partial}{\partial x} f_1(x_k, y_k, z_k) & \frac{\partial}{\partial y} f_1(x_k, y_k, z_k) & \frac{\partial}{\partial z} f_1(x_k, y_k, z_k) \\ \frac{\partial}{\partial x} f_2(x_k, y_k, z_k) & \frac{\partial}{\partial y} f_2(x_k, y_k, z_k) & \frac{\partial}{\partial z} f_2(x_k, y_k, z_k) \\ \frac{\partial}{\partial x} f_3(x_k, y_k, z_k) & \frac{\partial}{\partial y} f_3(x_k, y_k, z_k) & \frac{\partial}{\partial z} f_3(x_k, y_k, z_k) \end{bmatrix} \quad (4.8)$$

3 етап: розв'яжемо систему рівнянь:

$$J(P_k) \cdot \Delta P = -F(P_k)$$

Якщо матриця Якобі не вироджена, то розв'язок системи запишемо у вигляді:

$$\Delta P \approx -J(P_k)^{-1} \cdot F(P_k)$$

4 етап: обчислимо координати наступної точки – наступне наближення до розв'язку має вигляд:

$$P_1 = P_0 + \Delta P \approx P_0 - J(P_k)^{-1} \cdot F(P_k) \quad (4.9)$$

При графічному представленні процесу розв'язання системи нелінійних алгебраїчних рівнянь методом Ньютона початкову точку розглядаємо в якості опорної, Якобіан визначає напрямок вектору до наступної точки розв'язку, а значення функції, обчислене по даним

початкової точки, – величину кроку у визначеному Якобіаном напрямку.

Приклад 4.1. Розв’яжемо систему нелінійних рівнянь:

$$\begin{cases} x^2 - 2x - y + 0,5 = 0 \\ x^2 + 4y^2 - 4 = 0 \end{cases}$$

Початкові значення для даної системи нелінійних рівнянь при розв’язанні методом Ньютона: $(x_0, y_0) = (2.00, 0.25)$.

Сформуємо вектор-функцію і обчислимо матрицю Якобі:

$$F(x, y) = \begin{bmatrix} x^2 - 2x - y + 0,5 \\ x^2 + 4y^2 - 4 \end{bmatrix}, \quad J(x, y) = \begin{bmatrix} 2x - 2 & -1 \\ 2x & 8y \end{bmatrix}.$$

В початковій точці вони приймуть значення:

$$F(2.00, 0.25) = \begin{bmatrix} 0.25 \\ 0.25 \end{bmatrix}, \quad J(2.00; 0.25) = \begin{bmatrix} 2.0 & -1.0 \\ 4.0 & 2.0 \end{bmatrix}.$$

Обчислимо Δx , Δy з лінійної системи рівнянь:

$$\begin{bmatrix} 2.0 & -1.0 \\ 4.0 & 2.0 \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = - \begin{bmatrix} 0.25 \\ 0.25 \end{bmatrix}$$

Значення невідомих знаходять будь-яким, переважно точним, методом для розв’язання систем лінійних рівнянь:

$$\Delta P = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} -0.09375 \\ 0.0625 \end{bmatrix}$$

Розрахуємо значення координат наступної точки:

$$P_1 = P_0 + \Delta P = \begin{bmatrix} 2.00 \\ 0.25 \end{bmatrix} + \begin{bmatrix} -0.09375 \\ 0.0625 \end{bmatrix} = \begin{bmatrix} 1.90625 \\ 0.3125 \end{bmatrix}$$

Аналогічно знайдемо два наступні розв’язки:

$$P_2 = \begin{bmatrix} 1.900691 \\ 0.311213 \end{bmatrix} \quad P_3 = \begin{bmatrix} 1.900677 \\ 0.311219 \end{bmatrix}$$

Ітерації продовжуємо до досягнення заданої точності обчислень. В розглянутому прикладі точність обмежувалася чотирма десятковими знаками після коми, оскільки

$$|x_3 - x_2| = |1.900677 - 1.900691| = 0.000014 < 0.0001 = \varepsilon .$$

Для розв'язання систем нелінійних рівнянь можемо також застосовувати нелінійний ітераційний метод Зейделя [6], який передбачає незалежне розв'язання кожного нелінійного рівняння системи відносно однієї (чергової) змінної, як це виконувалося і для систем лінійних алгебраїчних рівнянь (розділ 3, п. 3.3.2).

Контрольні питання

1. В чому полягає суть методів розв'язання систем нелінійних та трансцендентних рівнянь?
2. Розкрийте фізичний сенс Якобіана.
3. За яких умов ітерації в методі Ньютона збігаються до нерухомої точки? Чому дана умова є достатньою?
4. З яких етапів складається алгоритм методу Ньютона для розв'язання нелінійних та трансцендентних систем рівнянь?
5. В чому полягає особливість програмної реалізації алгоритму методу Ньютона для розв'язання нелінійних та трансцендентних систем рівнянь?
6. Яким чином застосовується метод Зейделя для розв'язання систем нелінійних алгебраїчних рівнянь?

Розділ 5. Інтерполяція функцій. Інтерполяційні поліноми Лагранжа. Сплайн-інтерполяція

На практиці часто приходиться розв'язувати задачі, в яких складні функції простіше обчислити по їх наближеним аналогам [8]. Наприклад, для обчислення стандартних функцій $\sin(x)$, $\cos(x)$, e^x в пакетах прикладних програм використовуються обчислювальні процедури, які ґрунтуються на заміні заданої функції наближеними функціями, побудованими на основі поліномів n -го порядку.

Ще одним прикладом застосування поліномів є випадок, коли функція задана у вигляді таблиці вузлових точок. Для відображення поданих значень функцією і подальшому застосуванні цієї функції у розрахунках, будують поліноміальну криву $y = P(x)$, що проходить через вузлові точки (для обчислення поліноміальної кривої повинен бути визначений проміжок, на якому будується наближення). За допомогою такої функції можливо знайти наближені значення в точках, що не є вузловими. Якщо така точка знаходиться у межах інтервалу наближення $x_0 < x < x_N$, її значення називають *інтерполяційним*, якщо за межами інтервалу $x < x_0$ чи $x_N < x$ – *екстраполяційними*. Так побудова поліному для знаходження проміжних точок називається *інтерполяцією*, для знаходження значень за межами заданого інтервалу – *екстраполяцією*. Широке застосування наближуючі поліноми знаходять також у чисельному диференціюванні, чисельному інтегруванні, розв'язанні нелінійних рівнянь та обробці графічних зображень. Наближення реальної функції деякою простішою функцією називається *апроксимацією*.

Припустимо, що функція $f(x)$ задана таблицею значень $f(x_0)$, $f(x_1)$, $f(x_2)$, ..., $f(x_n)$ для деякої кінцевої множини аргументів x_i (вузлові точки) і при розв'язуванні задачі потрібно використовувати значення функції $f(x)$ у проміжних точках. Для апроксимації функції $f(x)$ використовують функцію $\varphi(x)$, яку будують так, щоб у заданих точках $x_0, x_1, x_2, \dots, x_n$ вона приймала значення, які збігаються зі значеннями $f(x_0), f(x_1), f(x_2), \dots, f(x_n)$, а в інших точках відрізка $[a, b]$ приблизно зображувала функцію $f(x)$. Ступінь точності опису визначається в залежності від вибраного методу апроксимації.

Найчастіше функцію $\varphi(x)$ будують у вигляді поліному. Теорема Вєрштрасса [8] стверджує, що функція $f(x)$ може бути досить добре наближена за допомогою полінома деякого порядку m на множині

точок $x_0, x_1, x_2, \dots, x_n$. Функцію $\varphi(x)$, яка наближує вихідну функцію $f(x)$, у загальному випадку можна записати у вигляді:

$$\varphi(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \dots + c_m\varphi_m(x) \quad (5.1)$$

де c_0, c_1, \dots, c_m – деякі константи. Функція $f(x)$ вважається наближено описана поліномом $\varphi(x)$, якщо $f(x)$ і $\varphi(x)$ мають однакові значення у вузлових точках. Іноді, якщо функція і поліном диференційовані на проміжку наближення $[a, b]$, вимагають збігу у вузлах похідних $f(x)$ і $\varphi(x)$ визначених порядків.

На практиці в якості базисних функцій $\{\varphi(x)\}$ обирають послідовність степеневих функцій відносно x : $1, x, x^2, x^3, \dots, x^m$, тобто $\varphi(x_0)=1, \varphi(x_1)=x, \varphi(x_2)=x^2, \dots, \varphi(x_m)=x^m$. В такому випадку поліном запишемо у вигляді:

$$\varphi(x) = c_0 + c_1x + c_2x^2 + \dots + c_mx^m \quad (5.2)$$

Для знаходження коефіцієнтів c_i використаємо умову $\varphi(x_j) = f(x_j), j = 0, 1, \dots, n$ і складемо систему рівнянь з $(n+1)$ алгебраїчного рівняння.

$$\begin{aligned} \varphi_0(x_0)c_0 + \varphi_1(x_0)c_1 + \dots + \varphi_m(x_0)c_m &= f(x_0) \\ \varphi_0(x_1)c_0 + \varphi_1(x_1)c_1 + \dots + \varphi_m(x_1)c_m &= f(x_1) \\ \dots \dots \dots \dots \dots \dots \dots \dots & \\ \varphi_0(x_n)c_0 + \varphi_1(x_n)c_1 + \dots + \varphi_m(x_n)c_m &= f(x_n) \end{aligned} \quad (5.3)$$

Якщо $m = n$, система рівнянь (3) має єдиний розв'язок за умови, що вектори $\varphi_i(x_j), i, j = 0, 1, \dots, n$ лінійно незалежні. Наведена вище задача називається *задачею інтерполяції*.

Якщо $m = n$ і в якості базисних функцій $\{\varphi(x)\}$ вибрати степеневі функції, то система рівнянь матиме вигляд:

$$\begin{aligned} c_0 + x_0c_1 + x_0^2c_2 + \dots + x_0^nc_n &= f(x_0) \\ c_0 + x_1c_1 + x_1^2c_2 + \dots + x_1^nc_n &= f(x_1) \\ \dots \dots \dots \dots \dots \dots \dots \dots & \\ c_0 + x_nc_1 + x_n^2c_2 + \dots + x_n^nc_n &= f(x_n) \end{aligned} \quad (5.4)$$

Визначник цієї системи, який називають *визначником Вандермонда* [6]:

$$\Delta = \begin{vmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{vmatrix}, \quad (5.5)$$

який не перетворюється в нуль, якщо серед сукупності вузлів немає повторень. У такому випадку матриця коефіцієнтів системи рівнянь є не виродженою і система має єдиний розв'язок.

В якості базисних функцій можемо вибрати також послідовність показникових функцій [9]:

$$\varphi_0(x) = 1, \quad \varphi_1(x) = e^{\alpha_1 x}, \quad \varphi_2(x) = e^{\alpha_2 x}, \quad \dots, \quad \varphi_m(x) = e^{\alpha_m x} \quad (5.6)$$

де $\varphi_i(x)$ – деяка числова послідовність попарно різних дійсних чисел. У такому випадку інтерполяційний поліном запишемо у вигляді:

$$\varphi(x) = c_0 + c_1 e^{\alpha_1 x} + c_2 e^{\alpha_2 x} + \dots + c_m e^{\alpha_m x}. \quad (5.7)$$

5.1. Кусково-лінійна інтерполяція. Інтерполяційні поліноми вищих порядків. Інтерполяційний поліном Лагранжа

Найпростішим прикладом побудови інтерполяційного поліному є *кусово-лінійна інтерполяція*. При такому інтерполюванні будується відрізок прямої, яка проходить через дві сусідні вузлові точки (рис. 5.1).

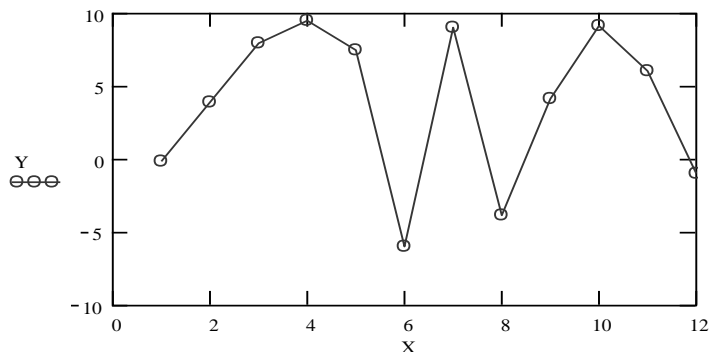


Рис. 5.1. Кусково-лінійна інтерполяція.

Тангенс кута нахилу між цими точками (x_0, y_0) , (x_1, y_1) дорівнює $m = \frac{(y_1 - y_0)}{(x_1 - x_0)}$. Для точки y , що належить проміжку $[y_0, y_1]$, можемо записати:

$$y = P(x) = y_0 + m \cdot (x - x_0) = y_0 + (y_1 - y_0) \frac{(x - x_0)}{(x_1 - x_0)} \quad (5.8)$$

При побудові інтерполяційної функції необхідно, щоб початкова функція $F(x) \approx P(x)$. При класичному розв'язуванні задачі вимагається також строгий збіг значень $F(x)$ та $P(x)$ у вузлах інтерполяції x_i .

Інтерполяційний поліном Лагранжа. Французький математик Жозеф Луї Лагранж (1736-1813 р.р.) запропонував для побудови інтерполяційних поліномів наступний метод. Для двох вузлових точок (x_0, y_0) , (x_1, y_1) він записав [10]:

$$y = P(x) = y_0 \frac{(x - x_1)}{(x_0 - x_1)} + y_1 \frac{(x - x_0)}{(x_1 - x_0)} \quad (5.9)$$

Відношення при y_i називаються коефіцієнтами Лагранжа. Побудуємо коефіцієнти Лагранжа для n точок. Особливість їх побудови полягає у виключенні з ряду різниць, що перемножуються у чисельнику і знаменнику, різниці зі значенням x_i після знаку “-”. Для n точок коефіцієнт Лагранжа запишемо у загальному вигляді так:

$$l_i(x) = \frac{(x - x_0) \cdot (x - x_1) \cdot \dots \cdot (x - x_{i-1}) \cdot (x - x_{i+1}) \cdot \dots \cdot (x - x_{n-1}) \cdot (x - x_n)}{(x_i - x_0) \cdot (x_i - x_1) \cdot \dots \cdot (x_i - x_{i-1}) \cdot (x_i - x_{i+1}) \cdot \dots \cdot (x_i - x_{n-1}) \cdot (x_i - x_n)} \quad (5.10)$$

Інтерполяційний поліном Лагранжа має вигляд:

$$P(x) = L(x) = \sum_{i=0}^n y_i \cdot l_i(x) \quad (5.11)$$

Він дозволяє досить просто побудувати наближення функції по відомим вузловим точкам у випадку невеликої кількості точок. Зі збільшенням числа точок n порядок поліному збільшується відповідно до ступеню $(n-1)$.

Наприклад: таблично задана функція:

n	0	1	2
x	1	3	4
y	12	4	6

Ступінь поліному $n = 2$. Складемо інтерполяційний поліном Лагранжа для функції з трьох вузлових точок:

$$L_2(x) = 12 \frac{(x-3) \cdot (x-4)}{(1-3) \cdot (1-4)} + 4 \frac{(x-1) \cdot (x-4)}{(3-1) \cdot (3-4)} + 6 \frac{(x-1) \cdot (x-3)}{(4-1) \cdot (4-3)} =$$

$$= 2 \cdot (x^2 - 7x + 12) - 2 \cdot (x^2 - 5x + 4) + 2 \cdot (x^2 - 4x + 3) = 2x^2 - 12x + 22$$

Графік, що відображає результати опису функції, представлений на рис. 5.2.

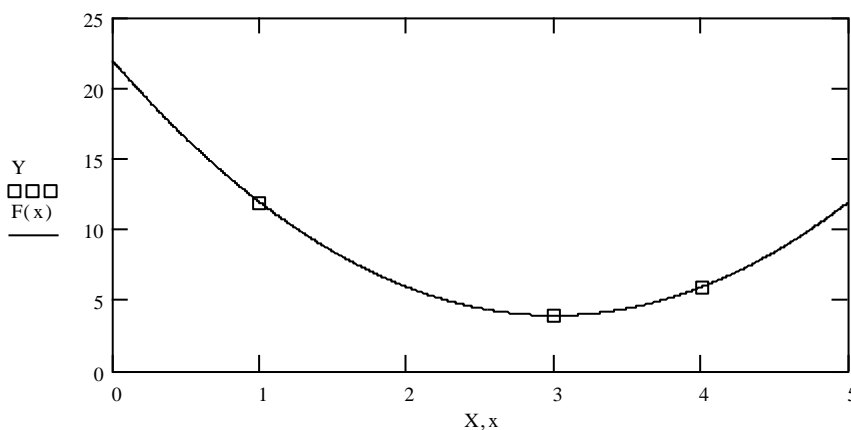


Рис. 5.2. Результати апроксимації поліномом Лагранжа функції, що задана таблицею.

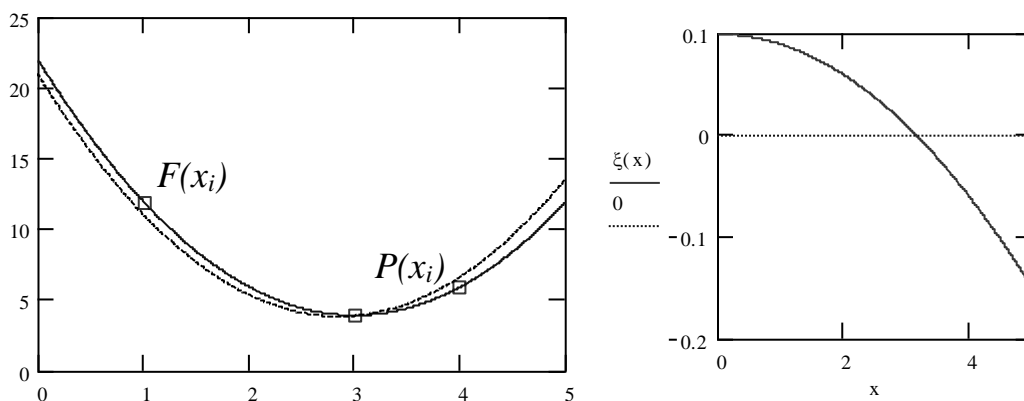


Рис. 5.3. Функція помилки апроксимації функції $F(x)$ поліномом $P(x)$.

Похибку інтерполяційного поліному Лагранжа можна оцінити за формулою [4]:

$$|F(x) - P_n(x)| \leq \frac{\max_{x \in [a,b]} |F^{(n+1)}(x)|}{(n+1)!} |(x-x_0)(x-x_1)\dots(x-x_n)| \quad (5.12)$$

При апроксимації складних функцій одним поліномом, такий поліном описує функцію з досить великими помилками: $\varepsilon = |F(x_j) - P(x_j)|$ де j – номери точок, які не є вузловими. У вузлових точках функція помилки за визначенням має значення нуль $\varepsilon = |F(x_j) - P(x_j)| = 0$.

Тому для опису таких функцій застосовують інші типи опису. Для опису складних функцій можуть застосовуватися інші поліноми (поліном Ньютона, ортогональні поліноми Чебишева, Лежандра, Ерміта та ін.) з рівновіддаленими чи адаптивнопідібраними вузловими точками. Але існує інший метод наближення – інтерполювання сплайнами.

5.2. Сплайн-інтерполяція

При *сплайн-інтерполяції* зберігається умова проходження наближуючої функції через вузлові точки. Сплайн (spline) переводиться з англійської, як гнучка лінійка. Сплайн-функції різняться ступенем поліному, що зображає функцію. Сутністю сплайн-інтерполяції є моделювання фрагментів заданої функції рядом поліномів низького ступеню. Прикладом такого інтерполювання є кусково-лінійне інтерполювання (рис. 5.1), де кожна з функцій будується по двом сусіднім вузловим точкам. Такий же принцип побудови зберігається і для формування поліномів вищих порядків, тільки кількість точок, на основі яких будується кожен з поліномів може бути більшою. Так, наприклад, для кусково-квадратичного поліному на проміжку $[x_0, x_N]$ береться кожен інтервал $[x_{2k}, x_{2k+2}]$ для побудови на ньому квадратичного поліному. Але при побудові такого сплайну кривизна в парних вузлах різко змінюється, що відбивається на якості апроксимації.

Найбільш оптимальною на даний момент є *кусково-кубічна сплайн-інтерполяція* [4, 11-13]. Вона широко застосовується у системах комп'ютерної графіки та системах проектування з використанням комп'ютера: CAD-системах (системи комп'ютери-

зованої допомоги в дизайні та кресленні) та GIS-системах (географічні інформаційні системи).

За допомогою цього виду інтерполяції можливо побудувати гладку криву, що проходить через задані точки.

Означення. Припустимо, задані $(N+1)$ точки, де їх абсциси x_i задовольняють умові $a = x_0 < x_1 < \dots < x_N = b$. Функція $S(x)$ називається кубічним сплайном, якщо існують N кубічних поліномів $S_k(x)$ з коефіцієнтами $s_{k,0}$, $s_{k,1}$, $s_{k,2}$ та $s_{k,3}$, що задовольняють наступним умовам:

- 1) $S(x) = S_k(x) = s_{k,0} + s_{k,1}(x-x_k) + s_{k,2}(x-x_k)^2 + s_{k,3}(x-x_k)^3$, де $x \in [x_k, x_{k+1}]$, $k = 0, 1, \dots, N-1$;
- 2) кусочно-кубічний поліном (сплайн) заданий сукупністю точок $S(x_k) = y_k$, де $k = 0, 1, \dots, N$;
- 3) сплайн-функція складається з поліномів (гладких функцій), які сполучуються у вузлових точках $S_k(x_{k+1}) = S_{k+1}(x_{k+1})$, де $k = 0, 1, \dots, N-2$;
- 4) сплайн-функція буде неперервною, якщо існує і безперервна її перша похідна $S'_k(x_{k+1}) = S'_{k+1}(x_{k+1})$, де $k = 0, 1, \dots, N-2$;
- 5) радіус кривизни сплайн-функції визначений в кожній точці (гладкість функції), якщо існує і безперервна її друга похідна $S''_k(x_{k+1}) = S''_{k+1}(x_{k+1})$, де $k = 0, 1, \dots, N-2$.

Загальний вираз для кубічної сплайн-функції $S_k(x)$ має вигляд:

$$S_k(x) = -\frac{m_k}{6h_k}(x_{k+1} - x)^3 + \frac{m_{k+1}}{6h_k}(x - x_k)^3 + \left(\frac{y_k}{h_k} - \frac{m_k h_k}{6}\right) \cdot (x_{k+1} - x) + \left(\frac{y_{k+1}}{h_k} - \frac{m_{k+1} h_k}{6}\right) \cdot (x - x_k), \quad (5.13)$$

Кубічний поліноми для утворення сплайн-функції будується по трьом сусіднім точкам за загальною формулою:

$$h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} = u_k \quad (5.14)$$

де $h_k = x_{k+1} - x_k$, $m_k = S''(x_k)$, $u_k = 6(d_k - d_{k-1})$, де в свою чергу $d_k = (y_{k+1} - y_k)/h_k$.

При формуванні сплайн-функції будуються $N-1$ рівнянь з $N+1$ змінними. Для їх розв'язання систему треба доповнити двома додатковими рівняннями. В якості таких рівнянь вибирають обмеження в крайніх точках сплайну, що наведені в таблиці 5.1.

Таблиця 5.1

Обмеження в крайніх точках кубічного сплайну [4].

№	Тип сплайну	Обмеження в крайніх точках
1	Зімкнений кубічний сплайн (найкращий, якщо відомі похідні): задається $S'(x_0)$, $S'(x_n)$	$m_0 = \frac{3}{h_0}(d_0 - S'(x_0)) - \frac{m_1}{2}$ $m_N = \frac{3}{h_{N-1}}(S'(x_N) - d_{N-1}) - \frac{m_{N-1}}{2}$
2	Природний кубічний сплайн ("релаксована крива")	$m_0 = 0, m_N = 0$
3	Екстраполяція $S''(x)$ в крайніх обмежуючих точках	$m_0 = m_1 - \frac{h_0(m_2 - m_1)}{h_1}$ $m_N = m_{N-1} - \frac{h_{N-1}(m_{N-1} - m_{N-2})}{h_{N-2}}$
4	$S''(x)$ постійна біля крайніх точок	$m_0 = m_1, m_N = m_{N-1}$
5	Задавання $S''(x)$ в кожній крайній точці	$m_0 = S''(x_0), m_N = S''(x_N)$

Розглянемо побудову зімкненого кубічного сплайну. Якщо відома перша похідна у початковій точці, можна обчислити m_0 і перше рівняння прийме вигляд:

$$2(h_0 + h_1)m_1 + h_1m_2 = u_1 - h_0m_0 \quad (5.15)$$

Аналогічно обчислюємо останнє рівняння:

$$h_{N-2}m_{N-2} + 2(h_{N-2} + h_{N-1})m_{N-1} = u_{N-1} - h_{N-1}m_N \quad (5.16)$$

Отримуємо тридіагональну лінійну систему рівнянь:

$$\begin{bmatrix} b_1 & c_1 & 0 & 0 & 0 \\ a_1 & b_2 & c_2 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & a_{N-3} & b_{N-2} & c_{N-2} \\ 0 & 0 & 0 & a_{N-2} & b_{N-1} \end{bmatrix} \cdot \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_{N-2} \\ m_{N-1} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N-2} \\ v_{N-1} \end{bmatrix} \quad (5.17)$$

Коефіцієнти сплайну обчислюємо за формулами:

$$\begin{aligned}
s_{k,0} &= y_k; & s_{k,1} &= d_k - \frac{h_k(2m_k + m_{k+1})}{6}; \\
s_{k,2} &= \frac{m_k}{2}; & s_{k,3} &= \frac{m_{k+1} - m_k}{6h_k}
\end{aligned}
\tag{5.18}$$

Для зручності обчислень кожен кубічний сплайн доцільно записувати у формі вкладених перемножень:

$$S_k(x) = ((s_{k,3}\omega + s_{k,2})\omega + s_{k,1})\omega + y_k, \tag{5.19}$$

де $\omega = x - x_k$; $x_k \leq x \leq x_{k+1}$. При побудові сплайнів зі складними обмеженнями в крайніх точках, що враховують першу похідну, як, наприклад, в зімкненому кубічному сплайні, отримуємо більш точний результат, ніж при константному заданні значень другої похідної (2 та 4 сплайни в таблиці 5.1). Найкращий результат отримуємо при екстраполяційному заданні других похідних. Це пояснюється адаптивністю даних обмежень відносно вхідних даних, що дозволяє точніше враховувати особливості функції не тільки в середині проміжку інтерполяції, а і біля кінців проміжку, де похибка апроксимації досягає найбільших значень.

У випадку константного задання обмежень частину експериментальних точок на початку і вкінці проміжку «обрізають» для усунення великих похибок, які виникають в цих діапазонах. Тому при плануванні експерименту такі обчислювальні особливості потрібно враховувати.

Приклад 1. Обчислимо зімкнений кубічний сплайн, що проходить через точки $(0, 0)$, $(1, 0.5)$, $(2, 2)$ та $(3, 1.5)$. Перша похідна задовольняє граничним умовам $S'(0) = 0.2$ та $S'(3) = -1$. Обчислимо проміжні величини:

$$\begin{aligned}
h_0 &= h_1 = h_2 = 1; \\
d_0 &= (y_1 - y_0)/h_0 = (0.5 - 0)/1 = 0.5; \\
d_1 &= (y_2 - y_1)/h_1 = (2.0 - 0.5)/1 = 1.5; \\
d_2 &= (y_3 - y_2)/h_2 = (1.5 - 2)/1 = -0.5; \\
u_1 &= 6(d_1 - d_0) = 6(1.5 - 0.5) = 6; \\
u_2 &= 6(d_2 - d_1) = 6(-0.5 - 1.5) = -12.
\end{aligned}$$

Складемо основні рівняння (перше і останнє рівняння складають за формулами з табл. 5.1., інші рівняння – за формулою (5.14)):

$$\left(\frac{3}{2}h_0 + 2h_1\right)m_1 + h_1m_2 = u_1 - 3(d_0 - S'(x_0))$$

$$h_{k-1}m_{k-1} + 2(h_{k-1} + h_k)m_k + h_k m_{k+1} = u_k \quad \text{для } k = 2, 3, \dots, N-1$$

$$h_{N-2}m_{N-2} + \left(2h_{N-2} + \frac{3}{2}h_{N-1}\right)m_{N-1} = u_{N-1} - 3(S'(x_N) - d_{N-1})$$

Підставимо конкретні значення:

$$\left(\frac{3}{2} + 2\right)m_1 + m_2 = 6 - 3(0,5 - 2) \quad \left(\frac{3}{2} + 2\right)m_1 + m_2 = 5.1;$$

$$m_1 + \left(\frac{3}{2} + 2\right)m_2 = -12 - 3(-1 - (-0,5)) \quad m_1 + \left(\frac{3}{2} + 2\right)m_2 = -10.5$$

При спрощенні системи отримаємо:

$$\begin{bmatrix} 3.5 & 1.0 \\ 1.0 & 3.5 \end{bmatrix} \cdot \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} 5.1 \\ -10.5 \end{bmatrix}$$

Обчислимо значення $m_1 = 2.52$ та $m_2 = -3.72$ Скористаємося додатковими рівняннями з таблиці 5.1:

$$m_0 = 3(0.5 - 0.2) - \frac{2.52}{2} = -0.36$$

$$m_3 = 3(-1 + 0.5) - \frac{-3.72}{2} = 0.36$$

Підставимо розраховані значення у формули (5.18), за якими обчислимо коефіцієнти сплайн-функцій. Отримаємо наступні вирази сплайн-функцій:

$$S_0(x) = 0.48x^3 - 0.18x^2 + 0.2x \quad 0 \leq x \leq 1;$$

$$S_1(x) = -1.04(x-1)^3 + 1.26(x-1)^2 + 1.28(x-1) + 0.5 \quad 1 \leq x \leq 2;$$

$$S_2(x) = 0.68(x-2)^3 - 1.86(x-2)^2 + 0.68(x-2) + 2 \quad 2 \leq x \leq 3$$

Графічний розв'язок системи поданий на рис. 5.4.

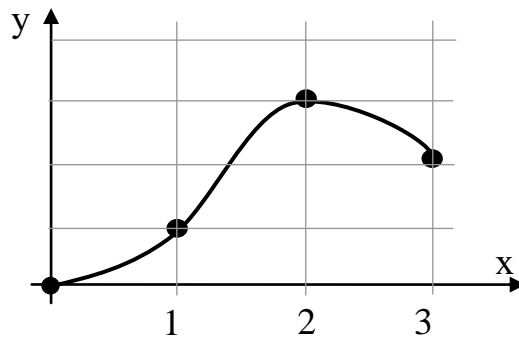


Рис. 5.4. Зімкнений кубічний сплайн.

Приклад 2. При формуванні у середовищі MatLAB [14] зімкненого кубічного сплайну, який проходить через точки $(0, 0.0)$, $(1, 0.5)$, $(2, 2.0)$, $(3, 1.5)$, з першою похідною у граничних точках $S'(0) = 0.2$; $S'(3) = -1$, функція задається так:

```
>> X = [0 1 2 3]; Y = [0 0.5 2.0 1.5]; dx0 = 0.2; dxn = -1;
>> S = zspline (X, Y, dx0, dxn)
S =
    0.4800   -0.1800    0.2000    0
   -1.0400    1.2600    1.2800    0.5000
    0.6800   -1.8600    0.6800    2.0000
```

Графік сплайну будують, використовуючи команду *polyval*, яку потрібно задавати для кожного відрізка сплайну окремо:

```
>> x1 = 0 : .01 : 1; y1 = polyval (S(1,:),x1-X(1));
>> x2 = 1 : .01 : 2; y2 = polyval (S(2,:),x2-X(2));
>> x3 = 2 : .01 : 3; y3 = polyval (S(3,:),x3-X(3));
>> plot (x1,y1,x2,y2,x3,y3,X,Y,')
```

Контрольні питання

1. Що називається інтерполяцією та екстраполяцією?
2. Від яких чинників залежить точність інтерполяції?
3. Який принцип апроксимації функцій поліномом Лагранжа?
4. Що називається кубічною сплайн-функцією? Які умови забезпечують неперервність та гладкість сплайн-функції?
5. Які бувають види кубічних сплайнів в залежності від обмежень в крайніх точках і які з них дозволяють отримати найвищу точність?
6. Чому точність інтерполяції кубічною сплайн-функцією вища ніж поліномом Лагранжа?

Розділ 6. Апроксимація експериментальних даних. Метод найменших квадратів. Інтерполяція функцій за допомогою ортогональних поліномів

6.1. Апроксимація експериментальних даних. Метод найменших квадратів

При проведенні експериментів дослідник отримує дані, які є наближеними через неточність вимірювальних приладів, неточності відтворення умов спостережень, випадкові помилки та інші чинники. Для відновлення функції результатів спостережень або для заміни складної функції простішою для прискорення розрахунків часто використовують *метод найменших квадратів*.

Розглянемо задачу відновлення функції $f(t)$ результатів спостережень методом найменших квадратів [6]. Це задача мінімізації функціоналу. При її розв'язанні необхідно забезпечити мінімум міри відхилень експериментальних значень від обраної функції в $(n+1)$ вузлових точках (x_i, y_i) . Цільову функцію для цієї умови запишемо у вигляді:

$$I = \sum_{i=0}^n (f(x_i) - \varphi(x_i))^2 \rightarrow \min \quad (6.1)$$

де $\varphi(x_i)$ – базисні функції. Найчастіше базисні функції $\varphi(x_i)$ представляють у вигляді поліномів. Якщо функція $\varphi(x_i)$ задовольняє рівнянню 6.1, вона є поліномом найкращого середньоквадратичного наближення:

$$\varphi(x) = c_0 + c_1x + c_2x^2 + \dots + c_nx^k, \quad k < n \quad (6.2)$$

де c_0, c_1, \dots, c_n – коефіцієнти узагальненого полінома. Тобто, побудова полінома найкращого середньоквадратичного наближення для заданої функції $f(t)$ зводиться до знаходження оптимального набору коефіцієнтів для $\varphi(x)$ у рівнянні (6.2). Для розв'язання цієї задачі використаємо метод найменших квадратів.

У випадку, коли $k = n$ (формула 6.2) функція $\varphi(x)$ збігається з інтерполяційним поліномом Лагранжа, побудованим по тих же вузлових точках. Для мінімізації цільової функції 6.1, функціоналу, необхідно, щоб перші похідні функціоналу I за його аргументами

c_0, c_1, \dots, c_n дорівнювали нулю (необхідна умова екстремуму функції [7]):

$$\frac{\partial I}{\partial c_i} = 0 \quad \forall i = 0, 1, 2, \dots, k. \quad (6.3)$$

Обчислимо похідні функціоналу по невідомим:

$$\frac{\partial I}{\partial c_0} = 2 \sum_{i=0}^n (c_0 + c_1 x_i + c_2 x_i^2 + \dots + c_k x_i^k - y_i) \cdot 1 = 0$$

$$\sum_{i=0}^n c_0 + c_1 \sum_{i=0}^n x_i + c_2 \sum_{i=0}^n x_i^2 + \dots + c_k \sum_{i=0}^n x_i^k = \sum_{i=0}^n y_i \quad (6.4)$$

Проведемо заміну: $\bar{x} = \frac{\sum_{i=0}^n x_i}{n+1}$

$$(n+1)c_0 + c_1(n+1)\bar{x} + c_2(n+1)\bar{x}^2 + \dots + c_k(n+1)\bar{x}^k = (n+1)\bar{y}$$

$$c_0 + c_1\bar{x} + c_2\bar{x}^2 + \dots + c_k\bar{x}^k = \bar{y} \quad (6.5)$$

Обчислимо наступні похідні:

$$\frac{\partial I}{\partial c_1} = 2 \sum_{i=0}^n (c_0 + c_1 x_i + c_2 x_i^2 + \dots + c_k x_i^k - y_i) \cdot x_i = 0$$

$$c_0 \sum_{i=0}^n x_i + c_1 \sum_{i=0}^n x_i^2 + c_2 \sum_{i=0}^n x_i^3 + \dots + c_k \sum_{i=0}^n x_i^{k+1} = \sum_{i=0}^n x_i y_i \quad (6.6)$$

Отримаємо рівняння:

$$c_0\bar{x} + c_1\bar{x}^2 + c_2\bar{x}^3 + \dots + c_k\bar{x}^{k+1} = \bar{x}\bar{y}, \quad \text{де} \quad \bar{x}\bar{y} = \frac{\sum_{i=0}^n x_i y_i}{n+1} \quad (6.7)$$

У загальному випадку похідні функціоналу по невідомим:

$$\frac{\partial I}{\partial c_j} = 2 \sum_{i=0}^n (c_0 + c_1 x_i + c_2 x_i^2 + \dots + c_j x_i^j + \dots + c_k x_i^k - y_i) \cdot x_i^j = 0$$

$$\forall j = 2, 3, \dots, k \quad (6.8)$$

Отримаємо систему рівнянь:

$$c_0 \bar{x}^j + c_1 \bar{x}^{j+1} + c_2 \bar{x}^{j+2} + \dots + c_k \bar{x}^{j+k} = \bar{x}^j \bar{y} \quad (6.9)$$

Якщо система має єдиний розв'язок, то він визначає коефіцієнти полінома найкращого середньоквадратичного наближення. Отриманий поліном $\varphi(x) = c_0 + c_1 x + c_2 x^2 + \dots + c_n x^k$ називається **рівнянням регресії**.

Приклад. Запишемо систему рівнянь, складену за методом найменших квадратів для функціонала: $I(a_1, a_2, a_3) = a_1 + a_2 x + a_3 x^2$.

$$\begin{cases} a_1 \cdot (n+1) + a_2 \sum_{i=0}^n x_i + a_3 \sum_{i=0}^n x_i^2 = \sum_{i=0}^n y_i \\ a_1 \sum_{i=0}^n x_i + a_2 \sum_{i=0}^n x_i^2 + a_3 \sum_{i=0}^n x_i^3 = \sum_{i=0}^n (y_i \cdot x_i) \\ a_1 \sum_{i=0}^n x_i^2 + a_2 \sum_{i=0}^n x_i^3 + a_3 \sum_{i=0}^n x_i^4 = \sum_{i=0}^n (y_i \cdot x_i^2) \end{cases} \quad (6.10)$$

Якість апроксимації експериментальних даних поліномом $\varphi(x)$ оцінюється залишковою дисперсією:

$$S^2 = \frac{\sum_{i=0}^n (y_i - \varphi(x_i, a_0, a_1, \dots, a_m))^2}{p} \quad (6.11)$$

де p – число ступенів свободи, яке обчислюється як різниця між числом дослідних значень n і кількістю коефіцієнтів апроксимуючого поліному m ($p = n - m$). Для функцій з різною кількістю коефіцієнтів апроксимуючого поліному значення дисперсії будуть різними, з них вибирають той, який має найменше значення S^2 .

6.2. Інтерполяція функцій ортогональними поліномами

Альтернативою методу найменших квадратів є метод, що оснований на використанні скалярних добутків базисних функцій [5, 11]. Якщо кожне рівняння системи:

$$\begin{aligned}
\varphi_0(x_0)c_0 + \varphi_1(x_0)c_1 + \dots + \varphi_m(x_0)c_m &= f(x_0) \\
\varphi_0(x_1)c_0 + \varphi_1(x_1)c_1 + \dots + \varphi_m(x_1)c_m &= f(x_1) \\
\dots \dots \dots \dots \dots \dots \dots \dots & \\
\varphi_0(x_n)c_0 + \varphi_1(x_n)c_1 + \dots + \varphi_m(x_n)c_m &= f(x_n)
\end{aligned}
\tag{6.12}$$

помножити на відповідну базисну функцію $\varphi_j(x)$, $j = 0, 1, \dots, m$, отримаємо систему рівнянь у *нормальній* формі:

$$\begin{aligned}
(\varphi_0\varphi_0)c_0 + (\varphi_0\varphi_1)c_1 + \dots + (\varphi_0\varphi_m)c_m &= (\varphi_0f) \\
(\varphi_1\varphi_0)c_0 + (\varphi_1\varphi_1)c_1 + \dots + (\varphi_1\varphi_m)c_m &= (\varphi_1f) \\
\dots \dots \dots \dots \dots \dots \dots \dots & \\
(\varphi_m\varphi_0)c_0 + (\varphi_m\varphi_1)c_1 + \dots + (\varphi_m\varphi_m)c_m &= (\varphi_mf)
\end{aligned}
\tag{6.13}$$

Система має $(m+1)$ рівнянь відносно невідомих коефіцієнтів c_j , $j=0, 1, \dots, m$ за значеннями функції f , що визначена на множині точок x_i , $i=0, 1, \dots, n$ для будь-якого $n \geq m$. Збільшення числа вузлових точок впливає лише на точність визначення скалярних добутків і не збільшує розмірності системи рівнянь.

Якщо в якості базисних функцій вибрати ортогональні поліноми (Лежандра, Чебишева, Ерміта та ін.), для яких $(\varphi_i, \varphi_j) = 0$, $i \neq j$, $\|\varphi_i\| \neq 0$, обчислення коефіцієнтів c_j значно спрощується:

$$c_j = \frac{(f, \varphi_j)}{(\varphi_j, \varphi_j)}
\tag{6.14}$$

У цьому випадку похибка ортогональна до всіх $\varphi_j(x)$, $j = 0, 1, \dots, m$, тобто:

$$\sum_{j=0}^m (c_j\varphi_j - f\varphi_k) = 0, \quad k = 0, 1, 2, \dots, m
\tag{6.15}$$

Спільною характеристикою для ортогональних поліномів є форма системи рівнянь:

$$\begin{aligned}
\varphi_0(x) &= c_{00} \\
\varphi_1(x) &= c_{10} + c_{11}x \\
\varphi_2(x) &= c_{20} + c_{21}x + c_{22}x^2 \\
&\dots \dots \dots \dots \dots \dots \dots \\
\varphi_m(x) &= c_{m0} + c_{m1}x + c_{m2}x^2 + \dots + c_{mm}x^m
\end{aligned}
\tag{6.16}$$

У загальному випадку систему можемо записати:

$$\begin{aligned}
\varphi^* &= Ax^* \\
\varphi^*(x) &= [\varphi_0(x), \varphi_1(x), \dots, \varphi_m(x)] \quad x^* = [x^0, x^1, x^2, \dots, x^m]^T
\end{aligned}
\tag{6.17}$$

У даній системі $A = \{a_{ij}\}$, $i, j = 0, 1, 2, \dots, m$ – нижня трикутна матриця порядку $(m+1)$. Розв’язок задачі знайдемо з рівняння:

$$x^* = A^{-1}\varphi^*(x) = B\varphi^*(x)
\tag{6.18}$$

звідки:

$$\begin{cases}
x^0 = b_{00}\varphi_0 \\
x^1 = b_{10}\varphi_0 + b_{11}\varphi_1 \\
x^2 = b_{20}\varphi_0 + b_{21}\varphi_1 + b_{22}\varphi_2 \\
\dots \dots \dots \dots \dots \dots \dots \\
x^m = b_{m0}\varphi_0 + b_{m1}\varphi_1 + b_{m2}\varphi_2 + \dots + b_{mm}\varphi_m
\end{cases}
\tag{6.19}$$

Тобто, будь-який поліном n -го ступеню може бути представлений сукупністю базисних ортогональних поліномів виду:

$$p(x) = c_0\varphi_0(x) + c_1\varphi_1(x) + \dots + c_m\varphi_m(x)
\tag{6.20}$$

Такий алгоритм дозволяє проводити наближення функції ортогональними поліномами за допомогою множення на відповідну базисну функцію (6.13), а потім представляти наближення звичайною поліноміальною формою (6.20). Загальні формули ортогональних поліномів Лежандра, Чебишева і Ерміта наведені в таблиці 6.1.

Апарат кусочно-лінійної інтерполяції (див. розділ 5) дозволяє ввести об’єкти, на яких базується один із найбільш відомих сучасних сіткових чисельних методів — **метод кінцевих елементів** (п. 10.2.2). Досліджувана область розбивається сіткою, в загальному випадку

нерівномірною. Вузлам сітки $\{t_n\}$ ставиться у відповідність набір базисних функцій $\varphi_n(t)$, кожна з яких відповідає певному вузлу t_n , до того ж $\varphi_n(t_k) = \delta_k^n$, $\varphi_n(t_{k-1}) = \varphi_n(t_{k+1}) = 0$, $\varphi_n(t_n) = 1$, а в інших точках вона обчислюється за допомогою кусково-лінійної інтерполяції.

Таблиця 6.1.

Властивості ортогональних поліномів [5]

Властивості	Поліном Лежандра	Поліном Чебишева	Поліном Ерміта
Область визначення	$-1 \leq x \leq 1$	$-1 \leq x \leq 1$	$-\infty \leq x \leq \infty$
φ_0	$P_0(x) = 1$	$T_0(x) = 1$	$H_0(x) = 1$
φ_1	$P_1(x) = x$	$T_1(x) = x$	$H_1(x) = x$
φ_2	$P_2(x) = 1/2(3x^2 - 1)$	$T_2(x) = 2x^2 - 1$	$H_2(x) = 4x^2 - 2$
Рекурентна формула обчислення наступних поліномів	$P_{n+1}(x) = \frac{2n+1}{n+1}xP_n(x) - \frac{n}{n+1}P_{n-1}(x)$	$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$	$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x)$
Загальна формула опису	$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} \cdot [(x^2 - 1)^n]$	$T_n(x) = \cos(n \cdot \arccos(x))$	$H_n(x) = (-1)^n e^{x^2} \cdot \frac{d^n}{dx^n} e^{-x^2}$

Узагальнений інтерполяційний поліном $F(t)$ в даному випадку можемо записати у вигляді:

$$F(t) = \sum_{n=0}^N f(t_n) \cdot \varphi(t_n) \tag{6.21}$$

В обчислювальній математиці часто використовується кусково-поліноміальна інтерполяція. Так, ермітовим кубічним інтерполянтом називається кусково-кубічний інтерполянт з неперервною першою похідною, кубічним сплайном називається кусково-кубічний інтерполянт з двома першими неперервними похідними.

6.3. Базисні сплайни (В-сплайни)

Сплайни з локальним носієм називають базисними сплайнами (В-сплайнами). Останнім часом в обчислювальній практиці широке

розповсюдження набули В-сплайни (від англійського слова bell — дзвін (колокол)), зосереджені на скінченному відрізку [7, 11]. Вони використовуються як для інтерполяції функцій, так і в якості базисних функцій при побудові методів типу скінченних елементів.

В-сплайном (базисним сплайном) ступеню $n-1$ дефекту 1 відносно вузлів $\{t_i\}_{i=n}^{n+N}$ називається функція:

$$B_{N-1,n} = B_{N-1}(t_n, t_{n+1}, \dots, t_{n+N}, t) = N \sum_{i=n}^{n+N} \frac{(t_i - t)_{\max}^{N-1}}{\prod_{\substack{j=n \\ j \neq i}}^{n+1} (t_i - t_j)} \quad (6.22)$$

$$\text{де } (t_i - t)_{\max}^{N-1} = \begin{cases} (t_i - t)^{N-1}, & t \leq t_i \\ 0 & , t > t_i \end{cases}.$$

Припустимо область розв'язку задачі розбита рівномірною сіткою $t_{n+1} = t_n + i \cdot k$. Розглянемо кілька частинних випадків В-сплайнів.

При $N = 2$ **В-сплайн** (1-го порядку) будується найпростіше:

$$\begin{aligned} B_{1,n}(t) = B_1(t_n, t_{n+1}, t_{n+2}, t) &= 2 \left[\frac{(t_n - t)_{\max}}{(t_n - t_{n+1})(t_n - t_{n+2})} + \right. \\ &\left. \frac{(t_{n+1} - t)_{\max}}{(t_{n+1} - t_n)(t_{n+1} - t_{n+2})} + \frac{(t_{n+2} - t)_{\max}}{(t_{n+2} - t_n)(t_{n+2} - t_{n+1})} \right] = \\ &= \frac{1}{\tau^2} [(t_n - t)_{\max} - 2(t_{n+1} - t)_{\max} + (t_{n+2} - t)_{\max}] \end{aligned} \quad (6.23)$$

або

$$B(t) = \begin{cases} \frac{1}{\tau^2} (t_n - t - 2t_{n+1} + 2t + t_{n+2} - t) = 0 & t \leq t_n \\ \frac{1}{\tau^2} (0 - 2t_{n+1} + 2t + t_{n+2} - t) = \frac{1}{\tau} + \frac{t - t_{n+1}}{\tau^2} & t_n \leq t \leq t_{n+1} \\ \frac{1}{\tau^2} (0 - 0 + t_{n+2} - t) = \frac{1}{\tau} - \frac{t - t_{n+1}}{\tau^2} & t_{n+1} \leq t \leq t_{n+2} \\ 0 & t \geq t_{n+2} \end{cases} \quad (6.24)$$

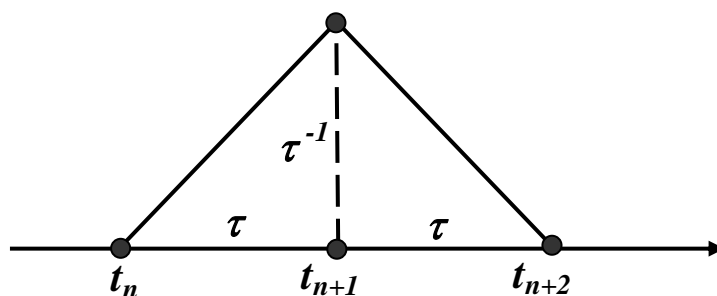


Рис. 6.1. В-сплайн першого порядку.

В-сплайн 2-го порядку запишемо у вигляді:

$$S_k(x) = \begin{cases} x^2 & x = \frac{t - t_{k-2}}{t_{k-1} - t_{k-2}} & t \in [t_{k-2}, t_{k-1}] \\ 1 + 2x - x^2 & x = \frac{t - t_{k-1}}{t_k - t_{k-1}} & t \in [t_{k-1}, t_k] \\ 2 - x^2 & x = \frac{t - t_k}{t_{k+1} - t_k} & t \in [t_k, t_{k+1}] \\ (1 - x)^2 & x = \frac{t - t_{k+1}}{t_{k+2} - t_{k+1}} & t \in [t_{k+1}, t_{k+2}] \end{cases} \quad (6/25)$$

При $t < t_{k-2}$, $t > t_{k+2}$ $S_k(x) \equiv 0$. Побудований сплайн має наступні властивості:

$$\begin{aligned} \text{а) } S'_i(t_{k-2}) &= S'_i(t_{k+2}) = 0, \\ \text{б) } S(t_{k-1}) &= S(t_{k+1}) = 1, \\ \text{в) } S(t_{k-2}) &= S(t_{k+2}) = 0. \end{aligned} \quad (6.26)$$

При інтерполяції функції можемо додати граничні умови. Для інтерполяції за допомогою сплайну необхідно виконання умови:

$$b_{i-1}S_{i-1} + b_iS_i + b_{i+1}S_{i+1} = f_i, \quad (6.27)$$

де b — коефіцієнти інтерполяції, S — В-сплайн, індекс вказує на точку, для якої побудований сплайн і в якій він досягає максимуму. Система таких співвідношень, доповнюється граничними умовами. Отримана система для визначення коефіцієнтів розкладу буде мати тридіагональну матрицю з діагональною перевагою при виконанні обмеження на довжину сусідніх кроків: вони повинні різнитися не більше ніж в $\frac{1 + \sqrt{13}}{2}$ рази.

Кубічний В-сплайн ($N = 4$) має вигляд:

$$B_{3,n}(t) = \frac{1}{6\tau^4} \left[(t_n - t)_{\max}^3 - 4(t_{n+1} - t)_{\max}^3 + 6(t_{n+2} - t)_{\max}^3 - 4(t_{n+3} - t)_{\max}^3 + (t_{n+4} - t)_{\max}^3 \right] \quad (6.29)$$

Після деяких спрощень можемо записати попередню формулу у вигляді:

$$\left\{ \begin{array}{ll} 0 & t < t_n \\ \frac{1}{6\tau^4} (t - t_n)^3 & t_n \leq t < t_{n+1} \\ \frac{1}{6\tau} + \frac{1}{2\tau^2} (t - t_{n+1}) + \frac{1}{2\tau^3} (t - t_{n+1})^2 - \frac{1}{2\tau^4} (t - t_{n+1})^3 & t_{n+1} \leq t < t_{n+2} \\ \frac{1}{6\tau} + \frac{1}{2\tau^2} (t_{n+3} - t) + \frac{1}{2\tau^3} (t_{n+3} - t)^2 - \frac{1}{2\tau^4} (t_{n+3} - t)^3 & t_{n+2} \leq t < t_{n+3} \\ \frac{1}{6\tau^4} (t_{n+4} - t)^3 & t_{n+3} \leq t \leq t_{n+4} \\ 0 & t > t_{n+4} \end{array} \right.$$

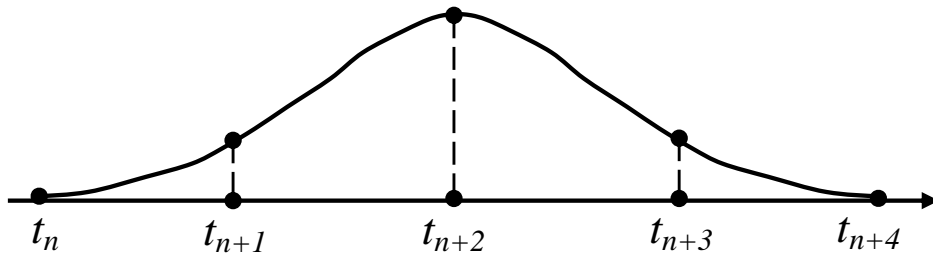


Рис. 6.2. Графік кубічного В-сплайну.

Базисні сплайни (В-сплайни) заданого ступеню є лінійно незалежними функціями і утворюють базис в функціональних просторах, цю властивість можемо використати для представлення за їх допомогою інших функцій цих же просторів. Будь-яка, наприклад, кусково-постійна функція на відрізку, що вміщує інтервали рівної довжини, може бути єдиним чином представлена лінійною комбінацією В-сплайнів нульового ступеню, будь-яка кусково-лінійна функція — В-сплайнів першого ступеню і т.і. Базисні сплайни відіграють значну роль при побудові чисельних методів розв'язання задач математичної фізики, наприклад, метода скінченних елементів в

теорії наближень функцій, при розв'язку задач комп'ютерної графіки та інших [15, 17].

Контрольні питання

1. Для яких задач використовують метод найменших квадратів?
2. В чому полягає суть методу найменших квадратів?
3. Як будується поліном найкращого середньоквадратичного наближення?
4. Яке рівняння називається рівнянням регресії?
5. Як оцінюється якість апроксимації експериментальних даних?
6. Назвіть види ортогональних поліномів?
7. Що називається базисним сплайном (B-сплайном)?
8. Назвіть види базисних сплайнів? Для яких задач вони застосовуються ?

Частина 2. Чисельне диференціювання та інтегрування. Розв'язання диференціальних рівнянь.

Розділ 7. Чисельне диференціювання та чисельне інтегрування

7.1. Чисельне диференціювання

У випадках, коли похідну функції, яка задана аналітично, важко обчислити по причині її складності чи тому, що похідна набуває незручну для застосування форму, використовують чисельне (наближене) диференціювання. Цей метод незамінний, коли функція задана таблично. Найбільш розповсюджені способи чисельного диференціювання – використання скінченних різниць та інтерполяційних поліномів. Розглянемо їх послідовно.

7.1.1. Використання скінченних різниць

Звернемося до класичного визначення похідної. За визначенням похідна функції $f(x)$ є нова функція, що дорівнює при кожному значенні x границі відношення прирощення функції $(f(x + \Delta x) - f(x))$ до відповідного прирощення аргументу Δx , тобто:

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (7.1)$$

Геометричний сенс похідної. Похідна функції в точці x є тангенсом кута нахилу дотичної до графіку функції в цій точці, якщо кут відраховується від позитивного напрямку осі Ox проти годинникової стрілки (рис. 7.1):

$$f'(x) = \operatorname{tg} \alpha .$$

Якщо перейти від нескінченно малих різниць $(\Delta x \rightarrow 0)$ до кінцевих різниць, отримаємо формулу чисельного диференціювання:

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} \quad (7.2)$$

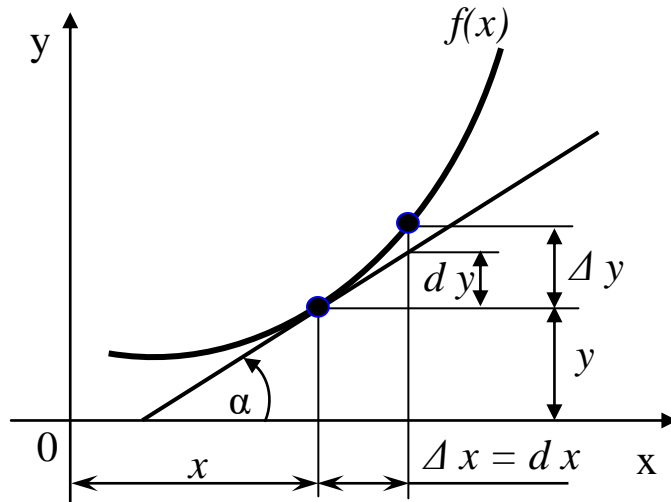


Рис. 7.1. Геометричний сенс похідної функції.

За формулою 7.2 можемо обчислити значення похідної функції $f(x)$ в точці x при заданому значенні прирощення Δx . Заміна нескінченно малих Δx на кінцеві викликає появу похибок, що є похибками методу. Для оцінки цих похибок розкладемо функцію $f(x)$ в точці $(x + \Delta x)$ в ряд Тейлора:

$$f(x + \Delta x) = f(x) + \frac{f'(x)}{1!} \cdot \Delta x + \frac{f''(x)}{2!} \cdot (\Delta x)^2 + \dots + \frac{f^{(n)}(x)}{n!} \cdot (\Delta x)^n + \dots \quad (7.3)$$

При підстановці формули (7.3) у формулу (7.2) отримаємо наступний вираз:

$$f'(x) = f'(x) + \frac{f''(x)}{2!} \cdot \Delta x + \frac{f'''(x)}{3!} \cdot (\Delta x)^2 + \dots \quad (7.4)$$

З отриманої рівності бачимо, що елементи ряду, починаючи з другого, визначають відмінність числового значення похідної від її точного значення. Основним елементом похибки є значення $\frac{f''(x)}{2!} \cdot \Delta x$. Вважають, що така похибка має перший порядок точності по Δx , тобто $O(\Delta x)$.

При обчисленні похідної за симетричною формулою:

$$f'(x) = \frac{f(x + \Delta x) - f(x - \Delta x)}{2 \cdot \Delta x} \quad (7.5)$$

можемо обчислити похибку за тим же методом. Похибка цієї формули дорівнює $\frac{f'''(x)}{3!} \cdot (\Delta x)^2$, вона має другий порядок точності по Δx , тобто $O(\Delta x)^2$.

При аналізі похибок загальну похибку чисельного диференціювання представимо рівнянням :

$$E(f(x), \Delta x) = E_{мет}(f(x), \Delta x) + E_{обч}(f(x), \Delta x) = \frac{f''(x)}{2!} \cdot \Delta x + \frac{e_{(+1)} - e_{(-1)}}{2 \cdot \Delta x} \quad (7.6)$$

$$E(f(x), \Delta x) \leq \frac{M \cdot \Delta x}{2} + \frac{\varepsilon}{\Delta x}$$

де $E_{мет}$ – похибка методу, $E_{обч}$ – похибка обчислень, $e_{(+1)} \leq \varepsilon$, $e_{(-1)} \leq \varepsilon$ і $M = \max_{a \leq x \leq b} \{ |f''(x)| \}$.

Для симетричної формули похибку чисельного диференціювання запишемо у вигляді:

$$\begin{aligned} E_{симетр}(f(x), \Delta x) &= E_{мет}(f(x), \Delta x) + E_{обч}(f(x), \Delta x) = \\ &= \frac{f'''(x)}{3!} \cdot (\Delta x)^2 + \frac{e_{(+1)} - e_{(-1)}}{2 \cdot \Delta x} \end{aligned} \quad (7.7)$$

Якщо $e_{(+1)} \leq \varepsilon$, $e_{(-1)} \leq \varepsilon$ і $M = \max_{a \leq x \leq b} \{ |f'''(x)| \}$, то

$$E(f(x), \Delta x) \leq \frac{M \cdot (\Delta x)^2}{6} + \frac{\varepsilon}{\Delta x} \quad (7.8)$$

Знайдемо значення Δx (крок диференціювання), що мінімізує праву частину рівняння (7.7) з врахуванням симетричності формули диференціювання:

$$\Delta x = \left(\frac{3\varepsilon}{M} \right)^{\frac{1}{3}} \quad (7.9)$$

Використовуючи симетричну формулу четвертого порядку ($O(\Delta x)^4$) для обчислення першої похідної отримаємо:

$$f'(x) \approx \frac{-f(x + 2\Delta x) + 8f(x + \Delta x) - 8f(x - \Delta x) + f(x - 2\Delta x)}{12 \cdot \Delta x} \quad (7.10)$$

У даному прикладі похибку методу запишемо у вигляді:

$$E_{мет}(f(x), \Delta x) = \frac{f^{(5)}(x)}{30} (\Delta x)^4 \quad (7.11)$$

Загальна похибка чисельного диференціювання:

$$\begin{aligned} E(f(x), \Delta x) &= E_{мет}(f(x), \Delta x) + E_{обч}(f(x), \Delta x) = \\ &= \frac{f^{(5)}(x)}{30} \cdot (\Delta x)^4 + \frac{-e_{(+2)} + 8e_{(+1)} - 8e_{(-1)} - e_{(-2)}}{2 \cdot \Delta x} \end{aligned} \quad (7.12)$$

Визначення кроку диференціювання проводимо за формулою:

$$\Delta x = \left(\frac{45\varepsilon}{4M} \right)^{1/5} \quad (7.13)$$

7.1.2. Чисельне диференціювання на основі інтерполяційної формули Лагранжа

Розглянемо поліном Лагранжа з рівновіддаленими вузлами (розділ 5, п. 5.1):

$$P(x) = L(x) = \sum_{i=0}^n y_i \cdot l_i(x),$$

$$\text{де } l_i(x) = \frac{(x-x_0) \cdot (x-x_1) \cdot \dots \cdot (x-x_{i-1}) \cdot (x-x_{i+1}) \cdot \dots \cdot (x-x_{n-1}) \cdot (x-x_n)}{(x_i-x_0) \cdot (x_i-x_1) \cdot \dots \cdot (x_i-x_{i-1}) \cdot (x_i-x_{i+1}) \cdot \dots \cdot (x_i-x_{n-1}) \cdot (x_i-x_n)}.$$

Проведемо заміну: $t = \frac{x-x_0}{h}$. Тоді поліном Лагранжа можемо записати у такому вигляді:

$$P(x) = L(x) = \sum_{i=0}^n \frac{(-1)^{n-i} y_i t^{[n+1]}}{i! (n-1)! t-i} \quad (7.14)$$

При диференціюванні поліному Лагранжа, беручи до уваги, що $dx/dt = h$, отримаємо:

$$f'(x) = L'(x) = \frac{1}{h} \sum_{i=0}^n \frac{(-1)^{n-i} f(x_i)}{i! (n-1)!} \frac{d}{dt} \left(\frac{t^{[n+1]}}{t-i} \right) \quad (7.15)$$

З цієї формули отримуємо деякі загальні формули для правих і лівих різниць порядку ($O(h)^2$):

$$f'(x_0) \approx \frac{-3f_0 + 4f_1 - f_2}{2h} \quad (\text{права різниця}) \quad (7.16)$$

$$f'(x_0) \approx \frac{3f_0 - 4f_{-1} + f_{-2}}{2h} \quad (\text{ліва різниця}) \quad (7.17)$$

Аналогічно отримують різницеві формули для визначення других похідних:

$$f''(x_0) \approx \frac{2f_0 - 5f_1 + 4f_2 - f_3}{h^2} \quad (\text{права різниця}) \quad (7.18)$$

$$f''(x_0) \approx \frac{2f_0 - 5f_{-1} + 4f_{-2} - f_{-3}}{h^2} \quad (\text{ліва різниця}) \quad (7.19)$$

Використовуючи різницеві формули можемо розрахувати значення похідної функції з рівновіддаленими вузлами у вузловій точці по значеннях функції у сусідніх вузлових точках.

7.1.3. Застосування похідних

Якщо величина y залежить від x , то швидкість змінення значень y в залежності від x визначається як $\frac{dy}{dx}$. Наприклад швидкість змінен-

ня тиску p в залежності від висоти h запишемо як $\frac{dp}{dh}$. Якщо мова йде

про швидкість змінення величини в залежності від часу то говорять: «швидкість змінення величини», а слова «в залежності від часу» пропускають. Наприклад швидкість змінення току запишемо $\frac{di}{dt}$.

Для визначення швидкості змінення показників в багатьох задачах фізики, техніки, економіки, соціології застосовується диференціювання. Якщо про диференціювати необхідно функцію, похідну якої складно обчислити, застосовують чисельне диференціювання $\frac{df(x)}{dx}$, суть якого полягає в розбитті області визначення змінної x , де $x \in [a, b]$, на відрізки $a = x_0 < x_1 < \dots < x_n =$

b і обчислення на кожному з них значення похідної за різницевою формулою, наприклад, формулою (7.2) чи більш точними формулами, після чого знаходять суму всіх отриманих значень. Таким чином, отримуємо наближене значення похідної, точність якого залежить від виду функції $f(x)$, відстані між сусідніми вузловими точками, точності формули чисельного диференціювання. Точність чисельного диференціювання оцінюємо за відповідною формулою похибок (формули 7.8, 7.11).

Якщо задана точність ε , з якою необхідно отримати значення похідної $f'(x)$, то обчислюємо величину відстані між сусідніми точками Δx за формулами 7.9, 7.13.

Диференціювання у Matlab. Для отримання значень похідних функції, що задана таблично, використовують внутрішні функції Matlab [11]:

- **diff (X)** – повертає кінцеві різниці, обчислені по суміжним елементам вектора X. Довжина вектора, що повертається функцією diff (X), на одиницю менше ніж довжина вектора X. Якщо X = матриця, то функція повертає матрицю, що вміщує кінцеві елементи, які обчислені по кожному стовпцю;
- **diff (X, n)** – повертає кінцеві різниці n -го порядку;
- **diff (X, n, dim)** – повертає кінцеві різниці n -го порядку по стовпцям ($dim = 1$) чи рядкам матриці.

7.2. Чисельне інтегрування

В наукових та інженерних задачах часто потрібно обчислювати визначені інтеграли. Але не всі вони мають аналітичний розв'язок. Для знаходження таких інтегралів застосовуються методи чисельного інтегрування.

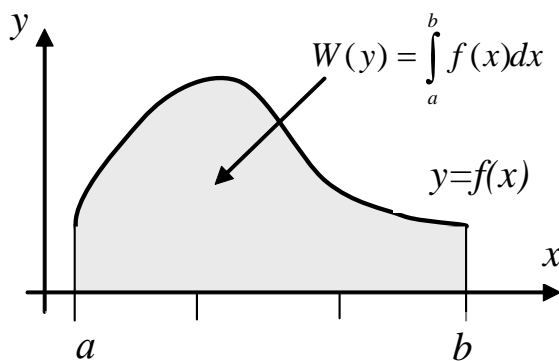


Рис. 7.2. Фізичний сенс інтегралу.

В основу чисельних методів інтегрування закладений фізичний сенс визначеного інтеграла. Визначений інтеграл дорівнює площі фігури, що обмежена неперервною підінтегральною функцією $f(x)$ на проміжку $[a, b]$ (рис. 7.2).

Класично визначений інтеграл обчислюється за формулою Ньютона-Лейбниці:

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

Але ця формула малоприматна для практичного застосування, оскільки клас функцій, для яких первісні $F(a)$ та $F(b)$ можна виразити через елементарні функції дуже вузький. Крім того, часто на практиці підінтегральна функція задається таблицею і саме поняття первісної втрачає сенс. Тому в наш час для обчислення інтегралів велике розповсюдження отримали чисельні методи.

Задача чисельного інтегрування полягає у знаходженні визначеного інтегралу від функції $f(x)$ на інтервалі $[a, b]$, обчислюючи функцію $f(x)$ на скінченій множині підінтервалів (рис. 7.3).

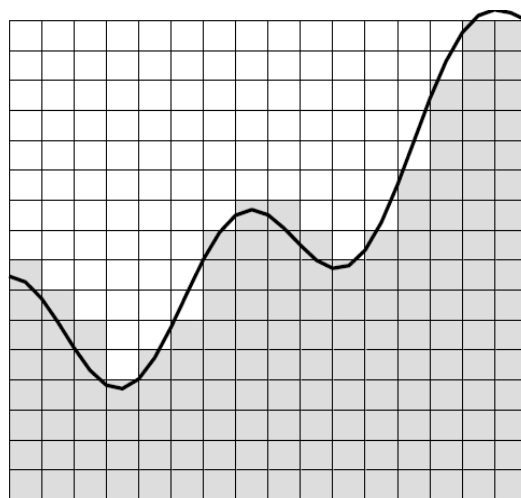


Рис. 7.3. Дискретизація області визначення задачі чисельного інтегрування.

Виберемо проміжні точки так, щоб $a = x_0 < x_1 < \dots < x_M = b$. **Формулою чисельного інтегрування** або **формулою квадратури** називають:

$$Q[f] = \sum_{k=0}^M \omega_k f(x_k) = \omega_0 f(x_0) + \omega_1 f(x_1) + \dots + \omega_M f(x_M) \quad (7.20)$$

яка відповідає визначеному інтегралу

$$\int_a^b f(x)dx = Q[f] - E[f] \quad (7.21)$$

У формулі $E[f]$ – похибка інтегрування. Вона має сенс лише тоді, коли функція $f(x)$ задана аналітично. Функції $\{\omega_k\}_{k=0}^M$ називають ваговими коефіцієнтами. Вузлові точки для чисельного інтегрування можуть вибиратися рівновіддаленими, як у формулах Сімпсона чи Буля, або розташованими за певними правилами, як для формули Гаусса-Лежандра [4, 6].

Ступінь точності формули квадратури дорівнює позитивному цілому числу n , такому що $E[P_i] = 0$ для всіх поліномів $P_i(x)$ ступеня $i \leq n$, але водночас $E[P_{n+1}] \neq 0$ у випадку поліному $P_{n+1}(x)$ ступеня $(n+1)$. Формула для помилки інтегрування має вигляд:

$$E[f] = Kf^{(n+1)}(c) \quad (7.22)$$

де K – константа, n – ступінь точності.

7.2.1. Квадратурні формули інтерполяційного типу

Щоб отримати інтерполяційну формулу для складних функцій нерідко використовують інтерполяційні поліноми, що наближено описують ці функції. За визначенням існує єдиний поліном (зазначеного типу) $P_M(x)$ ступеня $\leq M$, який проходить через $(M+1)$ рівновіддалених точок.

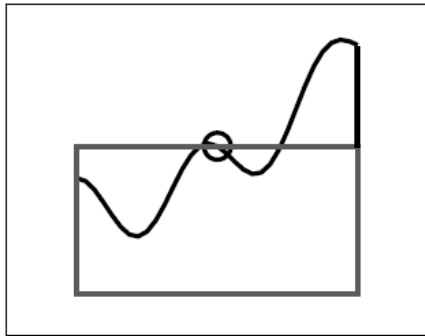
При використанні поліному Лагранжа з рівновіддаленими вузлами для наближення функції $f(x)$ на інтервалі $[a, b]$, інтеграл від $f(x)$ наближено обчислюється за допомогою інтегралу від $P_L(x)$, в результаті отримуємо формулу квадратури Ньютона-Котеса (Newton-Cotes). Коли початкова і кінцева точки $x_0 = a$ та $x_M = b$, формулу називають *замкнутою*. Формули для наближення поліномами $M = 1, 2, 3$ та 4 ступеню відповідно наведені у таблиці 7.1, схеми визначення інтегралу за формулами квадратури подані на рисунку 7.4.

Таблиця 7.1

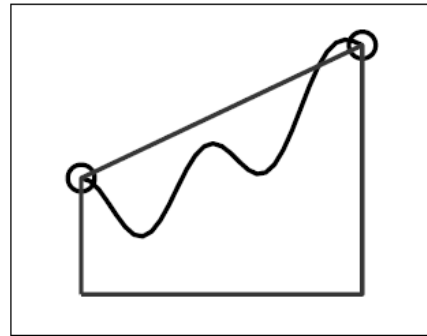
Формули квадратури Ньютона-Котеса [4]

Назва формули	Формула	Похибка формули	Ступінь точності n
Трапецій	$\int_{x_0}^{x_1} f(x)dx \approx \frac{h}{2} \cdot (f_0 + f_1)$	$\frac{h^3}{12} \cdot f^{(2)}(c)$	$n = 1$
Симпсон а	$\int_{x_0}^{x_2} f(x)dx \approx \frac{h}{3} \cdot (f_0 + 4f_1 + f_2)$	$\frac{h^5}{90} \cdot f^{(4)}(c)$	$n = 3$
Симпсон а 3/8	$\int_{x_0}^{x_3} f(x)dx \approx \frac{3h}{8} \cdot (f_0 + 3f_1 + 3f_2 + f_3)$	$\frac{3h^5}{80} \cdot f^{(4)}(c)$	$n = 3$
Буля	$\int_{x_0}^{x_4} f(x)dx \approx \frac{2h}{45} \cdot (7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4)$	$\frac{8h^7}{945} \cdot f^{(6)}(c)$	$n = 5$

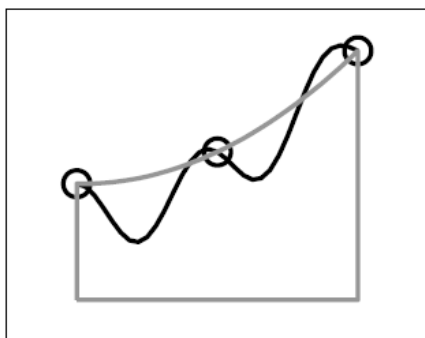
Формула прямокутника



Формула трапецій



Формула Симпсона



Формула Буля

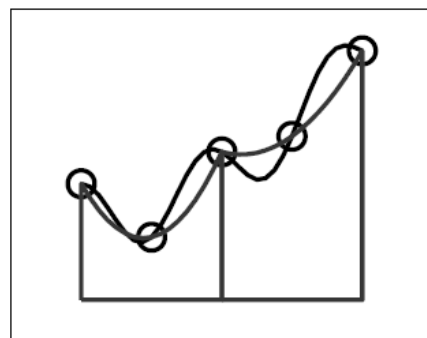


Рис. 7.4. Схеми визначення інтегралу за формулами квадратури.

7.2.2. Замкнуті формули квадратури Ньютона-Котеса

При обчисленні інтегралу можемо отримати проміжних точок більше, ніж використовується у формулі квадратури. Тому для обчислення інтегралу на всьому заданому проміжку скористаємося складною формулою квадратури. Так складна формула трапецій має вигляд:

$$\int_a^b f(x)dx \approx T(f, h) \quad (7.23)$$

$$T(f, h) = \frac{h}{2} \cdot \sum_{k=0}^M (f(x_{k-1}) + f(x_k)) = \frac{h}{2} \cdot (f(a) + f(b)) + h \sum_{k=1}^{M-1} f(x_k) \quad (7.24)$$

Складна формула Сімпсона:

$$\int_a^b f(x)dx \approx S(f, h) \quad (7.25)$$

$$S(f, h) = \frac{h}{3} \cdot \sum_{k=0}^M (f(x_{2k-2}) + 4f(x_{2k-1}) + f(x_{2k})) =$$

$$= \frac{h}{3} \cdot (f(a) + f(b)) + \frac{4h}{3} \sum_{k=1}^M f(x_{2k-1}) + \frac{2h}{3} \sum_{k=1}^{M-1} f(x_{2k}) \quad (7.26)$$

Для того, щоб формула квадратури була замкнутою, потрібно, щоб кількість вузлових точок на проміжку $[a, b]$ була кратною числу точок, на яких будується вибрана формула квадратури з врахуванням початкової та кінцевої точок кожної локальної формули квадратури (рис. 7.5).

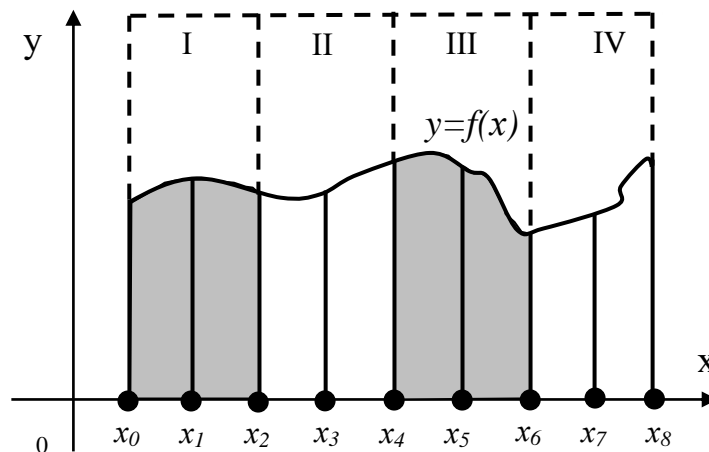


Рис. 7.5. Розрахунок визначеного інтегралу за замкнутою формулою Симпсона.

З рисунку 7.5 бачимо, що для розрахунку визначеного інтегралу на проміжку $[x_0, x_8]$ інтервал розбили на 4 підінтервали, формула Симпсона використовувалася 4 рази. Якщо позначити кількість точок, що використовується в простій формулі Симпсона $n = 3$ і число застосувань формули $m = 4$ (значення функції в точках x_2, x_4, x_6 використовуються двічі), то розрахунок кількості вузлів матиме вигляд: $k = (n-1) \cdot m + 1 = 2 \cdot 4 + 1 = 9$. Аналогічно розраховується кількість точок за іншими формулами квадратури.

Контрольні питання

1. В чому полягає суть чисельного диференціювання?

2. Що називається різничним аналогом похідної? Які існують види різничних аналогів?
3. В чому полягає задача чисельного інтегрування?
4. Що називається формулою квадратури?
5. Які існують формули квадратури Ньютона-Котеса? Як здійснюється оцінка точності формул квадратури Ньютона-Котеса?
6. Що називається замкнутою квадратурною формулою? Як формується замкнута формула квадратури?

Розділ 8. Чисельні методи розв'язання задачі Коші для звичайних диференціальних рівнянь

8.1. Основні поняття

При побудові динамічних математичних моделей технічних і природничих об'єктів часто приходять до розв'язку *звичайного диференціального рівняння* чи *системи диференціальних рівнянь*, що задовольняють певній *початковій умові* (задача Коші). В небагатьох випадках можливо проінтегрувати таке рівняння у скінченному вигляді, але навіть якщо інтегрування можливе, то шукана функція часто входить до розв'язку у неявному вигляді, що не зручно для подальшого використання у математичних розрахунках.

Для розв'язання таких задач використовують *чисельне (наближене) інтегрування* диференціальних рівнянь.

Задача Коші полягає у знаходженні розв'язку $y(x)$ диференціального рівняння

$$\frac{dy}{dx} = f(x, y), \quad (8.1)$$

який задовольняє початковій умові

$$y(x)|_{x=x_0} = y_0 \quad (8.2)$$

При геометричній інтерпретації розв'язку задачі – треба знайти інтегральну криву $y(x)$ рівняння (8.1), яка проходить через точку (x_0, y_0) . При розгляді рис. 8.1 інтегральна крива, що відповідає початковій умові $y(t)|_{t=1} = 1,5$ має коефіцієнт $C = 0$.

Задача Коші має *єдиний розв'язок* при виконанні наступної теореми.

Теорема Пікара: Якщо функція $f(x, y)$ двох змінних x і y неперервна у замкненому прямокутнику з центром у точці (x_0, y_0)

$$\bar{\Delta} = \{(x, y) : |x - x_0| \leq l, |y - y_0| \leq b\}$$

і задовольняє в ньому умові Ліпшиця по змінній y , тобто існує число $K > 0$, яке не залежить від x і y , таке, що

$$|f(x_1, y_1) - f(x_2, y_2)| \leq K|y_1 - y_2| \quad (8.3)$$

для будь-яких точок $(x_1, y_1) \in \bar{\Delta}$ та $(x_2, y_2) \in \bar{\Delta}$, то існує єдина диференційована функція $y = \varphi(x)$, яка є розв'язком диференційного рівняння (8.1), що задовольняє початковій умові (8.2).

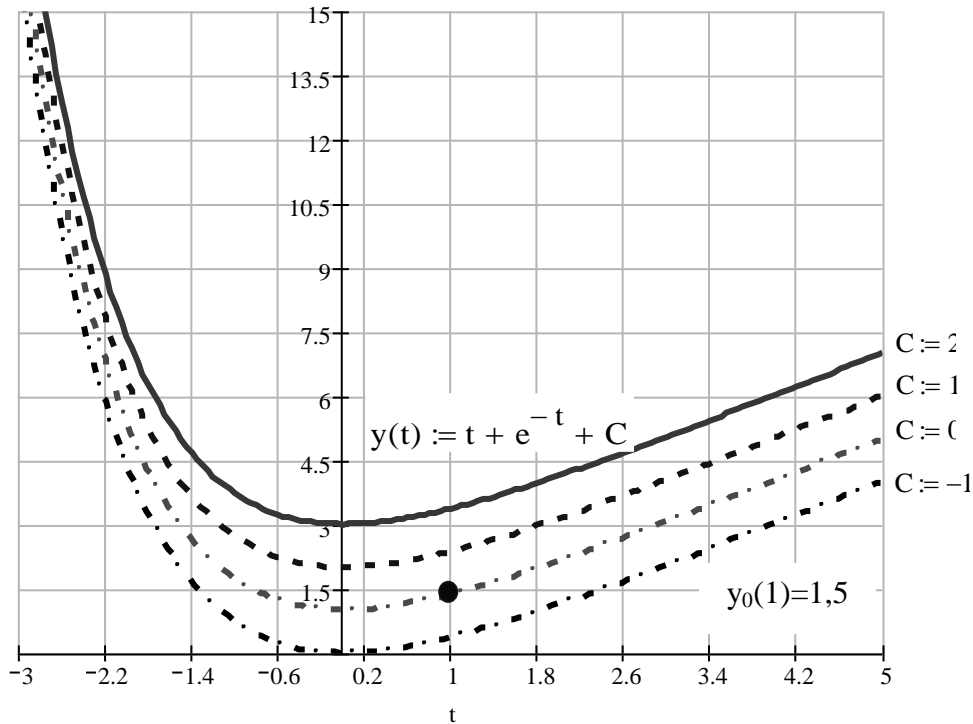


Рис. 8.1. Множина інтегральних кривих, що є розв'язком диференційного рівняння $y(t) = t + e^{-t} + C$

При розгляді, наприклад, фізичної системи y – вектор фазових змінних (розмірністю n), які характеризують запас енергії у системі, x часто є часом перебігу реального процесу. При чисельному розв'язанні рівняння отримують значення $y(x)$ на потрібному інтервалі часу, наприклад $0 \dots t_k$, при заданих початкових умовах (формула 8.2). При розв'язанні задачі на проміжку інтегрування виділяють певне число точок t_m , у яких розраховують значення $y(x)$. Інтервал між сусідніми точками називають кроком інтегрування і позначають:

$$h_m = t_{m+1} - t_m.$$

Знайти послідовність вузлових точок при $h = \text{const}$ дозволяє формула:

$$x_k = x_0 - kh.$$

До методу чисельного розв'язку (інтегрування) диференційних рівнянь пред'являються наступні **основні вимоги**:

- універсальність, тобто можливість застосування до будь-якого об'єкту із заданого класу,
- алгоритмічна надійність,
- достатня точність результатів,
- помірні затрати машинного часу і пам'яті.

Основними характеристиками методів інтегрування, від яких залежить їх ефективність, є **точність** і **стійкість** методів, а також пов'язана з ними стратегія **вибору величини кроку інтегрування**.

Точність інтегрування залежить від таких основних складових:

- ◆ *похибки апроксимації* (похибки методу інтегрування);
- ◆ *похибки обчислень*, що пов'язана з похибками округлення чисел у персональному комп'ютері і заміною неарифметичних операцій та функцій арифметичними;
- ◆ *похибки накопичення*, яка дорівнює повній похибці на попередньому кроці.

Стійкість методів чисельного інтегрування пов'язана з характером змінення накопиченої похибки. Якщо похибка у ході інтегрування значно не зростає зі збільшенням числа кроків, а змінюється у деяких допустимих межах, то використовуваний метод інтегрування буде чисельно стійким. Якщо навіть при невеликих помилках апроксимації чи обчислень накопичена похибка зростає від кроку до кроку, то використовуваний метод інтегрування нестійкий при даних величинах кроків, а результати таких обчислень не несуть потрібної інформації.

Оцінку кроку інтегрування можливо отримати, дослідивши власні значення матриці коефіцієнтів системи рівнянь. У загальному випадку власні значення – комплексні величини, тобто $\lambda_i = \text{Re}(\lambda_i) + j \cdot \text{Im}(\lambda_i)$.

Крок інтегрування знаходиться із співвідношення $h_m < 2 \tau_{min}$, де τ_{min} – мінімальна постійна часу системи. Постійна часу системи знаходиться з виразу:

$$\tau_i = \frac{1}{\text{Re}|\lambda_i|} \quad (8.4)$$

Для розв'язання задачі Коші використовують методи Гюна, Тейлора, Ейлера (явний, неявний та модифікації), а також методи Рунге-Кутта.

8.2. Явний метод Ейлера

У даному методі значення знаходяться по значеннях функції у попередній точці починаючи від x_0 . Наступне значення функції по значеннях у попередній k -тій точці можна знайти за формулою:

$$y(x_{k+1}) = y(x_k) + \int_{x_k}^{x_{k+1}} f(x, y(x)) dx \quad (8.5)$$

Якщо класичними методами розв'язати задачу не вдається, користуються **формулою Ейлера**, в якій інтеграл наближено обчислюють за формулою лівих прямокутників:

$$y_{k+1} = y_k + hf(x_k, y_k) \quad (8.6)$$

Геометричну інтерпретацію методу Ейлера можемо представити таким чином: якщо побудувати дотичну до графіку функції $y(x)$ в точці x_0 , яка описується рівнянням $y = y_0 + f(x_0, y_0)(x - x_0)$, можемо помітити, що при досить малому кроці $h = x_1 - x_0$ формула дотичної: $y_1 = y_0 + f(x_0, y_0)(x_1 - x_0) = y_0 + f(x_0, y_0) \cdot h$. При малих значеннях h обчислена ордината $y_1(x)$ наближається до $y(x)$ в точці x_1 (рис. 8.2) і чергова точка (x_1, y_1) може бути вибрана за початкову точку для розрахунку значення функції $y(x)$ на наступному інтервалі $[x_1, x_2]$.

При побудові на наступних інтервалах дотичної до функції $y(x)$ отримаємо ламану криву, яку називають ламаною Ейлера [5].

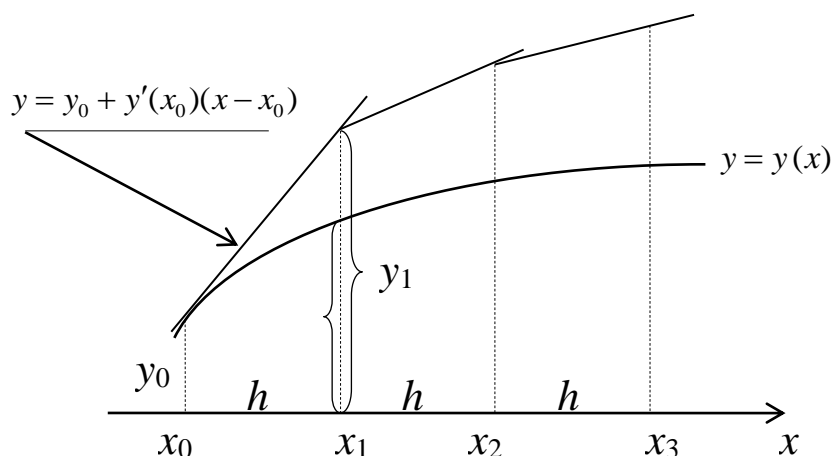


Рис. 8.2. Геометрична інтерпретація методу Ейлера.

Алгоритм явного методу Ейлера:

1. Задавання початкових значень і початкових даних $x_0, x_{кін}, y(x_0) = y_0, \varepsilon_{доп}$ (останній параметр нерідко обчислюється).
2. Обчислення власних значень матриці коефіцієнтів системи.
3. Визначення кроку $h_m < 2 \tau_{min}$.
4. Виконання ітерації за формулою Ейлера (8.6).
5. Перехід до наступної вузлової точки $x_{к+1} = x_k + h$ і вивід результатів.
6. Якщо $x_{к+1} < x_{кін}$, то перейти до пункту 4, інакше завершити ітерації.

Приклад 8.1. 1. Розв'яжемо систему звичайних диференціальних рівнянь:

$$\begin{cases} dx_1/dt = -\frac{5}{6}x_1 + \frac{1}{3}x_2 + \frac{5}{2} \\ dx_2/dt = \frac{1}{3}x_1 - \frac{1}{3}x_2 \end{cases} \quad (8.7)$$

2. Початкові умови $x_{1,0} = x_{2,0} = 0$.
3. Обчислимо розв'язки характеристичного рівняння $\det(A - \lambda I) = 0$.

$$A = \begin{pmatrix} -\frac{5}{6} & \frac{1}{3} \\ \frac{1}{3} & -\frac{1}{3} \end{pmatrix};$$

$$\det(A - \lambda I) = \begin{vmatrix} -\frac{5}{6} - \lambda & \frac{1}{3} \\ \frac{1}{3} & -\frac{1}{3} - \lambda \end{vmatrix} = \begin{vmatrix} -\frac{5}{6} - \lambda & \frac{1}{3} \\ \frac{1}{3} & -\frac{1}{3} - \lambda \end{vmatrix} = 0$$

$$\left(-\frac{5}{6} - \lambda\right)\left(-\frac{1}{3} - \lambda\right) - \frac{1}{3} \cdot \frac{1}{3} = 0; \quad \lambda^2 + \frac{7}{6}\lambda + \frac{3}{18} = 0; \quad \lambda_1 = -1; \quad \lambda_2 = -\frac{1}{6}$$

4. Виберемо крок інтегрування. Постійні часу системи, обчислені за формулою (8.4), $\tau_1 = 1, \tau_2 = 6 \Rightarrow \tau_{min} = 1$. Звідси $h < 2\tau_{min}, h < 2$. Керуючись обмеженням $h < 2$ та необхідною точністю обчислень приймаємо $h = 0,3$.
5. Обчислюємо послідовно значення функції за формулою Ейлера:

1) $x_{1,0} = 0, x_{2,0} = 0$.

$$2) \quad x_{1,1} = 0 + 0,3 \left(-\frac{5}{6} \cdot 0 + \frac{1}{3} \cdot 0 + \frac{5}{2} \right) = 0,75; \quad x_{2,1} = 0 + 0,3 \left(\frac{1}{3} \cdot 0 - \frac{1}{3} \cdot 0 \right) = 0$$

3)

$$x_{1,2} = 0,75 + 0,3 \left(-\frac{5}{6} \cdot 0,75 + \frac{1}{3} \cdot 0 + \frac{5}{2} \right) = 1,3125;$$

$$x_{2,2} = 0 + 0,3 \left(\frac{1}{3} \cdot 0,75 - \frac{1}{3} \cdot 0 \right) = 0,075$$

4) наступні ітерації аналогічно ...

Явний метод Ейлера ефективний тільки для розрахунку систем звичайних диференціальних рівнянь з малим розсіюванням постійних часу, тому що він є обмежено стійким методом інтегрування [3, 4]. Крім того, зустрічаються задачі, де $|\text{Im}(\lambda_i)| \gg |\text{Re}(\lambda_i)|$. У цьому випадку явний метод Ейлера застосовувати недоцільно, тому що він нестійкий в області уявної вісі.

8.3. Неявний метод Ейлера

Формула інтегрування неявного методу Ейлера відрізняється від формули явного методу обчисленням значень у наступній точці по значеннях (x_{k+1}, y_{k+1}) в цій же точці (неявне представлення шуканої величини y_{k+1}) і має вигляд:

$$y_{k+1} = y_k + hf(x_{k+1}, y_{k+1}) \quad (8.8)$$

Особливістю цієї формули інтегрування є необхідність розв'язувати систему рівнянь на кожному кроці інтегрування відносно змінної y_{k+1} , яка присутня в правій і лівій частинах рівняння. Крок у неявному методі Ейлера обмежений тільки точністю інтегрування. Допустиму похибку апроксимації на крок можемо задати, як $\varepsilon_{\text{don}}^i = 0,01 \cdot |y_{\text{max}}^i|$. Після виконання чергового кроку інтегрування розраховується похибка ε_{κ}^i , що повинна бути менше допустимої,

$$\varepsilon_{\kappa}^i = \left| \frac{h_k}{h_k + h_{k+1}} \left((y_{k+1} - y_k) - \frac{h_k}{h_{k-1}} (y_k - y_{k-1}) \right) \right| \quad (8.9)$$

Обчислення за формулою 8.9 починаємо з 4-ї ітерації, оскільки для розрахунку потрібні значення y_{k-1}, y_k, y_{k+1} . Якщо хоча б одне

$|\varepsilon_k^i| > \varepsilon_{\text{дон}}^i$, то крок зменшується і повторюються розрахунки. Якщо ж $|\varepsilon_k^i| < \varepsilon_{\text{дон}}^i$ для всіх i , то крок розраховується за формулою:

$$h_{k+1}^i = \sqrt{\frac{\varepsilon_{\text{дон}}^i}{\varepsilon_k^i}} \cdot h_k, \quad (8.10)$$

Також крок можемо вибирати за методом трьох зон:

$$h_{k+1}^i = \begin{cases} h_k/2, & \text{якщо } |\varepsilon_k^i| > \varepsilon_{\text{дон}}^i \\ h_k, & \text{якщо } \left(\varepsilon_{\text{дон}}^i/m^2\right) < |\varepsilon_k^i| < \varepsilon_{\text{дон}}^i \\ mh_k, & \text{якщо } |\varepsilon_k^i| < \left(\varepsilon_{\text{дон}}^i/m^2\right) \end{cases} \quad (8.11)$$

Остаточню вибирається найменший крок серед всіх значень h^i . У формулі (8.11) часто використовують значення $m = 2$. При жорстких вимогах до точності значення m вибирають більшим.

Приклад 8.2: Розв'яжемо систему звичайних диференціальних рівнянь, задану формулою 8.7.

1. Задаємо початкові дані $x_{\text{поч}}, x_{\text{кін}}, \varepsilon_{\text{дон}}$: $x_{1,0} = x_{2,0} = 0, \varepsilon_{\text{дон}} = 0,1$.
2. Розрахуємо крок інтегрування за формулою 8.10 чи 8.11. Для початкових ітерацій виберемо його таким як і в попередньому прикладі: $h = 0,3$.
3. Розв'яжемо систему відносно x_{k+1} .

$$x_{1,1} - 0 - 0,3 \left(-\frac{5}{6} x_{1,1} + \frac{1}{3} x_{2,1} + \frac{5}{2} \right) = 0 \quad x_{1,1} = 0,604$$

$$x_{2,1} - 0 - 0,3 \left(\frac{1}{3} x_{1,1} - \frac{1}{3} x_{2,1} \right) = 0 \quad x_{2,1} = 0,055$$

4. Виведемо на екран результати.
5. Розрахуємо значення невідомих на перших трьох ітераціях.
6. Після 3-ї ітерації обчислимо ε_k^i (формула 8.9). Якщо умови точності виконуються $|\varepsilon_k^i| < \varepsilon_{\text{дон}}^i$ та $x_{k+1} < x_{\text{кін}}$, перейдемо до наступної ітерації з тим же кроком, тобто до п.3. Інакше, якщо $|\varepsilon_k^i| > \varepsilon_{\text{дон}}^i$ та $x_{k+1} < x_{\text{кін}}$, розрахуємо новий крок – перейдемо до п.2. Якщо $x_{k+1} \geq x_{\text{кін}}$, припиняємо ітерації.

Неявний метод Ейлера може застосовуватися для розв'язання більшості систем звичайних диференціальних рівнянь як з малим, так і з великим розсіюванням постійних часу, але трудомісткість кожного кроку значно вище ніж у попередньому методі. Він абсолютно стійкий в області уявної вісі, але дає більш сильне затухання коливальних процесів у порівнянні з реальними коливаннями, тобто має велику похибку розв'язків [3].

8.4. Модифікації методу Ейлера

Існують різноманітні модифікації методу Ейлера, які спрямовані на уточнення напрямку переходу з точки (x_i, y_i) в точку (x_{i+1}, y_{i+1}) . Розглянемо декілька простих модифікацій явного методу Ейлера [17]. Всі наведені нижче методи є методами другого порядку точності, тобто локальна похибка на кожному кроці має порядок $O(h^3)$. Умовою припинення ітерацій для всіх наведених у даному пункті методів є виконання нерівності:

$$|y_{i+1}^k - y_{i+1}^{k-1}| < \varepsilon \quad (8.12)$$

Виправлений метод Ейлера. Припустимо, що наближене значення $y_i \approx y(x_i)$ є розв'язком $y = y(x)$ задачі Коші й необхідно обчислити $y_{i+1} \approx y(x_{i+1})$, де $x_{i+1} = x_i + h$. Розкладемо розв'язок за формулою Тейлора p -го порядку, приймаючи за базову точку x_i і покладемо в цьому розкладі $x = x_{i+1}$. Отримаємо

$$y(x_{i+1}) = y(x_i) + h y'(x_i) + \frac{1}{2!} h^2 y''(x_i) + \dots + \frac{1}{p!} h^p y^{(p)}(x_i) + O(h^{p+1}). \quad (8.12)$$

Якщо обмежитись двома першими елементами в правій частині розкладу (8.12), то отримаємо звичайний метод Ейлера (8.6).

Нижче розглянемо, що дає врахування третього доданку. При $p = 2$ із (8.12) отримаємо рівняння:

$$y(x_{i+1}) = y(x_i) + h y'(x_i) + \frac{h^2}{2} y''(x_i) + O(h^3) \quad (8.13)$$

Значення першої похідної в точці x_i , приблизно відоме:

$$y'(x_i) = f(x_i, y(x_i)) \approx f(x_i, y_i) \quad (8.14)$$

Диференціюючи рівняння $\frac{dy}{dx} = f(x, y)$ за формулою повної похідної

$$y''(x) = f'_x(x, y) + f'_y(x, y)y'$$

знаходимо наближене значення другої похідної:

$$\begin{aligned} y''(x_i) &= f'_x(x_i, y(x_i)) + f'_y(x_i, y(x_i))f(x_i, y(x_i)) \approx \\ &\approx f'_x(x_i, y_i) + f'_y(x_i, y_i)f(x_i, y_i) \end{aligned} \quad (8.15)$$

Підставляючи наближені вирази $y(x_i)$, $y'(x_i)$ та $y''(x_i)$ в рівність (8.13), отримуємо формулу для обчислення значення функції у наступній точці $y_{i+1} \approx y(x_{i+1})$ при $i = 0, 1, \dots, n$:

$$y_{i+1} = y_i + h \left[f(x_i, y_i) + \frac{h}{2} (f'_x(x_i, y_i) + f'_y(x_i, y_i)f(x_i, y_i)) \right] \quad (8.16)$$

Це формула *виправленого методу Ейлера*.

Оскільки при $i=0$ формули (8.14) і (8.15) є точними, а $y_0 = y(x_0)$, згідно початкової умови задачі Коші, то на першому кроці обчислення за формулою (8.16) буде виникати похибка, пов'язана тільки з усіканням ряду Тейлора. Тобто, локальна похибка методу (8.16) складає величину $O(h^3)$, таким чином *виправлений метод Ейлера* належить до методів другого порядку.

Удосконалений метод Ейлера називають методом середньої точки, оскільки в ітераційних формулах використовується проміжна точка $\left(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}} \right)$, яка ділить відрізок між i -ю та $(i+1)$ -ю точкою навпіл. Цим методом розв'язок задачі Коші обчислюється в два етапи за формулами:

$$\begin{aligned} y_{i+\frac{1}{2}} &= y_i + \frac{h}{2} \cdot f(x_i, y_i), \\ y_{i+1} &= y_i + hf\left(x_i + \frac{h}{2}, y_{i+\frac{1}{2}}\right). \end{aligned} \quad (8.17)$$

Метод Ейлера-Коші (метод Хойна) є одним з варіантів спільного використання методу Ейлера та неявного методу трапецій. В даному методі розрахунки проводяться спочатку за формулою явного методу Ейлера (8.18), а на другому етапі значення функцій усереднюється (формула 8.19):

$$y_{i+1}^* = y_i + hf(x_i, y_i) \quad (8.18)$$

$$y_{i+1} = y_i + h \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^*)}{2} \quad (8.19)$$

Геометрична інтерпретація методу полягає у визначенні напрямку інтегральної кривої в точках (x_i, y_i) та (x_{i+1}, y_{i+1}^*) , в якості остаточного береться середнє значення цих напрямків.

Удосконалений метод Ейлера-Коші з ітераційною обробкою. Можна досягти більшої точності, якщо, виходячи з того ж самого початкового наближення

$$y_{i+1}^0 = y_i + hf(x_i, y_i) \quad (8.20)$$

зробити не одну, а кілька ітерацій (як правило $k = 3 \div 6$) за методом трапецій:

$$y_{i+1}^{(k)} = y_i + h \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{(k-1)})}{2}, \quad k = 1, 2, \dots \quad (8.21)$$

Уточнений метод Ейлера. Для отримання наступної модифікації методу Ейлера, проінтегруємо рівняння $\frac{dy}{dx} = f(x, y)$ на відріжку $[x_{i-1}, x_{i+1}]$:

$$\int_{x_{i-1}}^{x_{i+1}} y'(x) dx = \int_{x_{i-1}}^{x_{i+1}} f(x, y(x)) dx,$$

звідки отримаємо:

$$y(x_{i+1}) = y(x_{i-1}) + \int_{x_{i-1}}^{x_{i+1}} f(x, y(x)) dx \quad (8.22)$$

Застосовуючи до останнього інтегралу односточкову квадратурну формулу середніх прямокутників і замінюючи значення $y(x_{i-1})$ і $y(x_i)$ відомими наближеними значеннями y_{i-1} та y_i відповідно, із (8.22) отримаємо формулу для обчислення наближеного значення $y(x_{i+1})$:

$$y_{i+1} = y_{i-1} + 2hf(x_i, y_i), \quad i = 1, 2, \dots, n-1 \quad (8.23)$$

яку будемо називати *уточненим методом Ейлера*.

Звернемо увагу на одну принципову відмінність методу (8.23) від всіх інших розглянутих до цього моменту методів: уточнений метод є двокроковим. Тут для обчислення значення y_{i+1} використовуються два попередніх значення y_i та y_{i-1} . Двокровість накладає певні обмеження, принаймні, на початок чисельного процесу: значення $y_1 \approx y(x_1)$ не може бути знайдене безпосередньо за цим методом. Тому недостатню для процесу (8.23) другу початкову точку треба отримувати іншим методом, наприклад, явним методом Ейлера. Так, необхідне значення y_1 можна знайти за формулами (8.6) чи (8.17), а потім продовжувати обрахунки за формулою (8.23).

8.5. Метод Гюна

Цей метод використовує фундаментальну теорему аналізу, тобто інтегрування функції $y'(x)$ на інтервалі $[x_k; x_{k+1}]$:

$$\int_{x_k}^{x_{k+1}} f(x, y(x)) dx = \int_{x_k}^{x_{k+1}} y'(x) dx = y(x_{k+1}) - y(x_k) \quad (8.24)$$

Якщо з формули (8.24) виразити $y(x_{k+1})$, то отримаємо формулу (8.22):

$$y(x_{k+1}) = y(x_k) + \int_{x_k}^{x_{k+1}} f(x, y(x)) dx$$

Для наближеного обчислення визначеного інтегралу використаємо метод трапецій з кроком $h = x_1 - x_0$:

$$y(x_{k+1}) \approx y(x_k) + \frac{h}{2} (f(x_k, y(x_k)) + f(x_{k+1}, y(x_{k+1}))) \quad (8.25)$$

Оскільки рівняння неявне – у правій і лівій частинах є значення $f(x_{k+1}, y(x_{k+1}))$ – для перетворення його у явний вигляд скористаємося формулою Ейлера для $y(x_{k+1})$, що знаходиться у правій частині рівняння:

$$y(x_{k+1}) \approx y(x_k) + \frac{h}{2}(f(x_k, y(x_k)) + f(x_{k+1}, y(x_k) + hf(x_k, y(x_k)))) \quad (8.26)$$

Для обчислення послідовності точок, що є розв'язком задачі, використовуємо перше наближення за методом Ейлера, позначивши вираз для заміненого $y(x_{k+1})$ через p_{k+1} :

$$p_{k+1} = y(x_k) + hf(x_k, y(x_k))$$

$$y(x_{k+1}) \approx y(x_k) + \frac{h}{2}(f(x_k, y(x_k)) + f(x_{k+1}, p_{k+1})) \quad (8.27)$$

При визначенні похибки остаточний елемент у формулі трапецій, що використовується для наближення інтегралу з формули (8.22), дорівнює

$$E_{мет}^i = -y''(c_k) \frac{h^3}{12} \quad (8.28)$$

Якщо ми обрахуємо на кожному кроці лише цю похибку, то накопичена похибка на m кроках:

$$E_{\Sigma мет} = -\sum_{k=1}^m y''(c_k) \frac{h^3}{12} \approx \frac{b-a}{12} \cdot y''(c_k) h^2 = O(h^2) \quad (8.29)$$

Тобто при зменшенні кроку вдвічі, похибка повинна зменшитися в 4 рази.

8.6. Метод рядів Тейлора

Цей метод є еталонним методом, за яким порівнюють точність різних чисельних методів при розв'язанні задачі Коші.

Теорема Тейлора: Припустимо, що $y(x) \in C^{n+1}[x_0, x_n]$ і що функцію $y(x)$ можна розкласти в ряд Тейлора порядку n у околі фіксованої точки $x = x_k \in [x_0, x_n]$:

$$y(x_k + h) = y(x_k) + hT_n(x_k, y(x_k)) + O(h^{n+1}) \quad (8.30)$$

де інтеграл наближено представлений рядом Тейлора:

$$T_n(x_k, y(x_k)) = \sum_{j=1}^n \frac{y^{(j)}(x_k)}{j!} h^{j-1} \quad (8.31)$$

У формулі 8.31 $y^{(j)}(x_k) = f^{(j-1)}(x_k, y(x_k))$ є $(j-1)$ -ю похідною по (x_k) функції f . Отримаємо формулу для похідної за допомогою рекурентної процедури:

$$\begin{aligned} y'(x) &= f, \\ y''(x) &= f_x + f_y y' = f_x + f_y f, \\ y^{(3)}(x) &= f_{xx} + 2f_{xy} y' + f_y y'' + f_{yy} (y')^2 = \\ &= f_{xx} + 2f_{xy} f + f_y (f_x + f_y f) + f_y f^2 \end{aligned} \quad (8.32)$$

де f_x, f_y є частинними похідними по відповідним змінним.

У загальному виді:

$$y^{(n)}(x) = P^{(n-1)} f(x, y(x)) \quad (8.33)$$

де $P = \left(\frac{\partial}{\partial x} + f \frac{\partial}{\partial y} \right)$ – диференціальний оператор. При розв'язанні

задачі Коші ітерація Тейлора має вигляд:

$$y_{k+1} = y_k + d_1 h + \frac{d_2 h^2}{2!} + \frac{d_3 h^3}{3!} + \dots + \frac{d_n h^n}{n!} \quad (8.34)$$

де $d_j = y^{(j)}(x_k)$, $j=1,2,\dots,n$. Остаточна похибка методу Тейлора має порядок $O(h^{n+1})$. Точність методу Тейлора залежить від кількості елементів ряду. Для $n=4$ і для кроків h та $h/2$ похибка обчислень:

$$\begin{aligned} E(y(b), h) &\approx Ch^4, \\ E(y(b), \frac{h}{2}) &\approx C \frac{h^4}{16} = \frac{1}{16} Ch^4 \approx \frac{1}{16} E(y(b), h) \end{aligned} \quad (8.35)$$

8.7. Методи Рунге-Кутта

Ці методи виводяться з відповідних методів Тейлора таким чином, щоб остаточна загальна помилка методів не перевищувала

порядку $O(h^N)$. Методи були виведені для того, щоб на кожному кроці виключити необхідність підрахунку похідних вищих порядків.

Методи Рунге-Кутта будуються для будь-яких порядків N , що обумовлені точністю. Найчастіше застосовується метод Рунге-Кутта 4-го порядку ($N=4$). Він дає досить малу помилку при невеликій кількості розрахунків і не складний для програмування. Метод оснований на обчисленні наступного значення за попереднім по формулі:

$$y_{k+1} = y_k + w_1 k_1 + w_2 k_2 + w_3 k_3 + w_4 k_4 \quad (8.36)$$

де k_1, k_2, k_3, k_4 – коефіцієнти, що розраховуються за формулами:

$$\begin{aligned} k_1 &= hf(x_k, y_k) \\ k_2 &= hf(x_k + a_1 h, y_k + b_1 k_1) \\ k_3 &= hf(x_k + a_2 h, y_k + b_2 k_1 + b_3 k_2) \\ k_4 &= hf(x_k + a_3 h, y_k + b_4 k_1 + b_5 k_2 + b_6 k_3) \end{aligned} \quad (8.37)$$

Рунге і Кутт отримали систему рівнянь, що має локальну похибку відсікання порядку $O(h^5)$:

$$\left\{ \begin{aligned} b_1 &= a_1 \\ b_2 + b_3 &= a_2 \\ b_4 + b_5 + b_6 &= a_3 \\ w_1 + w_2 + w_3 + w_4 &= 1 \\ w_2 a_1 + w_3 a_2 + w a_3 &= \frac{1}{2} \\ w_2 a_1^2 + w_3 a_2^2 + w a_3^2 &= \frac{1}{3} \\ w_2 a_1^3 + w_3 a_2^3 + w a_3^3 &= \frac{1}{4} \\ w_3 a_1 b_3 + w_4 (a_1 b_5 + a_2 b_6) &= \frac{1}{6} \\ w_3 a_1 a_2 b_3 + w_4 a_3 (a_1 b_5 + a_2 b_6) &= \frac{1}{8} \\ w_3 a_1^2 b_3 + w_4 (a_1^2 b_5 + a_2^2 b_6) &= \frac{1}{12} \\ w_4 a_1 b_3 b_6 &= \frac{1}{24} \end{aligned} \right. \quad (8.38)$$

Отримані 11 рівнянь вміщують 13 змінних, тому для їх розв'язання потрібні 2 додаткові умови. Такими умовами найчастіше є:

$$a_1 = \frac{1}{2} \quad \text{та} \quad b_2 = 0 \quad (8.39)$$

За таких умов розв'язками системи рівнянь будуть:

$$a_2 = \frac{1}{2}, \quad a_3 = 1, \quad b_1 = \frac{1}{2}, \quad b_3 = \frac{1}{2}, \quad b_4 = 0, \quad b_5 = 0, \quad b_6 = 1,$$

$$w_1 = \frac{1}{6}, \quad w_2 = \frac{1}{3}, \quad w_3 = \frac{1}{3}, \quad w_4 = \frac{1}{6}$$

Підставляючи значення (8.38) та (8.39) в початкове рівняння (8.36) отримаємо формулу для стандартного методу Рунге-Кутта 4-го порядку:

$$y_{k+1} = y_k + \frac{h(f_1 + 2f_2 + 2f_3 + f_4)}{6}, \quad (8.40)$$

де

$$f_1 = f(x_k, y_k) \qquad f_2 = f\left(x_k + \frac{h}{2}, y_k + \frac{h}{2}f_1\right)$$

$$f_3 = f\left(x_k + \frac{h}{2}, y_k + \frac{h}{2}f_2\right) \qquad f_4 = f(x_k + h, y_k + hf_3)$$

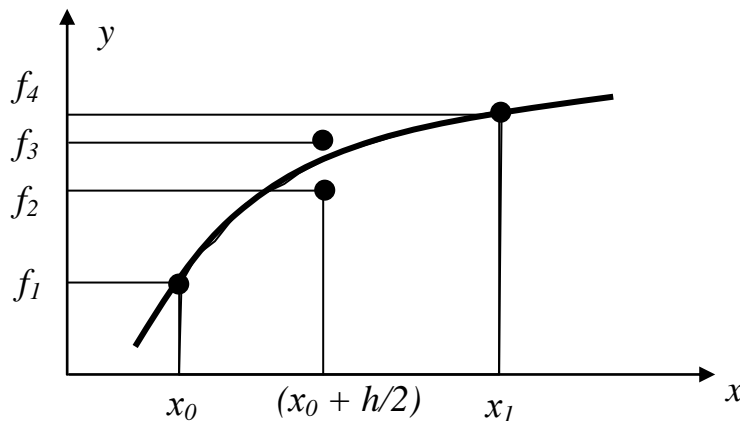


Рис. 8.3. Наближення інтеграла за методом Рунге-Кутта 4-го порядку.

Накопичена похибка в методі Рунге-Кутта складається з елементарних похибок, що утворюються на кожному кроці. Дана похибка методу розраховується за формулою :

$$\Delta_M = -\sum_{k=1}^M y^{(4)}(c_k) \frac{h^5}{2880} \approx \frac{b-a}{5760} \cdot y^{(4)}(c) h^4 \approx O(h^4). \quad (8.41)$$

Приклад 8.3. Задана функція $y' = (x-y)/2$ на інтервалі $[0; 3]$ з початковою умовою $y(0)=1$. Виберемо крок $h = 0,25$. На першій ітерації підрахуємо значення коефіцієнтів f_1, f_2, f_3, f_4 і чергове значення y_1 :

$$f_1 = \frac{x_0 - y_0}{2} = \frac{0-1}{2} = -0,5$$

$$f_2 = \frac{\left(x_0 + \frac{h}{2}\right) - \left(y_0 + \frac{h}{2} \cdot f_1\right)}{2} = \frac{\left(0 + \frac{0,25}{2}\right) - \left(1 + \frac{0,25}{2}(-0,5)\right)}{2} = -0,40625$$

$$f_3 = \frac{\left(x_0 + \frac{h}{2}\right) - \left(y_0 + \frac{h}{2} \cdot f_2\right)}{2} = \frac{\left(0 + \frac{0,25}{2}\right) - \left(1 + \frac{0,25}{2}(-0,40625)\right)}{2} = -0,41211$$

$$f_4 = \frac{(x_0 + h) - (y_0 + h \cdot f_3)}{2} = \frac{(0 + 0,25) - (1 + 0,25(-0,41211))}{2} = -0,3234863$$

$$y_1 = 1 + 0,25 \left(\frac{-0,5 + 2(-0,40625) + 2(-0,41211) - 0,3234863}{6} \right) = 0,89749$$

Всі інші значення на інтервалі $[0; 3]$ розраховуються аналогічно, не застосовуючи операцію диференціювання.

Контрольні питання

1. Що є розв'язком звичайного диференційного рівняння?
2. Що називається задачею Коші?
3. За яких умов задача Коші має єдиний розв'язок?
4. Які основні вимоги пред'являються до чисельних методів розв'язання диференційних рівнянь?
5. Від чого залежить точність розв'язання диференційного рівняння? Порівняйте ці складові з переліком похибок чисельного методу з п. 1.3.
6. Від яких критеріїв залежить розрахунок кроку інтегрування?
7. В яких випадках застосовують адаптивний вибір кроку інтегрування?
8. В чому полягає суть явного методу Ейлера?
9. В яких випадках застосовується неявний метод Ейлера і чому?
- 10.3 якою метою розроблені модифікації явного методу Ейлера?

11. За рахунок чого модифікації методу Ейлера дозволяють отримати більш точний результат, ніж при застосуванні явного методу Ейлера?
12. Який метод розв'язання задачі Коші є базовим для оцінки похибки і чому?
13. В чому полягає суть методу Гюна?
14. В чому полягає перевага методів Рунге-Кутта при розв'язанні звичайних диференціальних рівнянь?
15. В чому полягає суть алгоритму розв'язання диференціальних рівнянь методом Рунге-Кутта 4-го порядку?

Розділ 9. Багатокрокові методи розв'язання диференціальних рівнянь в частинних похідних

Методи Ейлера, Гюна, Тейлора, Рунге-Кута називаються однокроковими методами, тому що в них використовується лише інформація про значення диференціального рівняння в одній попередній точці для того, щоб обрахувати значення в наступній, тобто лише значення в початковій точці $(t_0; y_0)$ використовується для обчислення значення диференціального рівняння в точці $(t_1; y_1)$, і загалом, щоб обрахувати y_{k+1} необхідне лише y_k . Але після знаходження значень диференціального рівняння в декількох точках можна їх використовувати для отримання більш точних результатів у наступній вузловій точці. Наприклад, для чотирикрокового методу Адамса-Бешфорса-Маултона необхідні значення в точках $y_{k-3}, y_{k-2}, y_{k-1}$, та y_k для обчислення y_{k+1} . У цьому методі значення диференціального рівняння в чотирьох початкових точках, $(t_0, y_0), (t_1, y_1), (t_2, y_2)$ та (t_3, y_3) , розраховуються спочатку для подальшої генерації точок $\{(t_k, y_k) : k \geq 4\}$ з використанням багатокрокового методу.

Корисною властивістю багатокрокового методу є можливість визначити локальну помилку відсікання (ЛПВ) та ввести корегуючий елемент, який підвищує точність відповіді на кожному кроці. Також можна визначити, чи буде довжина кроку достатньою, щоб отримати значення y_{k+1} із заданим рівнем точності, або задати більший крок, який виключить непотрібні обрахування і забезпечить необхідну точність розрахунків.

9.1. Метод Адамса-Бешфорса-Маултона

Метод прогнозу-корекції Адамса-Бешфорса-Маултона – це багатокроковий метод, виведений із фундаментальної теореми аналізу:

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} f(t, y(t)) dt. \quad (9.1)$$

Прогноз використовує наближення поліномом Лагранжа для функції $f(t, y(t))$, що побудована по точках $(t_{k-3}, y_{k-3}), (t_{k-2}, y_{k-2}), (t_{k-1}, y_{k-1})$ і (t_k, y_k) . Для функції $f(t, y(t))$ з формули (9.1), яка інтегрується на інтервалі $[t_k, t_{k+1}]$ прогноз Адамса-Бешфорса, має вигляд

$$p_{k+1} = y_k + \frac{h}{24}(-9f_{k-3} + 37f_{k-2} - 59f_{k-1} + 55f_k). \quad (9.2)$$

Як будь-який прогноз, він потребує корекції. Коректор отримується аналогічно. А саме, як тільки значення p_{k+1} обчислено, його використовують для побудови наступного поліному Лагранжа для функції $f(t, y(t))$, який будується по точкам $(t_{k-2}; y_{k-2})$, $(t_{k-1}; y_{k-1})$, $(t_k; y_k)$ і новій точці $(t_{k+1}; y_{k+1})$, в якій замість y_{k+1} використовується прогноз p_{k+1} , тобто функція в точці t_{k+1} ; обчислюється як $f(t_{k+1}, p_{k+1})$. Після розрахунку поліному шляхом інтегрування на інтервалі $[t_k; t_{k+1}]$ отримуємо коректор Адамса-Маултона:

$$y_{k+1} = y_k + \frac{h}{24}(f_{k-2} - 5f_{k-1} + 19f_k + 9f(t_{k+1}, p_{k+1})). \quad (9.3)$$

Оцінка помилки і корекція. Залишковий елемент формули чисельного інтегрування використовується, щоб отримати і прогноз, і коректор порядку $O(h^5)$. Локальна помилка відсікання для формул (9.2) та (9.3) має вигляд :

$$y(t_{k+1}) - p_{k+1} = \frac{251}{720} y^{(5)}(c_{k+1}) h^5, \quad (\text{локальна похибка відсікання для прогнозу}) \quad (9.4)$$

$$y(t_{k+1}) - y_{k+1} = \frac{-19}{720} y^{(5)}(d_{k+1}) h^5. \quad (\text{локальна похибка відсікання для коректора}) \quad (9.5)$$

Припустимо, що h мале і $y^{(5)}(t)$ – майже усталена на інтервалі. Тоді можна виключити члени, що містять похідну 5-го порядку у формулах (9.4) та (9.5). В результаті отримуємо:

$$y(t_{k+1}) - y_{k+1} \approx \frac{-19}{270} (y_{k+1} - p_{k+1}). \quad (9.6)$$

Переваги методу прогнозу-корекції Адамса-Бешфорса-Маултона у тому, що формула (9.6) дає наближену оцінку помилки, яку отримуємо при обчисленні значень p_{k+1} і y_{k+1} , не використовуючи $y^{(5)}(t)$.

Коректор (9.3) використовує наближення $f_{k+1} \approx f(t_{k+1}, p_{k+1})$ при обчисленні y_{k+1} . Оскільки y_{k+1} також є оцінкою для $y(t_{k+1})$, її можна використовувати у коректорі (9.3) для генерування нового наближення для f_{k+1} , яке, в свою чергу, буде генерувати нове значення для y_{k+1} . Окрім того, якщо продовжити цю ітерацію у коректорі, вона

буде збігатись до фіксованої точки у формулі (9.3) швидше, ніж при класичному розв'язку диференційного рівняння. Якщо необхідно підвищити точність, то процедура корекції більш ефективна, ніж зменшення довжини кроку.

Формулу (9.6.) можна використовувати для покращення значень прогнозу. Якщо припустити, що різниця між значеннями прогнозу і корекції на кожному кроці змінюється повільно, то можна підставити P_k і Y_k замість P_{k+1} і Y_{k+1} в (9.6) і отримати наступний управляючий параметр:

$$m_{k+1} = p_{k+1} - 19 \frac{y_k - P_k}{270} \quad (9.7)$$

Його значення використовується замість P_{k+1} на кроці корекції, і формула (9.3) приймає вигляд:

$$y_{k+1} = y_k + \frac{h}{24} (f_{k-2} - 5f_{k-1} + 19f_k + 9f(t_{k+1}, m_{k+1})) \quad (9.8)$$

Формулу (9.6) також можна використовувати для визначення адаптивної довжини кроку. Можливо зменшити довжину кроку, наприклад, до $h/2$ і збільшити до $2h$. Припустимо $\varepsilon = 5 \cdot 10^{-6}$ – критерій відносної похибки, і нехай $k\varepsilon = 10^{-5}$

$$\text{Якщо} \quad \frac{19}{270} \frac{|y_{k+1} - p_{k+1}|}{|y_{k+1}| + k\varepsilon} > \varepsilon, \quad \text{тоді } h=h/2 \quad (9.9)$$

$$\text{Якщо} \quad \frac{19}{270} \frac{|y_{k+1} - p_{k+1}|}{|y_{k+1}| + k\varepsilon} < \frac{\varepsilon}{100}, \quad \text{тоді } h=2h \quad (9.10)$$

Якщо прогноз і корекція значень не дають відповідностей розв'язків у п'ять значущих цифр, то за формулою (9.9) зменшують довжину кроку. Якщо ж розв'язки збігаються до семи або більше значущих цифр, то за формулою (9.10) довжина кроку збільшується.

Зменшення довжини кроку потребує чотирьох нових початкових значень. Для заміщення значень, яких не вистачає і які ділять інтервали $[t_{k-2}, t_{k-1}]$ і $[t_{k-1}, t_k]$ навпіл, використаємо інтерполяцію функції $f(t, y(t))$ поліномом четвертого ступеню. Нові чотири вузлові точки, $t_{k-3/2}, t_{k-1}, t_{k-1/2}$ і t_k , використовуються в наступних розрахунках.

Інтерполяційна формула, необхідна для отримання нових початкових значень (рис. 9.1) для кроку $h/2$ має вигляд :

$$f_{k-1/2} = \frac{-5f_{k-4} + 28f_{k-3} - 70f_{k-2} + 140f_{k-1} + 35f_k}{128},$$

$$f_{k-3/2} = \frac{3f_{k-4} - 20f_{k-3} + 90f_{k-2} + 60f_{k-1} - 5f_k}{128}. \quad (9.11)$$

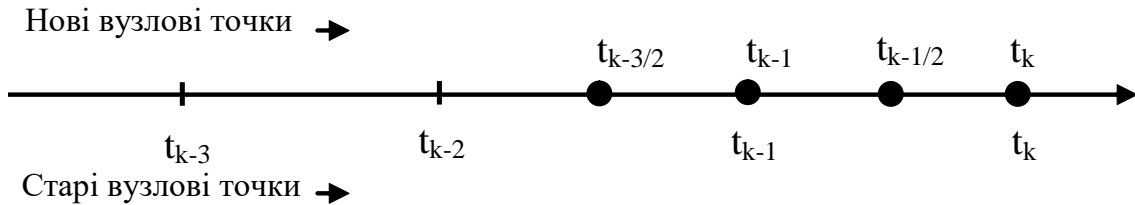


Рис. 9.1. Зменшення довжини кроку до $h/2$.

Набагато простіше збільшити довжину кроку, тоді для розрахунків за одно-кроковим методом потрібно розрахувати сім попередніх точок (рис. 9.2), щоб подвоїти крок.

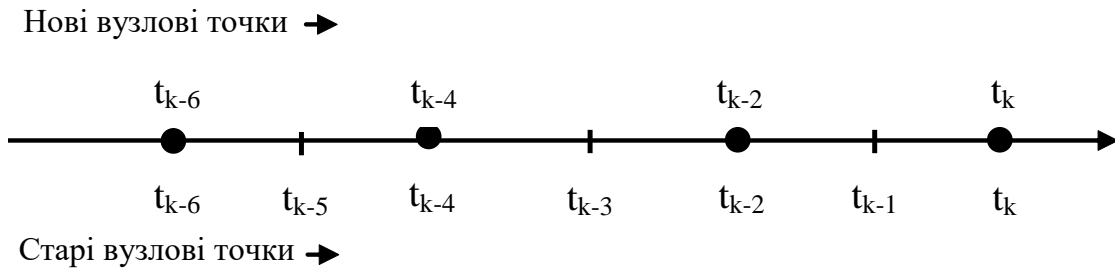


Рис. 9.2. Збільшення довжини кроку до $2 \cdot h$.

9.2. Метод Мілна-Сімпсона

Інший розповсюджений метод прогнозу-корекції це метод Мілна-Сімпсона. Його прогноз базується на інтегруванні функції $f(t, y(t))$ на інтервалі $[t_{k-3}, t_{k+1}]$:

$$y(t_{k+1}) = y(t_{k-3}) + \int_{t_{k-3}}^{t_{k+1}} f(t, y(t)) dt. \quad (9.12)$$

Прогноз використовує наближення поліномом Лагранжа для $f(t, y(t))$, побудоване по точкам (t_{k-3}, f_{k-3}) , (t_{k-2}, f_{k-2}) , (t_{k-1}, f_{k-1}) та (t_k, f_k) . Інтегруємо його по інтервалу $[t_{k-3}, t_{k+1}]$ і отримуємо прогноз Мілна:

$$p_{k+1} = y_{k-3} + \frac{4h}{3}(2f_{k-2} - f_{k-1} + 2f_k). \quad (9.13)$$

Коректор отримуємо аналогічно. На цьому етапі використаємо значення p_{k+1} . Будуємо інший поліном Лагранжа для функції $f(t, y(t))$ по точках (t_{k-1}, f_{k-1}) , (t_k, f_k) і у новій точці $(t_{k+1}, f_{k+1}) = (t_{k+1}, f(t_{k+1}, p_{k+1}))$. Інтегруємо поліномом на відрізку $[t_{k-1}, t_{k+1}]$ і в результаті отримаємо відому формулу Сімпсона:

$$y_{k+1} = y_{k-1} + \frac{h}{3}(f_{k-1} + 4f_k + f_{k+1}). \quad (9.14)$$

Оцінка помилки та корекція. Залишкова складова формули чисельного інтегрування використовується для отримання як прогнозу, так і коректору порядку $O(h^5)$. Остаточна загальна помилка (О.З.П.) для формул (9.13) та (9.14) дорівнює

$$y(t_{k+1}) - p_{k+1} = \frac{28}{90} y^{(5)}(c_{k+1}) h^5 \quad (\text{О.З.П. для прогнозу}) \quad (9.15)$$

$$y(t_{k+1}) - y_{k+1} = \frac{-1}{90} y^{(5)}(d_{k+1}) h^5 \quad (\text{О.З.П. для коректора}) \quad (9.16)$$

Припустимо, що крок h настільки малий, що $y^{(5)}(t)$ майже стала на інтервалі $[t_{k-3}, t_{k+1}]$. Тоді із формул (9.15) і (9.16) можемо виключити елементи, які містять похідну п'ятого порядку, і в результаті отримати формулу:

$$y(t_{k+1}) - p_{k+1} \approx \frac{28}{29} (y_{k+1} - p_{k+1}). \quad (9.17)$$

Формула (9.17) дає оцінку прогнозу, яка ґрунтується на обчисленні значень p_{k+1} і y_{k+1} і не використовує $y^{(5)}(t)$. Її можна застосовувати для покращення значень прогнозу. Якщо припустити, що різниця між значеннями прогнозу p_{k+1} і корекції y_{k+1} на кожному кроці змінюється повільно, то можна підставити p_k і y_k замість p_{k+1} і y_{k+1} в формулу (9.17) і отримати наступний *управляючий параметр*:

$$m_{k+1} = p_{k+1} + 28 \frac{y_k - p_k}{29} \quad (9.18)$$

Його значення використовується замість p_{k+1} на кроці корекції, і формула (9.14) приймає вигляд:

$$y_{k+1} = y_{k-1} + \frac{h}{3}(f_{k-1} + 4f_k + f(t_{k+1}, m_{k+1})) \quad (9.19)$$

Таким чином, в покращеному модифікованому методі Мілна-Сімпсона послідовно використовуються наступні формули:

$$\begin{aligned} p_{k+1} &= y_{k-3} + \frac{4h}{3}(2f_{k-2} - f_{k-1} + 2f_k) && \text{(прогноз)} \\ m_{k+1} &= p_{k+1} + 28\frac{y_k - p_k}{29} && \text{(управляючий параметр)} \\ f_{k+1} &= f(t_{k+1}, m_{k+1}) && (9.20) \\ y_{k+1} &= y_{k+1} + \frac{h}{3}(f_{k-1} + 4f_k + f_{k+1}) && \text{(коректор)} \end{aligned}$$

9.3. Метод Хеммінга

Ще один важливий метод для розв'язання задачі Коші $y' = f(t, y)$ з початковою умовою $y(a) = y_0$ на інтервалі $[a, b]$ – метод Хеммінга. Прогноз, як і в попередніх методах, базується на наближенні поліномом Лагранжа для $f(t, y(t))$. Він побудований по точкам (t_{k-3}, f_{k-3}) , (t_{k-2}, f_{k-2}) , (t_{k-1}, f_{k-1}) та (t_k, f_k) . При інтегруванні отримуємо прогноз Хеммінга:

$$p_{k+1} = y_{k-3} + \frac{4h}{3}(2f_{k-2} - f_{k-1} + 2f_k) \quad (9.21)$$

Коректор отримуємо, будуючи другий поліном Лагранжа для функції $f(t, y(t))$ по точках (t_{k-1}, f_{k-1}) , (t_k, f_k) і у новій точці $(t_{k+1}, f_{k+1}) = (t_{k+1}, f(t_{k+1}, p_{k+1}))$. В результаті одержуємо формулу:

$$y_{k+1} = \frac{-y_{k-2} + 9y_k}{8} + \frac{3h}{8}(-f_{k-1} + 2f_k + f_{k+1}). \quad (9.22)$$

Управляючий параметр:

$$m_{k+1} = p_{k+1} + 9\frac{y_k - p_k}{121} \quad (9.23)$$

9.4. Обмеження при застосуванні методів прогнозу-корекції

Методи прогнозу корекції при великому кроці розрахунку можуть бути не стійкими. При таких розрахунках похибка ітерації збільшується від кроку до кроку, що і обумовлює нестійкість. Для забезпечення стійкості розглянутих методів потрібно корегувати крок на етапі його вибору. З умови стійкості метод крок вибирається за нерівностями [5]:

$$\begin{aligned} h < \frac{0,75}{|f_y(t, y)|} & \quad (\text{в методі Адамса – Бешфорса – Мултона}) \\ h < \frac{0,45}{|f_y(t, y)|} & \quad (\text{в методі Мілна – Сімпсона}) \\ h < \frac{0,69}{|f_y(t, y)|} & \quad (\text{в методі Хемінга}) \end{aligned} \quad (9.24)$$

Методи Адамса-Бешфорса-Мултона і Хемінга дозволяють отримати результати з високим рівнем точності за невелику кількість кроків, тобто швидко збігаються до розв'язку. Метод Сімпсона застосовують переважно, коли розв'язок рівняння є негладкою функцією.

Наведемо *приклад* розв'язання задачі Коші для рівняння $y' = 30 - 5y$ з початковими умовами $y(0) = 1$ на інтервалі $[0; 10]$. Всі три розглянуті вище багатокрокові методи мають порядок похибки розв'язку $O(h^4)$. Побудуємо графіки розв'язків рівняння при мінімальній (меншій за визначений за критерієм стійкості методу) кількості кроків і при кількості кроків, визначеній за формулами (9.24).

Для методу Адамса-Бешфорса-Мултона:

$$h \leq \frac{0,75}{|f_y(t, y)|} = \frac{0,75}{5} = 0,15 \Rightarrow \frac{b-a}{N} = \frac{10}{65} = 0,1538 ;$$

тому $N = 65$ (рис. 9.3).

*// У прикладі $|f_y(t, y)| = (30 - 5y)' = -5$

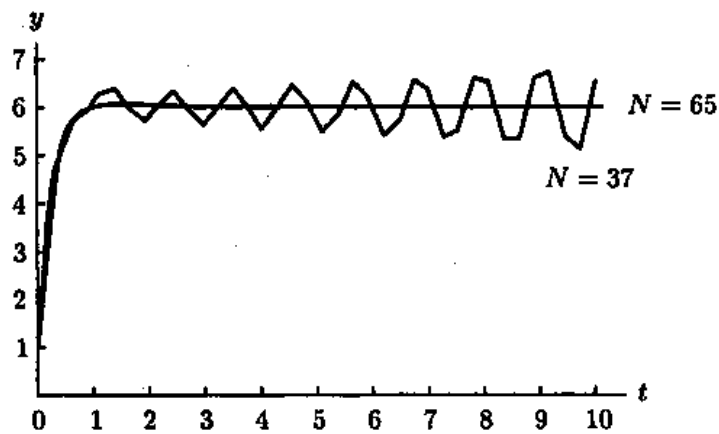


Рис. 9.3. Розв'язок задачі за методом Адамса-Бешфорса-Маултона.

Для методу Мілна-Сімпсона:

$$h \leq \frac{0,45}{|f_y(t, y)|} = \frac{0,45}{5} = 0,09 \Rightarrow \frac{b-a}{N} = \frac{10}{110} = 0,0909 ;$$

тому $N = 110$ (рис. 9.4).

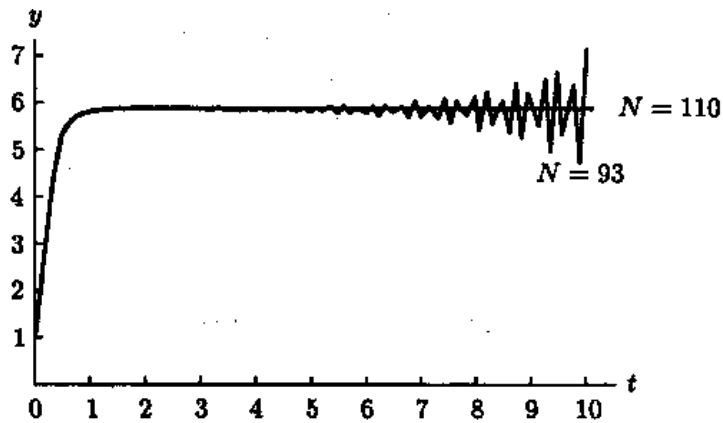


Рис. 9.4. Розв'язок задачі за методом Мілна-Сімпсона.

Для методу Хемінга:

$$h \leq \frac{0,69}{|f_y(t, y)|} = \frac{0,69}{5} = 0,138 \Rightarrow \frac{b-a}{N} = \frac{10}{70} = 0,1428 ;$$

тому $N = 70$ (рис. 9.5).

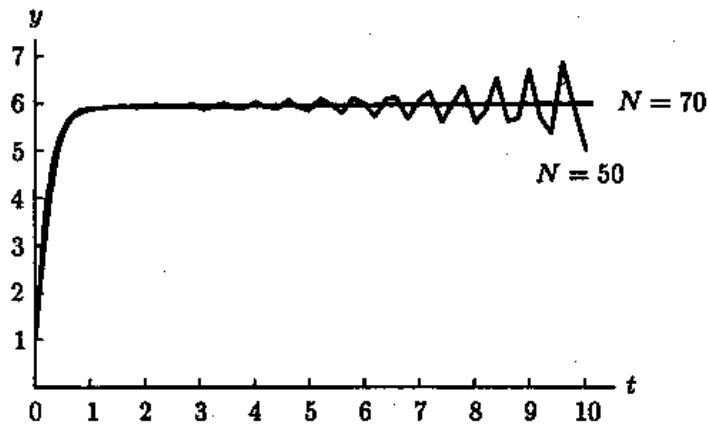


Рис. 9.5. Розв'язок задачі за методом Хемінга.

Таким чином, з прикладу видно, що при порушенні умов обмеження кроку розв'язок втрачає стійкість, що не дозволяє довіряти отриманим результатам.

Контрольні питання

1. В чому полягає суть багатокрокових методів розв'язання диференціальних рівнянь в частинних похідних?
2. Яка роль управляючого параметра у підвищенні точності багатокрокового методу розв'язання диференціальних рівнянь в частинних похідних?
3. Які обмеження мають вхідні параметри багатокрокових методів? Чим викликані дані обмеження?
4. Який з методів, розглянутих у розділі є найбільш ефективним для рівнянь з розв'язком, що представлений гладкою функцією?
5. Яку точність розв'язків забезпечують методи прогнозу-корекції?

Розділ 10. Крайові задачі для звичайних диференціальних рівнянь. Методи сіток

10.1. Основні поняття

Для знаходження єдиного розв'язку диференціального рівняння потрібно задати певні допоміжні умови, що використовуються для обчислення інтегралу (при класичному представленні розв'язку диференціального рівняння). Для рівняння n -го порядку мінімально необхідні n таких умов. Якщо умови задані для одного значення незалежної змінної (тобто для абсциси x_0 чи x_N одного з кінців відрізка розв'язання задачі), то це *початкові умови* для розв'язання задачі Коші. Якщо ж додаткові умови передбачені для значень x_i на різних кінцях відрізка, то маємо *диференціальну крайову задачу* і *граничні умови* для її розв'язання.

Щоб розв'язати звичайне диференціальне рівняння, що моделює конкретний неперервний процес, необхідно знати такі допоміжні умови:

- *початкові умови*, так для рівняння $\frac{dy}{dx} = f(x, y)$, розв'язок якого потрібно знайти на відрізку $[x_0, x_n]$, початкову умову можемо записати у вигляді $y(x)|_{x=x_0} = y_0$;
- *граничні умови*, тобто для рівняння $\frac{dy}{dx} = f(x, y)$, розв'язок якого потрібно знайти на відрізку $[x_0, x_n]$, граничними умовами вважаються $y(x_0)=y_0$ та $y(x_n)=y_n$.

Крайова диференціальна задача пошуку часткового розв'язку диференціального рівняння. Для двоточкової крайової задачі диференціальне рівняння другого порядку запишемо у вигляді:

$$\frac{d^2 y}{dx^2} = f(x, y, y'), \quad x \in [a, b] \quad (10.1)$$

з крайовими умовами

$$y(a) = y_0, \quad y(b) = y_n \quad (10.2)$$

Додаткові умови можуть зв'язувати між собою значення кількох функцій в різних точках.

Крайова задача може мати єдиний розв'язок, або навіть не один розв'язок, а також розв'язок задачі може не існувати. Тому перед початком розв'язання крайової задачі потрібно перевірити умови існування розв'язку задачі. Теорема 10.1 визначає загальні умови існування єдиного розв'язку диференційної крайової задачі.

Теорема 10.1. Допустимо, що $f(x, y, z)$ неперервна в області $D = \{(x, y, z) : a \leq x \leq b, -\infty < y < \infty, -\infty < z < \infty\}$ (10.3)

і що рівняння

$$\frac{\partial f}{\partial y} = f_y(x, y, z) \quad \text{і} \quad \frac{\partial f}{\partial z} = f_z(x, y, z) \quad (10.4)$$

теж неперервні на D . Якщо існує постійна $M > 0$, для якої виконуються умови

$$\begin{aligned} f_y(x, y, z) &> 0, \quad \text{для всіх} \quad (x, y, z) \in D \\ f_z(x, y, z) &\leq M, \quad \text{для всіх} \quad (x, y, z) \in D \end{aligned} \quad (10.5)$$

то крайова задача (4.1), (4.2) має єдиний розв'язок $y(x)$ для $a \leq x \leq b$.

У обчислювальній практиці найчастіше зустрічаються доточкові лінійні крайові задачі виду:

$$L(y) = y'' + p(x) \cdot y' + q(x) \cdot y = f(x), \quad x \in [a, b] \quad (10.6)$$

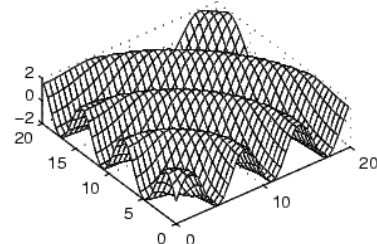
з крайовими умовами

$$\begin{aligned} l_a(y) &= \alpha_0 y(a) + \beta_0 y'(a) = \gamma_0 \\ l_b(y) &= \alpha_1 y(b) + \beta_1 y'(b) = \gamma_1 \end{aligned} \quad (10.7)$$

де $\alpha_0^2 + \beta_0^2 \neq 0$, $\alpha_1^2 + \beta_1^2 \neq 0$. Умови, яким повинні задовольняти функції $p(x)$, $q(x)$ та $f(x)$ для забезпечення єдиного розв'язку задачі визначаються наслідком з теореми 10.1.

Наслідок теореми 10.1. Якщо $p(x)$ і $q(x)$ неперервні на D і $q(x) < 0$, то задача (10.6, 10.7) має єдиний розв'язок на проміжку $[a, b]$.

Частинним випадком доточкової задачі є задача про механічні та електричні коливання, в якій функції $p(x)$, $q(x)$ та $f(x)$ є константами, а незалежною змінною, по якій відбувається диференціювання, є час t .



Граничні умови визначають ступінь крайової задачі.

✓ Перша крайова задача має умови $\alpha_0 = \alpha_1 = 0$. (10.8)

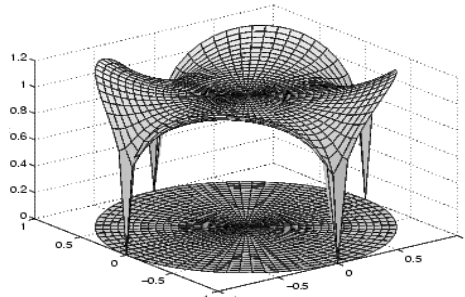
✓ Друга крайова задача має умови $\beta_0 = \beta_1 = 0$. (10.9)

✓ Третя крайова задача записується так:

$$\begin{aligned} l_a(y) &= \alpha_0 y(a) + \beta_0 y'(a) = \gamma_0 \\ l_b(y) &= \alpha_1 y(b) + \beta_1 y'(b) = \gamma_1 \end{aligned} \quad (10.10)$$

за умови, що $\alpha_0^2 + \beta_0^2 \neq 0$, $\alpha_1^2 + \beta_1^2 \neq 0$.

Знаходження розв'язку крайової задачі у точній (аналітичній формі) значно складніша задача, ніж розв'язок задачі Коші. Тому для розв'язування крайових задач розроблено багато наближених методів, що діляться на дві групи:



- *наближено-аналітичні методи*, які приводять до наближеного розв'язку крайової задачі на відрізку $[a, b]$ у вигляді аналітичної функції;
- *чисельні методи*, у яких розв'язок крайової задачі визначається у вигляді табличної функції, заданої на сітці відрізків $a \leq x_0 < x_1 < \dots < x_{n-1} < x_n \leq b$.

Розглянемо деякі з методів розв'язання диференційної крайової задачі.

10.2. Методи сіток.

При проектуванні технічних об'єктів виникає необхідність аналізу неперервних фізичних, економічних та інших процесів, що описуються диференційними рівняннями в частинних похідних. Це задачі дослідження напруженого стану конструкцій, їх деталей та розрахунки на міцність, розрахунок теплових режимів роботи деталей, вивчення аеро- та гідродинамічних властивостей технічних об'єктів, моделювання зміни курсу валюти в залежності від фінансово-економічних показників, вивчення динаміки змін в чисельності населення залежно від фінансово-економічних та соціально-

економічних чинників у державі та багато інших. Для розв'язання таких задач найбільш розповсюдженими є **методи сіток**.

Сутність цих методів полягає в апроксимації початкової безперервної функції множиною наближених значень, що розраховані в деяких точках області, які називаються вузлами $a = x_0 < x_1 < \dots < x_{n-1} < x_n = b$. Сукупність вузлів, з'єднаних певним чином, утворюють сітку, яка є *дискретною моделлю області визначення* шуканої функції.

За допомогою методів сіток *диференційну крайову задачу* зводять до системи лінійних або нелінійних алгебраїчних рівнянь відносно невідомих вузлових значень функції. Найбільше розповсюдження в сучасних математичних середовищах та системах автоматизованого проектування отримали два методи сіток:

- метод кінцевих різниць (МКР),
- метод кінцевих елементів (МКЕ).

10.2.1. Метод кінцевих різниць

Розглянемо *метод кінцевих різниць*. Суть методу полягає у заміні похідних диференційного рівняння задачі різницевиими аналогами, запису різницевих рівнянь відносно всіх точок області визначення задачі й у розв'язанні отриманої системи різницевих рівнянь. Порядок розв'язання диференційної крайової задачі МКР можемо відобразити наступними етапами:

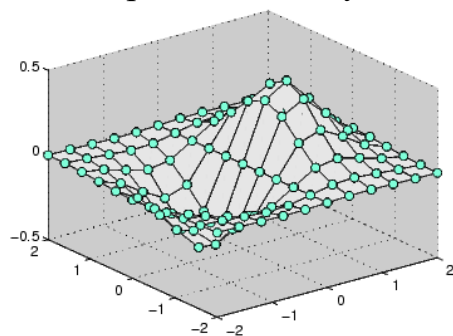
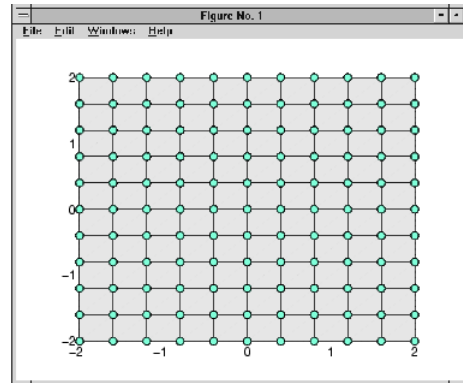
1) розбиття області визначення диференційної задачі вузловими точками,

2) заміна похідних у рівнянні, що описує задачу, відповідно до ступеня похідної та місця розташування вузлових точок різницевиими аналогами,


3) на основі отриманого в п.2 різницевого рівняння побудова системи різницевих рівнянь відносно всіх точок області визначення,

4) розв'язок системи різницевих рівнянь з врахуванням граничних умов,


5) представлення результатів розрахунків у зручному для аналізу вигляді (графіка, чи таблиці) та аналіз отриманих розв'язків.



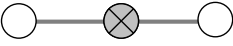
Для заміни похідних використовують ліві, праві і центральні різницеві аналоги. Центральні різницеві аналоги похідних першого – другого порядків представлені рівняннями (10.11) ÷ (10.21):




$$\left(\frac{\partial u}{\partial x} \right)_{ij} = \frac{u_{i+1,j} - u_{i-1,j}}{2h_1} \quad (10.11)$$



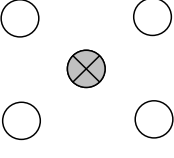
$$\left(\frac{\partial u}{\partial y} \right)_{ij} = \frac{u_{i,j+1} - u_{i,j-1}}{2h_2} \quad (10.12)$$



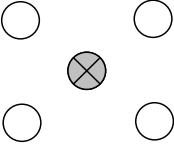
$$\left(\frac{\partial^2 u}{\partial x^2} \right)_{ij} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_1^2} \quad (10.13)$$



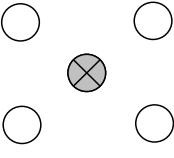
$$\left(\frac{\partial^2 u}{\partial y^2} \right)_{ij} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_2^2} \quad (10.14)$$



$$\left(\frac{\partial^2 u}{\partial x \partial y} \right)_{ij} \approx \frac{u_{i+1,j+1} - u_{i+1,j-1} - u_{i-1,j+1} + u_{i-1,j-1}}{4h_1 h_2} \quad (10.15)$$



$$\left(\frac{\partial u}{\partial x} \right)_{ij} \approx \frac{u_{i+1,j+1} - u_{i-1,j+1} + u_{i+1,j-1} - u_{i-1,j-1}}{4h_1} \quad (10.16)$$



$$\left(\frac{\partial u}{\partial y} \right)_{ij} \approx \frac{u_{i+1,j+1} - u_{i+1,j-1} + u_{i-1,j+1} - u_{i-1,j-1}}{4h_2} \quad (10.17)$$

$$\begin{array}{c} \circ - \circ - \otimes - \circ - \circ \end{array} \left(\frac{\partial^2 u}{\partial x^2} \right)_{ij} \approx \frac{-u_{i+2,j} + 16u_{i+1,j} - 30u_{i,j} + 16u_{i-1,j} - u_{i-2,j}}{12h_1^2} \quad (10.18)$$

$$\begin{array}{c} \circ \\ \circ \\ \circ - \otimes \\ \circ \\ \circ \end{array} \left(\frac{\partial^2 u}{\partial y^2} \right)_{ij} \approx \frac{-u_{i,j+2} + 16u_{i,j+1} - 30u_{i,j} + 16u_{i,j-1} - u_{i,j-2}}{12h_2^2} \quad (10.19)$$

$$\begin{array}{ccc} \circ & \circ & \circ \\ \circ & \otimes & \circ \\ \circ & \circ & \circ \end{array} \left(\frac{\partial^2 u}{\partial x^2} \right)_{ij} \approx \frac{1}{3h_1^2} (u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1} + u_{i+1,j} - \\ - 2u_{i,j} + u_{i-1,j} + u_{i+1,j-1} - 2u_{i,j-1} + u_{i-1,j-1}) \quad (10.20)$$

$$\begin{array}{ccc} \circ & \circ & \circ \\ \circ & \otimes & \circ \\ \circ & \circ & \circ \end{array} \left(\frac{\partial^2 u}{\partial y^2} \right)_{ij} \approx \frac{1}{3h_2^2} (u_{i+1,j+1} - 2u_{i+1,j} + u_{i+1,j-1} + u_{i,j+1} - \\ - 2u_{i,j} + u_{i,j-1} + u_{i-1,j+1} - 2u_{i-1,j} + u_{i-1,j-1}) \quad (10.21)$$

Розглянемо застосування метода кінцевих різниць (МКР) на прикладі стержня, закріпленого у стіні і підігрітого джерелом тепла, що підводиться до вільного кінця стержня. Припустимо, що існує невідома функція температури $\Phi(x, t)$, яка відображає нагрів стержня одиничної довжини $x \in [0, 1]$ у продовж часу $t \in [0, T]$ (рис. 10.3).

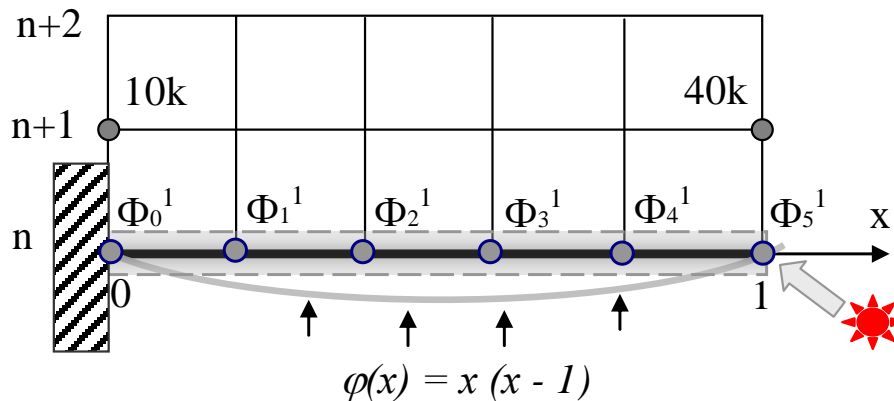


Рис 10.3. Одновимірний елемент, розбитий вузловими точками.

Нагрівання забезпечується зовнішнім джерелом тепла :

$$\varphi(x) = x(x - 1) \quad (10.22)$$

Диференціальне рівняння, що описує процес розповсюдження тепла у стержні запишемо у вигляді:

$$\frac{\partial \Phi}{\partial t} = L \frac{\partial^2 \Phi}{\partial x^2} \quad (10.23)$$

В момент часу t граничні умови задачі:

- при $x = 0$ температура дорівнює $10 \cdot k$;
- при $x = 1$ температура дорівнює $(k + 1) \cdot 40$.

Потрібно знайти розв'язок початково-крайової задачі :

$$\left\{ \begin{array}{l} \frac{\partial \Phi}{\partial t} = L \frac{\partial^2 \Phi}{\partial x^2} \\ \Phi(x, 0) = x^2 - x \\ \Phi(0, t) = k \cdot 10 \\ \Phi(1, t) = (k + 1) \cdot 40 \end{array} \right. \quad (10.24)$$

Різницева апроксимація диференціального рівняння має вигляд:

$$r\Phi_{j-1}^{n+1} - (1 + 2r)\Phi_j^{n+1} + r\Phi_{j+1}^{n+1} = -\Phi_j^n \quad (10.25)$$

Прийmemo: $\alpha = 1$, $h = \frac{1}{5}$, $\Delta t = \frac{1}{100}$, $r = \frac{\alpha \cdot \Delta t}{h^2} = \frac{\Delta t}{h^2}$. Тоді $r = \frac{1}{4}$.

Прийmemo також $n = 0$, $j = 1$.

Замінивши диференціальне рівняння (10.23) різницеvim (10.25) і записавши його відносно вузлових точок (рис.10.3) отримаємо систему лінійних рівнянь:

$$\begin{cases} \frac{1}{4}\Phi_0^1 - \frac{3}{2}\Phi_1^1 + \frac{1}{4}\Phi_2^1 = -\Phi_1^0 \\ \frac{1}{4}\Phi_1^1 - \frac{3}{2}\Phi_2^1 + \frac{1}{4}\Phi_3^1 = -\Phi_2^0 \\ \frac{1}{4}\Phi_2^1 - \frac{3}{2}\Phi_3^1 + \frac{1}{4}\Phi_4^1 = -\Phi_3^0 \\ \frac{1}{4}\Phi_3^1 - \frac{3}{2}\Phi_4^1 + \frac{1}{4}\Phi_5^1 = -\Phi_4^0 \end{cases} \quad (10.26)$$

Оскільки вузлових точок 6, а рівнянь в системі 4, використаємо граничні умови задачі. Нехай $\Phi_0^l = 10$, $\Phi_5^l = 80$, тобто $k = 1$. Кожне $\Phi_i^0 = n_i \cdot m_i$, де n_i – відстань в долях відрізка до початку стержня, m_i – відстань в долях відрізка до кінця стержня. Наприклад: $\Phi_2^0 = \frac{2}{5} \cdot \left(-\frac{3}{5}\right) = -\frac{6}{25}$.

Побудуємо систему рівнянь:

$$\begin{cases} -\frac{3}{2}\Phi_1^1 + \frac{1}{4}\Phi_2^1 + 0\Phi_3^1 + 0\Phi_4^1 = \frac{4}{25} - 10 \cdot \frac{1}{4} \\ \frac{1}{4}\Phi_1^1 - \frac{3}{2}\Phi_2^1 + \frac{1}{4}\Phi_3^1 + 0\Phi_4^1 = \frac{6}{25} \\ 0\Phi_1^1 + \frac{1}{4}\Phi_2^1 - \frac{3}{2}\Phi_3^1 + \frac{1}{4}\Phi_4^1 = \frac{6}{25} \\ 0\Phi_1^1 + 0\Phi_2^1 + \frac{1}{4}\Phi_3^1 - \frac{3}{2}\Phi_4^1 = \frac{4}{25} - 80 \cdot \frac{1}{4} \end{cases} \quad (10.27)$$

Приведемо коефіцієнти системи до цілих чисел:

$$\begin{cases} -150\Phi_1^1 + 12,5\Phi_2^1 + 0\Phi_3^1 + 0\Phi_4^1 = -117 \\ 12,5\Phi_1^1 - 75\Phi_2^1 + 12,5\Phi_3^1 + 0\Phi_4^1 = 12 \\ 0\Phi_1^1 + 12,5\Phi_2^1 - 75\Phi_3^1 + 12,5\Phi_4^1 = 12 \\ 0\Phi_1^1 + 0\Phi_2^1 + 12,5\Phi_3^1 - 75\Phi_4^1 = -\frac{496}{25} \end{cases}$$

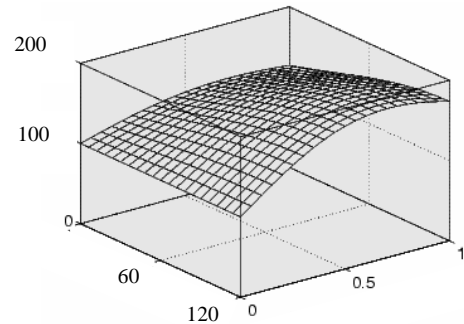
$$\begin{cases} -300\Phi_1^1 + 25\Phi_2^1 + 0\Phi_3^1 + 0\Phi_4^1 = -234 \\ 25\Phi_1^1 - 150\Phi_2^1 + 25\Phi_3^1 + 0\Phi_4^1 = 24 \\ 0\Phi_1^1 + 25\Phi_2^1 - 150\Phi_3^1 + 25\Phi_4^1 = 24 \\ 0\Phi_1^1 + 0\Phi_2^1 + 25\Phi_3^1 - 150\Phi_4^1 = -912 \end{cases}$$

Розв'язати отриману систему цілком можливо одним з точних методів розв'язання систем лінійних алгебраїчних рівнянь. Але

ефективніше застосовувати метод прогонки, оскільки система рівнянь є три діагональною.

Метод прогонки. Розглянемо лінійну крайову задачу на відрізку $x \in [a, b]$:

$$\begin{aligned} y'' + p(x)y' + q(x)y &= f(x) \\ \alpha_0 y(a) + \alpha_1 y'(a) &= A \\ \beta_0 y(b) + \beta_1 y'(b) &= B \end{aligned} \quad (10.28)$$



при $|\alpha_0| + |\alpha_1| \neq 0$, $|\beta_0| + |\beta_1| \neq 0$, та припустимо, що функції $p(x)$, $q(x)$, $f(x)$ неперервні на $[a, b]$.

Від диференційного рівняння (10.28) перейдемо до кінцево-різницевого. Для цього розіб'ємо відрізок $[a, b]$ на n рівних частин із кроком $h = \frac{b-a}{n}$. Покладаючи, що $x_i = x_0 + ih$, $x_0 = a$, $x_n = b$, ($i = 0, 1, \dots, n$) і вводячи позначення: $p(x_i) = p_i$, $q(x_i) = q_i$, $f(x_i) = f_i$, $y(x_i) = y_i$ для внутрішніх точок $x = x_i$ ($i = 1, 2, \dots, n-1$) відрізка $[a, b]$ замість диференційного рівняння (10.28) отримаємо систему кінцево-різничевих рівнянь:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p_i \frac{y_{i+1} - y_{i-1}}{2h} + q_i y_i = f_i$$

де ($i = 1, 2, \dots, n-1$).

Звідси матимемо

$$y_{i+1} + m_i y_i + n_i y_{i-1} = w_i h^2 \quad (i = 1, 2, \dots, n-1) \quad (10.29)$$

$$\text{де } m_i = -\frac{2 - q_i h^2}{1 + \frac{p_i h}{2}}, \quad n_i = \frac{1 - \frac{p_i h}{2}}{1 + \frac{p_i h}{2}}, \quad w_i = \frac{f_i}{1 + \frac{p_i h}{2}}. \quad (10.30)$$

Для похідних на кінцях $x_0 = a$ і $x_n = b$ беремо односторонні похідні:

$$y'_0 = \frac{y_1 - y_0}{h}, \quad y'_n = \frac{y_{n-1} - y_n}{-h}. \quad (10.31)$$

Згідно граничних умов (10.28) матимемо

$$\alpha_0 y_0 + \alpha_1 \frac{y_1 - y_0}{h} = A, \quad \beta_0 y_0 + \beta_1 \frac{y_{n-1} - y_n}{-h} = B. \quad (10.32)$$

Лінійна система диференціальних рівнянь (10.28) з граничними умовами (10.32) містить $(n + 1)$ рівняння першого ступеня відносно невідомих y_0, y_1, \dots, y_n . Розв'яжемо її методом прогонки [17]. Знайдемо y_i з рівняння (10.29)

$$y_i = \frac{w_i}{m_i} h^2 - \frac{1}{m_i} y_{i+1} - \frac{n_i}{m_i} y_{i-1}. \quad (10.33)$$

Припустимо, що за допомогою системи (10.28) з граничними умовами (10.32) із рівняння (10.33) виключена невідома y_{n-1} . Тоді

$$y_i = c_i (d_i - y_{i+1}), \quad (10.34)$$

де c_i, d_i - деякі коефіцієнти. Аналогічно $y_{i-1} = c_{i-1} (d_{i-1} - y_i)$.

Підставимо цей вираз у (10.29) і отримаємо

$$y_{i+1} + m_i y_i + n_i c_{i-1} (d_{i-1} - y_i) = w_i h^2,$$

звідки

$$y_i = \frac{(w_i h^2 - n_i c_{i-1} d_{i-1}) - y_{i+1}}{m_i - n_i c_{i-1}}. \quad (10.35)$$

З цього виразу, враховуючи формулу (10.34) отримаємо рекурентні формули

$$c_i = \frac{1}{m_i - n_i c_{i-1}}, \quad d_i = w_i h^2 - n_i c_{i-1} d_{i-1} \quad (i = 1, \dots, n-1). \quad (10.36)$$

Визначимо c_0 і d_0 . Із першої граничної умови (10.32)

$$\alpha_0 y_0 + \alpha_1 \frac{y_1 - y_0}{h} = A \quad \text{отримаємо}$$

$$y_0 = \frac{Ah - \alpha_1 y_1}{\alpha_0 h - \alpha_1}.$$

З іншого боку, з формули (10.34) при $i=0$ маємо

$$y_0 = c_0 (d_0 - y_1). \quad (10.37)$$

Порівнюючи дві останні нерівності, знаходимо

$$c_0 = \frac{\alpha_1}{\alpha_0 h - \alpha_1}, \quad d_0 = \frac{Ah}{\alpha_1}. \quad (10.38)$$

На основі формул (10.36), (10.38) послідовно визначаються коефіцієнти c_i , d_i , ($i = 1, \dots, n-1$) до c_{n-1} і d_{n-1} включно, це **прямий хід**.

Зворотній хід починається із розрахунку значення y_n . Використовуючи другу крайову умову (10.32) і формулу (10.34) при $i = n-1$, отримаємо систему двох рівнянь

$$\beta_0 y_n + \beta_1 \frac{y_{n-1} - y_n}{-h} = B, \quad y_{n-1} = c_{n-1} (d_{n-1} - y_n). \quad (10.39)$$

Розв'язуючи її відносно y_n , будемо мати

$$y_n = \frac{Bh + \beta_1 c_{n-1} d_{n-1}}{\beta_0 h + \beta_1 (c_{n-1} + 1)}. \quad (10.40)$$

Тепер за формулою (10.34) послідовно знаходимо y_{n-1} , y_{n-2} , y_{n-3} , ..., y_0 .

10.2.2. Метод кінцевих елементів

Розглянемо диференційну крайову задачу для двовимірного диференційного рівняння з частинними похідними. Розв'язок задачі, що визначається

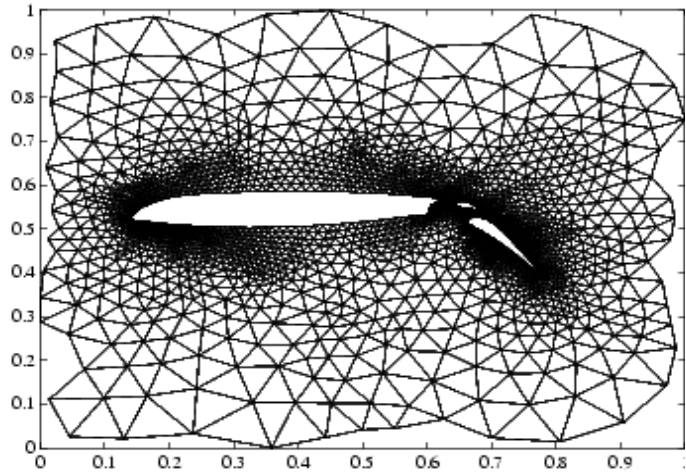


Рис. 10.4. Розбиття області з довільними границями на трикутні кінцеві елементи.

$$L[y] = y'' + p(x)y' + q(x)y = f(x) \quad (10.35)$$

$$x \in [a, b]$$

будемо шукати y вигляді лінійної комбінації простих однотипних функцій:

$$u(x) = \sum_{i=1}^{n-1} c_i \varphi_i(x) \quad (10.36)$$

ЩО МАЮТЬ ВИГЛЯД

$$\varphi_i(x) = \begin{cases} \frac{x - x_{i-1}}{h}, & \text{якщо } x \in [x_{i-1}, x_i], \\ -\frac{x - x_{i+1}}{h}, & \text{якщо } x \in [x_i, x_{i+1}], \\ 0, & \text{якщо } x \notin [x_{i-1}, x_{i+1}]. \end{cases} \quad (10.37)$$

Ці функції називаються фінітними. Графік фінітної функції $\varphi_i(x)$ наведений на рис. 10.5., з якого видно, що функція не дорівнює нулю тільки на інтервалі (x_{i-1}, x_{i+1}) . Фінітні функції, що задаються на відрізку $[a, b]$, лінійно незалежні (і ортогональні в спеціальній енергетичній нормі) і утворюють повну систему в просторі $L_1[a, b]$. Ці властивості є підставою для їх використання в якості базисних функцій в методі Гальоркіна [18].

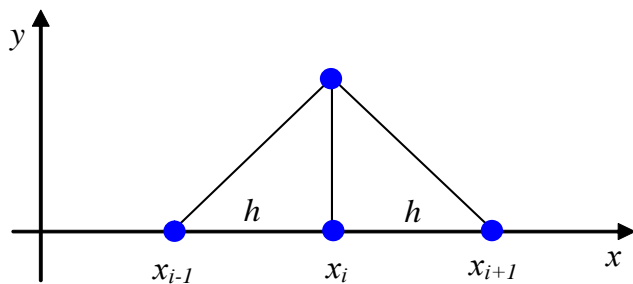


Рис.10.5. Графік фінітної функції

Умову ортогональності запишемо у вигляді:

$$\sum_{i=1}^{n-1} c_i \int_a^b \varphi_k(x) L[\varphi_i(x)] dx = \int_a^b \varphi_k(x) f(x) dx, \quad k = 1, 2, \dots, n-1 \quad (10.38)$$

звідки отримаємо систему лінійних алгебраїчних рівнянь для знаходження невідомих коефіцієнтів c_i . Якщо праву частину рівнянь (10.38) позначити через d_k , то отримаємо наступний вираз для їх обчислення:

$$\begin{aligned} d_k &= \int_a^b \varphi_k(x) f(x) dx = \sum_{k=1}^{n-1} \int_{x_{k-1}}^{x_{k+1}} \varphi_k(x) f(x) dx = \\ &= \int_{x_{k-1}}^{x_k} \frac{x - x_{k-1}}{h} f(x) dx + \int_{x_k}^{x_{k+1}} \frac{x - x_{k+1}}{h} f(x) dx \end{aligned} \quad (10.39)$$

Коефіцієнти системи рівнянь (10.37) позначимо через змінні i з врахуванням формули (10.35) запишемо:

$$a_{ki} = \int_a^b \varphi_k(x) L[\varphi_i(x)] dx = \int_a^b \varphi_k(x) (\varphi_i''(x) + p(x)\varphi_i'(x) + q(x)) dx \quad (10.40)$$

Перший інтеграл у формулі (10.40) обчислимо по частинах:

$$\int_a^b \varphi_k(x) \varphi_i''(x) dx = \varphi_k(x) \varphi_i'(x) \Big|_a^b - \int_a^b \varphi_k'(x) \varphi_i'(x) dx \quad (10.41)$$

При розв'язанні задачі ми використовуємо $(n-1)$ базисних функцій (10.36), які у точках a і b дорівнюють нулю. Тому можемо записати:

$$\int_a^b \varphi_k(x) \varphi_i''(x) dx = - \int_a^b \varphi_k'(x) \varphi_i'(x) dx \quad (10.42)$$

Обчислити коефіцієнти a_{ki} можемо за наступною формулою:

$$a_{ki} = - \int_a^b \varphi_k'(x) \varphi_i'(x) dx + \int_a^b p(x) \varphi_k'(x) \varphi_i(x) dx + \int_a^b q(x) \varphi_k(x) \varphi_i(x) dx = \sum_{j=0}^n \int_{x_j}^{x_{j+1}} (-\varphi_k'(x) \varphi_i'(x) + p(x) \varphi_k'(x) \varphi_i(x) + q(x) \varphi_k(x) \varphi_i(x)) dx \quad (10.43)$$

Для обчислення коефіцієнтів потрібно знайти похідні для фінітних функцій. Диференціюючи (10.37) отримаємо:

$$\varphi_i'(x) = \begin{cases} \frac{1}{h}, & \text{якщо } x \in [x_{i-1}, x_i], \\ -\frac{1}{h}, & \text{якщо } x \in [x_i, x_{i+1}], \\ 0, & \text{якщо } x \notin [x_{i-1}, x_{i+1}]. \end{cases} \quad (10.44)$$

Функція відмінна від нуля тільки на інтервалі $[x_{i-1}, x_{i+1}]$, на якому відмінними від нуля є також фінітні функції і їх похідні (10.37)

та 10.44). Тобто добутки $\varphi'_k(x)\varphi'_i(x)$, $\varphi'_k(x)\varphi_i(x)$, $\varphi_k(x)\varphi_i(x)$ у формулі (10.43) будуть відмінними від нуля тому, що розглядаються

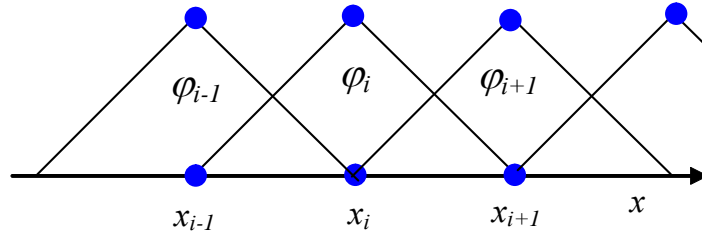


Рис.10.6. Система фінітних функцій.

значення на інтервалі $i-1 \leq k \leq i+1$. Але коли $|i-k| > 1$, то $a_{ki} = 0$. Таким чином, матриця $A = \{a_{ki}\}$ є тридіагональною.

Ненульові елементи матриці обчислюються таким чином, для діагональних елементів, для яких $i=k$, запишемо:

$$a_{ii} = \int_{x_{i-1}}^{x_i} \left[-\frac{1}{h^2} + p(x) \frac{x-x_{i-1}}{h^2} + q(x) \frac{(x-x_{i-1})^2}{h^2} \right] dx + \int_{x_i}^{x_{i+1}} \left[-\frac{1}{h^2} + p(x) \frac{x-x_{i+1}}{h^2} + q(x) \frac{(x-x_{i+1})^2}{h^2} \right] dx \quad (10.45)$$

Для елементів правої бічної діагоналі матриці A , для яких $i=k+1$:

$$a_{ii+1} = \int_{x_i}^{x_{i+1}} \left[\frac{1}{h^2} - p(x) \frac{x-x_{i+1}}{h^2} - q(x) \frac{(x-x_{i+1})(x-x_i)}{h^2} \right] dx \quad (10.46)$$

Для елементів лівої бічної діагоналі матриці A , для яких $i=k-1$:

$$a_{ii-1} = \int_{x_{i-1}}^{x_i} \left[\frac{1}{h^2} - p(x) \frac{x-x_{i-1}}{h^2} - q(x) \frac{(x-x_{i-1})(x-x_i)}{h^2} \right] dx \quad (10.47)$$

Дані елементи формують матрицю A системи алгебраїчних рівнянь, що визначаються формулою (10.38), для невідомих коефіцієнтів c_i .

10.2.3. Алгоритм методу кінцевих елементів.

Метод кінцевих елементів застосовується для розв'язання систем диференційних рівнянь, що описують певні неперервні

процеси у області з довільними границями (рис. 10.4). **Алгоритм методу кінцевих елементів** складається з наступних етапів:

- 1) розбиття заданої області на кінцеві елементи,
- 2) формування апроксимуючої функції кожного елемента, для довільної точки кінцевого елемента апроксимуючий поліном: $\varphi^{(e)} = A^{(e)}R + A_0$, де $A^{(e)}$ – вектор коефіцієнтів поліному, A_0 – вектор вільних елементів, R – вектор координат в точці розгляду. Вектор $A^{(e)}$ і вектор вільних елементів A_0 визначаємо з умов неперервності функції у вузлових точках. Коефіцієнти поліному виражаються через вектор $F^{(e)}$ вузлових значень функції і координати вузлів. Після еквівалентних перетворень отримуємо $\varphi^{(e)} = N^{(e)} F^{(e)}$, де $N^{(e)}$ – матриця-рядок, яка складається з функцій форми кінцевого елемента, ці функції визначаються в кожній точці кінцевого елемента через координати самої точки і координати вузлів елемента,
- 3) об'єднання кінцевих елементів в систему (в ансамбль), на цьому етапі рівняння $\varphi^{(e)} = N^{(e)} F^{(e)}$, що записані для кожного елемента, об'єднуються в систему лінійних чи нелінійних алгебраїчних рівнянь виду $\varphi = NF$, що є моделлю неперервної функції, яка відображає неперервний процес в досліджуваній області,
- 4) визначення вектора вузлових значень функції F , існує декілька алгоритмів обчислення даного вектора, одним з яких є мінімізація функціоналу: $G = \int_V f(\varphi, \varphi'_x, \varphi'_y, \varphi'_z) dV$, який відображає фізичний сенс розв'язуваної задачі.

1) Виділення кінцевих елементів. Від якості розбиття області на кінцеві елементи переважно залежить точність отриманих результатів. Так, наприклад, розбиття досліджуваної області на двовимірні об'єкти в формі рівностороннього трикутника, дозволяє отримати результати з меншою похибкою, ніж при розбитті на витягнуті по формі трикутники.

В МКЕ за рахунок варіювання розмірами кінцевих елементів можливо більш детально вивчати ті під області об'єктів, які найбільш насичені факторами впливу (навантаженнями, температурними, електричними, магнітними полями, неоднорідністю властивостей середовища).

Розбиття області на кінцеві елементи починається від її границь для найбільш точної форми апроксимації границь. На другому етапі розбивають на елементи внутрішні області. Часто розбиття області на елементи проводять в декілька етапів:

1) область розбивають на досить великі підобласті, границі між якими проходять в місцях змінення фізичних властивостей задачі, геометрії області чи області завантаженості певними факторами,

2) кожен під область розбивають на елементи, при формуванні елементів потрібно *уникати різкої зміни розмірів* елементів на границях підобласті (розміри елементів повинні змінюватися поступово, як, наприклад, на рис. 10.4).

Наступний етап – глобальна (в межах досліджуваної області) нумерація вузлів кінцевих елементів. Вона суттєво впливає на ефективність обчислень. В МКЕ матриця коефіцієнтів системи лінійних алгебраїчних рівнянь є сильно розрідженою матрицею стрічкової структури (рис 10.8). Ненульові елементи цієї матриці розташовані близько головної діагоналі. Ціле число L , що дорівнює найбільшій різниці індексів ненульових елементів в рядку, називається *шириною стрічки*. Чим менше значення L , тим менший обсяг пам'яті необхідний для зберігання матриці і тим менші витрати машинного часу на розв'язання системи ЛАР. Ширина стрічки залежить від числа ступенів свободи і від способу нумерації вузлів.

x_{11}	x_{12}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x_{21}	x_{22}	x_{23}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	x_{32}	x_{33}	x_{34}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	x_{43}	x_{44}	x_{45}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	x_{54}	x_{55}	x_{56}	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	x_{65}	x_{66}	x_{67}	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	x_{76}	x_{77}	x_{78}	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	x_{87}	x_{88}	x_{89}	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	x_{98}	x_{99}	x_{910}	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	x_{109}	x_{1010}	x_{1011}	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	x_{1110}	x_{1111}	x_{1112}	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	x_{1211}	x_{1212}	x_{1213}	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	x_{1312}	x_{1313}	x_{1314}	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	x_{1413}	x_{1414}	x_{1415}	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	x_{1514}	x_{1515}	x_{1516}	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	x_{1615}	x_{1616}	x_{1617}	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x_{1716}	x_{1717}	x_{1718}	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x_{1817}	x_{1818}	x_{1819}	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x_{1918}	x_{1919}	x_{1920}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x_{2019}	x_{2020}

Рис. 10.8. Стрічкова матриця на прикладі три діагональної матриці

Число ступенів свободи (M) – це кількість невідомих функцій, що визначаються в кожному вузлі. Наприклад, в двовимірній задачі гідравліки у вузлах визначають тиск і складові швидкості по осям X і Y . У такому випадку $M = 3$. Якщо максимальну різницю між індексами вузлів для окремого елемента позначити через N , то ширину стрічки обчислимо за формулою:

$$L = (N + 1)M \quad (10.47)$$

У деяких випадках зменшення величини N досягається послідовною нумерацією вузлів у напрямку найменшого розміру досліджуваної області (рис. 10.9, в).

Ширина стрічки при одному ступені свободи дорівнює:

$$L_a = (N_a + 1)M_a = (14 + 1) \cdot 1 = 15; \quad L_b = (6 + 1) \cdot 1 = 7; \quad L_v = (4 + 1) \cdot 1 = 5.$$

При двох ступенях свободи:

$$L_{a2} = (N_{a2} + 1)M_{a2} = (14 + 1) \cdot 2 = 30; \quad L_{b2} = (6 + 1) \cdot 2 = 14; \quad L_{v2} = (4 + 1) \cdot 2 = 10.$$

З прикладу бачимо, що раціональна нумерація вузлів в другому випадку скорочує вдвічі необхідний обсяг оперативної пам'яті в порівнянні з першим випадком, а у третьому випадку - втричі.

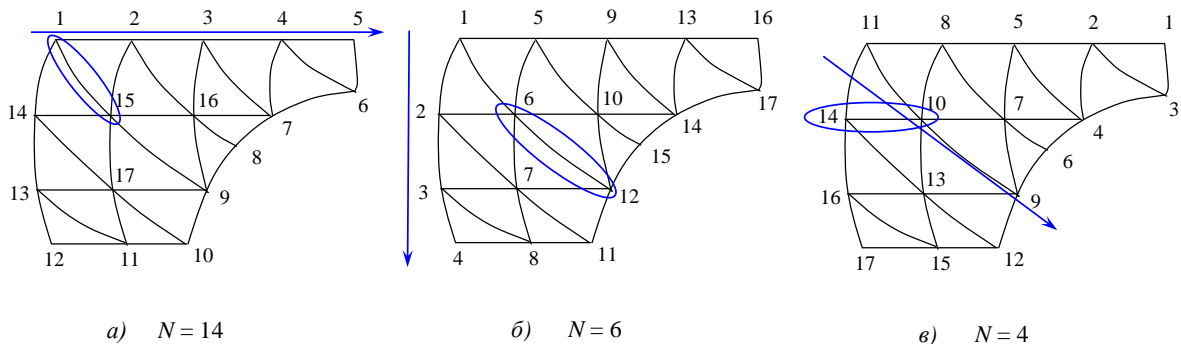


Рис. 10.9. Варіанти нумерації вузлів кінцевих елементів у криволінійній області.

2) Формування апроксимуючої функції. Для визначення апроксимуючої функції найчастіше використовуються поліноми. В залежності від їх ступеню кінцеві елементи поділяються на

- 1) симплекс-елементи – апроксимуються лінійними поліномами,
- 2) комплекс-елементи – апроксимуються поліномами, що мають лінійні й нелінійні складові,

3) мультиплекс елементи – вони мають вигляд комплекс-елементів, але на них накладаються додаткові умови: їх границі повинні бути паралельні координатним осям.

Одновимірний симплекс-елемент. Функція φ апроксимується поліномом:

$$\varphi = a_1 + a_2 x \quad (10.48)$$

Коефіцієнти a_1 і a_2 знаходимо через вузлові значення функції F_1 і F_2 у відповідності до умов неперервності.

$$\begin{aligned} \varphi = F_i \quad \text{при} \quad x = X_i \\ \varphi = F_j \quad \text{при} \quad x = X_j \end{aligned} \Rightarrow \begin{cases} F_i = a_1 + a_2 X_i \\ F_j = a_1 + a_2 X_j \end{cases} \Rightarrow \begin{cases} a_1 = F_i - a_2 X_i \\ F_j = F_i - a_2 X_i + a_2 X_j \end{cases}$$

$$\Rightarrow \begin{cases} a_1 = F_i - a_2 X_i \\ F_j - F_i = a_2 (X_j - X_i) \end{cases} \Rightarrow \begin{cases} a_1 = F_i - a_2 X_i \\ a_2 = \frac{F_j - F_i}{L}; \quad L = (X_j - X_i) \end{cases}$$

$$a_1 = F_i - \frac{F_j - F_i}{X_j - X_i} X_i = \frac{F_i X_j - F_i X_i - F_j X_i + F_i X_i}{X_j - X_i} = \frac{F_i X_j - F_j X_i}{X_j - X_i} = \frac{F_i X_j - F_j X_i}{L}$$

Скориставшись отриманими значеннями коефіцієнтів апроксимуючий поліном запишемо у вигляді:

$$\varphi = \frac{F_i X_j - F_j X_i}{L} + \frac{F_j - F_i}{L} X \Rightarrow \varphi = \frac{X_j - x}{L} F_i + \frac{x - X_i}{L} F_j \quad (10.49)$$

де $\frac{X_j - x}{L} = N_i$ та $\frac{x - X_i}{L} = N_j$ – функції форми одновимірного симплекс-елементу. **Функція форми з номером i дорівнює 1 у вузлі з відповідним номером, але дорівнює нулю у всіх інших вузлах.**

Таким чином у підсумку запишемо:

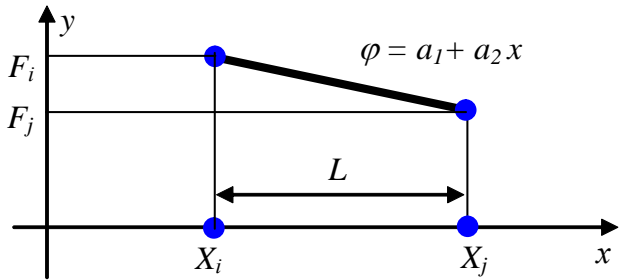


Рис.10.9. Одновимірний симплекс-елемент

$$\varphi = N_i F_i + N_j F_j = \begin{bmatrix} N_i & N_j \end{bmatrix} \cdot \begin{bmatrix} F_i \\ F_j \end{bmatrix} = NF \quad (10.50)$$

Двовимірний симплекс-елемент. Він може бути представлений у вигляді плоского трикутника з прямолінійними сторонами. Інтерполяційний поліном, що апроксимує неперервну функцію φ в межах трикутного симплекс-елемента має вигляд:

$$\varphi = a_1 + a_2 x + a_3 y \quad (10.51)$$

Пронумеруємо вузли елемента i, j, k , рухаючись проти часової стрілки. З умови неперервності функції у вузлах отримаємо:

$$\begin{cases} F_i = a_1 + a_2 X_i + a_3 Y_i \\ F_j = a_1 + a_2 X_j + a_3 Y_j \\ F_k = a_1 + a_2 X_k + a_3 Y_k \end{cases} \quad (10.52)$$

При розв'язанні цієї системи лінійних алгебраїчних рівнянь отримаємо:

$$\begin{aligned} a_1 &= \frac{1}{2S} ((X_j Y_k - X_k Y_j) F_i + (X_k Y_i - X_i Y_k) F_j + (X_i Y_j - X_j Y_i) F_k) \\ a_2 &= \frac{1}{2S} ((Y_j - Y_k) F_i + (Y_k - Y_i) F_j + (Y_i - Y_j) F_k) \\ a_3 &= \frac{1}{2S} ((X_k - X_j) F_i + (X_i - X_k) F_j + (X_j - X_i) F_k) \end{aligned} \quad (10.53)$$

де S – площа кінцевого елемента.

$$S = \frac{1}{2} (X_i (Y_j - Y_k) + X_j (Y_k - Y_i) + X_k (Y_i - Y_j)) \quad (10.54)$$

У підсумку отримаємо:

$$\varphi = N_i F_i + N_j F_j + N_k F_k = \begin{bmatrix} N_i & N_j & N_k \end{bmatrix} \cdot \begin{bmatrix} F_i \\ F_j \\ F_k \end{bmatrix} = NF \quad (10.55)$$

де функції форми двовимірного симплекс-елемента:

$$\begin{aligned} N_i &= \frac{1}{2S} ((X_j Y_k - X_k Y_j) + (Y_i - Y_k)x + (X_k - X_j)y) \\ N_j &= \frac{1}{2S} ((X_k Y_i - X_i Y_k) + (Y_k - Y_j)x + (X_i - X_k)y) \\ N_k &= \frac{1}{2S} ((X_i Y_j - X_j Y_i) + (Y_i - Y_j)x + (X_j - X_i)y) \end{aligned} \quad (10.56)$$

Аналогічно обчислюються функції інших типів кінцевих елементів. Складаються бібліотеки типових функцій кінцевих елементів.

3) Об'єднання кінцевих елементів в систему (ансамбль). Суть цього етапу полягає у заміні довільно вибраних номерів вузлів i, j, k на номери, які отримали вузлові точки при глобальному розбитті досліджуваної області на кінцеві елементи. У підсумку потрібно отримати систему лінійних алгебраїчних рівнянь, при розв'язанні якої обчислюємо значення шуканої функції у будь-якій точці досліджуваної області.

Складемо систему рівнянь для одновимірного об'єкту, що розбитий на 5 елементів. Функція для кожного елемента має вид:

$$\varphi = N_i F_i + N_j F_j \quad (10.57)$$

Запишемо співвідношення між довільними індексами i, j в цьому рівнянні і номерами вузлів в дискретній моделі об'єкту:

елемент 1	$i = 1$	$j = 2$
елемент 2	$i = 2$	$j = 3$
елемент 3	$i = 3$	$j = 4$
елемент 4	$i = 4$	$j = 5$
елемент 5	$i = 5$	$j = 6$

При підстановці значень номерів вузлів дискретної моделі у функцію (10.57) кожного елемента отримуємо систему рівнянь:

$$\begin{cases} \varphi^{(1)} = N_1^{(1)} F_1 + N_2^{(1)} F_2 \\ \varphi^{(2)} = N_2^{(2)} F_2 + N_3^{(2)} F_3 \\ \varphi^{(3)} = N_3^{(3)} F_3 + N_4^{(3)} F_4 \\ \varphi^{(4)} = N_4^{(4)} F_4 + N_5^{(4)} F_5 \\ \varphi^{(5)} = N_5^{(5)} F_5 + N_6^{(5)} F_6 \end{cases} \quad (10.58)$$

Слід звернути увагу на особливість - у системі рівнянь $N_3^{(2)} \neq N_3^{(3)}$, оскільки $N_3^{(2)} = \frac{x - X_2}{L^{(2)}}$; $N_3^{(3)} = \frac{X_3 - x}{L^{(3)}}$.

При відомих значеннях $F_1 \dots F_6$ можемо визначити функцію в будь-якій не вузловій точці області.

У матричній формі формула (10.58) виглядатиме так $\varphi = NF$,
де

$$N = \begin{bmatrix} N_1^{(1)} & N_2^{(1)} & 0 & 0 & 0 & 0 \\ 0 & N_2^{(2)} & N_3^{(2)} & 0 & 0 & 0 \\ 0 & 0 & N_3^{(3)} & N_4^{(3)} & 0 & 0 \\ 0 & 0 & 0 & N_4^{(4)} & N_5^{(4)} & 0 \\ 0 & 0 & 0 & 0 & N_5^{(5)} & N_6^{(6)} \end{bmatrix} \quad (10.59)$$

4) Визначення вектора вузлових значень. Дана задача може вирішуватися кількома методами. Один з методів – мінімізація функціоналу, що пов'язаний з фізичним сенсом задачі [5, 7].

Контрольні питання

1. Яка задача називається диференційною крайовою задачею?
2. Які умови забезпечують існування єдиного розв'язку диференційної крайової задачі?
3. Які умови визначають ступінь крайової задачі?
4. В чому полягає суть методів сіток?
5. В яких випадках застосовується метод кінцевих різниць?
6. На чому оснований алгоритм методу кінцевих різниць?
7. Алгоритм методу кінцевих елементів.
8. В чому особливості застосування методу кінцевих елементів?
9. Від яких факторів залежить оптимізація розрахунків за методом кінцевих елементів?

Додаток А. Короткі теоретичні відомості по програмуванню в інтегрованій системі автоматизації математичних розрахунків MatLab

Інтегрована система автоматизації математичних та науково-технічних розрахунків MatLab є добре апробованою надійною системою розв'язання математичних задач з поданням даних у вигляді матриць. Вона розроблена фірмою MathWorks Inc.

При створенні багатьох автоматизованих систем математичних обчислень розробники прагнули уникнути необхідності написання користувачем власних програмних кодів. Але такий підхід значно обмежував можливості користувача при розв'язанні задач, що використовують нові методи та алгоритми.

Розробники MatLab провели величезну роботу по створенню операторів і функцій для розв'язання багатьох практичних задач. До таких функцій належать як прості (транспонування матриць, обчислення косинуса та ін.) так і складні функції, що реалізують певні методи розв'язання задач. Число таких функцій сягає багатьох тисяч і невпинно збільшується.

А для вирішення задач з використанням алгоритмів користувача, MatLab має потужну *математично-орієнтовану мову програмування високого рівня*. Тому інтегрована система автоматизації математичних MatLab з успіхом застосовується в багатьох країнах Європи для розв'язання нових, нестандартних та найбільш складних математичних задач.

Мова програмування в системі MatLab нагадує Паскаль. Запис коду програми відбувається у редакторі системи, що викликається з рядка головного меню MatLab командами **File** → **New** → **M-file**, але попередньо може бути написаний у будь-якому текстовому редакторі. Файл коду програми називається *m*-файлом і має розширення *.m*, тобто можемо записати: *filename.m* Система MatLab має власний редактор і відладчик. Формули створюваних методів записуються у формі, притаманній мовам програмування, але ця незручність компенсується високою швидкістю обчислень, яка, наприклад, на порядок вища ніж у системі MathCad.

Мова програмування системи MatLab використовує майже всі відомі засоби програмування, в тому числі об'єктно-орієнтоване і візуальне програмування, тобто на ній можна створювати повноцінні

додатки з розвиненою функціональністю та сучасним інтерфейсом. У програмному кодї можна використовувати також внутрішні функції MatLab. Це забезпечує великі можливості при створенні програм як початківцям так і досвідченим програмістам.

В системі MatLab реалізовані два підходи до розв'язання задач:

- ✓ з використанням внутрішніх функцій, що реалізовані розробниками MatLab,
- ✓ за допомогою написаних користувачем власних програм, код яких формується у вигляді *m*-файлів.

Щоб скористатися першим способом розв'язання задач, достатньо задати значення змінної, набрати потрібне ім'я внутрішньої функції з обов'язковими параметрами у рядку запрошення вікна «Command Window» і натиснути «Enter» (рис. А1). Якщо у ході розв'язання задачі використовуються кілька методів, то слід

викликати їх послідовно, вказуючи необхідні параметри та розділяючи їх крапкою з комою (;). Кожен сеанс роботи в MatLab (у середовищі прийнято іменувати їх сесіями) відображається у вікні «Command Window», він вміщує змінні, значення яких задавалися під час сесії, функції, що викликалися, і всі наслідки «проб і помилок» користувача. Як правило при збереженні сеансу користувача цікавлять лише значення змінних та результати обчислень, їх запам'ятовують у вигляді *mat*-файлів з використанням функції *save filename*. Якщо потрібно записати значення лише одної або кількох змінних, то їх імена вказують наприкінці команди запам'ятовування, наприклад: *save filename X ; save filename X Y Z*. Якщо ж потрібно зберегти весь сеанс з відображенням всіх функцій, повідомлень про помилки та ін., то користуються функцією *diary filename*. Всі записані таким чином дані відкриваються у вікні «Command History» (рис. А1).

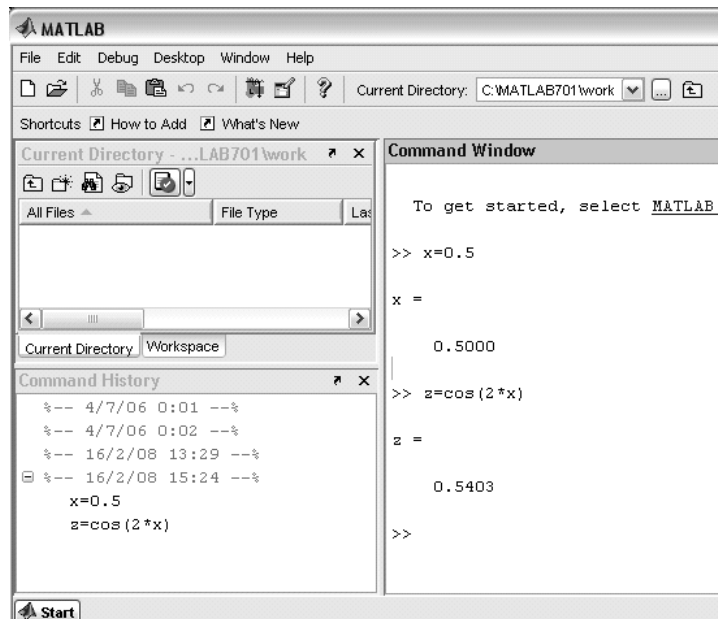


Рис. А1. Вікно системи MatLab при виконанні елементарних операцій.

Для реалізації власних алгоритмів і методів розв'язання задач скористайтеся редактором і відладчиком, які дозволяють отримати власну функцію для розв'язання задачі у вигляді *m*-файлу. При написанні власного коду і роботі над помилками користувач має викликати її у вікні «Command Window» із вказуванням необхідних параметрів, *при цьому у рядку «Current Directory» (рис.1) потрібно вказати повний шлях до папки, в якій знаходиться m-файл.* Якщо створений *m*-файл має помилки, то їх зміст і номер рядка з помилкою виводяться у поточному рядку командного вікна.

Серед *m*-файлів розрізняють **файли коду** та **файли-сценарії**. У файлах коду міститься програмний код, який може складатися з однієї чи кількох функцій. У файлах-сценаріях послідовно записують значення змінних, імена внутрішніх функцій та *m*-файлів, які використовуються для розрахунків.

Файли коду. У *m*-файлі, що містить код, ім'я вихідної функції повинно співпадати з іменем *m*-файлу (рис. 2). Внутрішня мова програмування MatLab дуже схожа з мовою програмування Pascal, в ній використовуються ті ж мовні конструкції, відмінності мають лише при зверненні до елементів матриць. Наприклад, якщо потрібно послідовно звертатися до елементів рядка чи стовпчика матриці в MatLab, то на відповідній позиції замість змінної пишуть двокрапку (:), тобто $M(i, :)$. При його створенні та перевірці редактор забезпечує синтаксичний контроль. Редактор має деякі відладочні можливості –

```

1 function Koren(x1,x2)
2 - l=x2-x1;
3 - step=1/10;
4 - if l>0.00000000000001
5 -     for i=1:1:10
6 -         if xor((exp(x1)-2*(x1)-5>0),(exp(x1+step)-2*(x1+step)-5>0))
7 -             Koren(x1,x1+step)
8 -         end
9 -         x1=x1+step;
10 -     end
11 - else
12 -     x=(x1+x2)/2
13 - end

```

Рис. А2. Приклад файлу-функції у редакторі MatLab.

він дозволяє встановити в тексті файлу контрольні точки, в яких призупиняється виконання програми і користувач може оцінити проміжні результати обчислень. Також є можливість покрокового виконання файлу, що особливо важливо при пошуку помилок. Редактор дозволяє зберегти файл в текстовому форматі з розширенням *.m*

Структура **m-файлу-функції** може бути представлена так:

```
function var=f_name(список_параметрів)
%Основний коментар
%Додатковий коментар
Тіло файлу з будь-якими виразами
var=вираз
```

Файли, записані з розширенням *.m*, можуть бути **файлами-сценаріями**. Такі файли можуть створюватися для задавання функції, введення даних у вигляді векторів чи матриць, побудови графіків із заданими параметрами і т.і. Всі записані в файлі-сценарії дані можливо вводити і в «Command Window», але форма файлу-сценарію більш компактна і його можна зберегти у вигляді файлу. При виклику такого файлу у «Command Window» за його ім'ям всі записані в файлі дані виводяться на екран, після чого можемо викликати файл коду, який їх використовує, реалізуючи певний метод обчислень.

Файл-сценарій (Script-файл) має наступну структуру:

```
%Основний коментар
%Додатковий коментар
Тіло файлу з будь-якими виразами.
```

Наприклад: визначимо функцію: $f(x) = 1 + \frac{x^3}{12}$. Файл сценарій

запишемо у вигляді:

```
function y = stepfunc(x)
y = 1 + x^3/12;
```

При виклику функції у «Command Window» з конкретним значенням змінної x відобразяться наступні записи (буде виведено обчислений результат):

```
» stepfunc(3)
ans =
    3.2500
```

Іноді корисним є розрахунок значень функції за допомогою команди `feval`. У її параметрах функція повинна бути представлена, як рядок:

```
» feval (inline('stepfunc'), 3)
ans =
```

3.2500

Матриці в середовищі MatLab. Всі дані, що вводяться в середовищі MatLab інтерпретуються, як матриці. При введенні матриць розділовим знаком між елементами рядка є пробіл, а між самими рядками – крапка з комою. Наприклад:

```
» M = [1 2 3; 4 5 6; 7 8 9]
```

```
M =
```

```
1 2 3
4 5 6
7 8 9
```

Для вказування певного елемента матриці, користуються індексами:

```
» M (2;2)
```

```
ans =
```

```
5
```

Можемо також задавати матриці у вигляді арифметичних виразів:

```
» C = [1+3^2 cos(0.5) sqrt(9)]
```

```
C =
```

```
10 0.8776 3
```

Задавати матрицю можна як змінну діапазону:

```
» U = 0:0.5:3
```

```
U =
```

```
0 0.5000 1.0000 1.5000 2.0000
2.5000 3.0000
```

При використанні її в якості аргументу отримаємо матрицю того ж розміру:

```
» sin(U)
```

```
ans =
```

```
0 0.4794 0.8415 0.9975 0.9093
0.5885 0.1411
```

В середовищі MatLab є функції для створення матриць з нульовими елементами (їх часто використовують, як матрицю для

запису результатів обчислень) і матриць з одиничними елементами, аргументами цих функцій є їх розмір, що задається у вигляді цілих чисел.

» `W = zeros(4,4);` створення нульової матриці розміром 4×4 .

» `H = ones(4,4);` створення матриці з одиничними елементами розміром 4×4 .

Операції над матрицями. З матрицями в середовищі MatLab можемо проводити наступні операції:

- + додавання,
- віднімання,
- * добуток (множення),
- ^ піднесення до степеня,
- ' транспонування і спряження.

Додаток Б. Пакети прикладних програм інтегрованої системи автоматизації математичних розрахунків MatLab

Система MATLAB має на сьогоднішній день 15 реалізацій, 15-а реалізація відома як версія 2008. Розробник системи фірма – MathWorks. Inc., – піклуючись про рейтинг системи, доповнила її сотнями пакетів прикладних програм, що призначені для розв'язання складних сучасних технічних задач, найбільш розповсюджені з них наведені в таблиці Б1.

Система MATLAB з точки зору її розробників пропонується, як мова програмування високого рівня для математичних та науково-технічних обчислень. Вона відобразила в собі не тільки весь досвід розвитку та комп'ютерної реалізації чисельних методів за останні три десятиліття, а і досвід розвитку математики за всю історію Людства. Система має компілятор, який дозволяє транслювати коди програм, написаних на внутрішній мові MATLAB, в коди мов високого рівня (C, C++, Fortran та ін.) Системою користуються більш ніж 500 000 тисяч легально зареєстрованих користувачів.

Перелік основних програмних продуктів класу MATLAB.

Таблиця Б1

№	Назва пакету програм	Призначення.
1	2	3
1	MATLAB for Windows	Ядро системи, основні бібліотеки для розв'язання науково-технічних, інженерних та математичних задач
	а) MATLAB Compiler	Компілятор для програм на мові програмування системи MATLAB. Транслює коди цих програм в програми на мові C. Застосування компілятора забезпечує можливість створення виконуваних кодів (програм), час виконання яких для програм з великим числом циклічних операцій зменшується у 10-15 раз.
	б) MATLAB C Math Library	Бібліотека додаткових математичних функцій на мові C, що розширює можливості базової системи по розв'язанню математичних задач.
	в) MATLAB C++ Math Library	Бібліотека додаткових математичних функцій на мові C++, що є більш сучасним варіантом мови C. Бібліотека розширює можливості базової системи по розв'язанню математичних задач.

1	2	3
2	Simulink for Windows	Система імітаційного моделювання моделей, що складаються з графічних блоків із заданими властивостями (параметрами). Компоненти моделей можуть бути одиничними графічними блоками та субмоделями, з особистою структурою та зв'язками. До складу моделей можна включати з бібліотеки Simulink джерела сигналів різного виду, віртуальні реєструючі пристрої, графічні засоби анімації.
3	Simulink Real Time Workshop (RTW)	Підсистема імітаційного моделювання в реальному масштабі часу (при наявності додаткових апаратних засобів у вигляді плат розширення ПК), що підключається до Simulink.
4	Excel Link	Програмний засіб для створення інтерфейсу зв'язку з табличним процесором класу Excel, створених фірмою Microsoft Inc. Пакет забезпечує двосторонній зв'язок та обмін даними, перегляд, редагування та обробка даних з середовища MATLAB в Excel, підготовка Excel-додатків та ін.
5	NAG Foundation Toolbox	Одна з найбільших бібліотек математичних функцій, створена групою The Numerical Algorithms Group Ltd. Пакет вміщує понад 240 функцій, організованих у вигляді m-файлів.
6	Neural Networks Toolbox	Пакет прикладних програм, що має засоби для побудови нейронних мереж, які ґрунтуються на поведінці нейрона. Пакет забезпечує ефективну підтримку проектування, навчання та моделювання багатьох відомих мережевих парадигм, від базових моделей перцептрона до найсучасніших асоціативних та самоорганізованих мереж. Пакет застосовується для обробки сигналів, нелінійного управління, фінансового моделювання та ін.
7	Spline Toolbox	Пакет прикладних програм для роботи зі сплайнами. Підтримує одновимірну та багатовимірну сплайн-інтерполяцію, апроксимацію. Забезпечує подання і відображення складних масивів даних і підтримку графіки.
8	Statistics Toolbox	Пакет прикладних програм по статистиці. Вміщує великий набір засобів генерації випадкових чисел, векторів, матриць та масивів з різними законами розподілу, а також великий набір статистичних функцій. Має кілька інтерактивних інструментів для динамічної візуалізації та аналізу даних.
9	Optimization Toolbox	Пакет прикладних програм для розв'язання задач оптимізації і систем нелінійних рівнянь.
10	Fuzzy Logic Toolbox	Пакет прикладних програм для розв'язання задач, що стосуються теорії розмитих множин.

1	2	3
11	Partial Differential Equations Toolbox	Пакет прикладних програм для розв'язання систем диференційних рівнянь у частинних похідних. Надає ефективні засоби для розв'язання систем диференційних рівнянь, у тому числі і жорстких. В пакеті використовується метод кінцевих елементів.
12	Symbolic Math Toolbox	Пакет прикладних програм для розв'язання задач у символному (аналітичному) вигляді, з реалізацією точної арифметики заданої розрядності.
13	Control System Toolbox	Пакет для моделювання аналізу і проектування систем автоматичного керування: безперервних та дискретних.
14	Nonlinear Control Design Toolbox	Пакет призначений для побудови нелінійних систем контролю та управління. Він реалізує метод динамічної оптимізації для систем керування.
15	Robust Control Toolbox	Пакет має засоби для проектування і аналізу багатопараметричних стійких систем керування. Це системи з модельними похибками, динаміка яких відома не повністю чи параметри яких можуть змінюватися в процесі моделювання.
16	LMI Control Toolbox	Пакет забезпечує інтегроване середовище для постановки і розв'язання задач лінійного програмування. Призначений попередньо для проектування систем керування, пакет дозволяє розв'язувати будь-які задачі лінійного програмування практично в будь-якій сфері діяльності.
17	System Identification Toolbox	Пакет вміщує засоби для створення математичних моделей динамічних систем на основі спостереження вхідних-вихідних даних.
18	Signal Processing Toolbox	Пакет призначений для аналізу, моделювання та проектування пристроїв обробки різноманітних сигналів, забезпечення їх фільтрації та інших перетворень.
19	Image Processing Toolbox	Пакет подає широкий спектр засобів для цифрової обробки і аналізу зображень.
20	Wavelet Toolbox	Пакет подає повний набір програм для дослідження багатовимірних нестационарних явищ. Методи пакету розширюють можливості користувача в тих областях, де зазвичай застосовується техніка Фур'є-розкладань. Пакет може застосовуватися для обробки звуків мови і аудіосигналів, телекомунікації, при вирішенні задач геофізики, фінансів, медицини.

Система MATLAB може використовуватися в режимі безпосередніх обчислень – командному режимі, а також – в режимі програмування. Вона є багатоплановою, і тому може виникнути питання – для чого використовувати програмування, якщо величезну

кількість задач є можливість розв'язати за допомогою внутрішніх та додаткових ресурсів системи. Та навіть у найбільшій і найпотужнішій системі нереально передбачити всіх засобів для розв'язання специфічних задач, що стоять перед користувачем. Мова програмування системи MATLAB дозволяє їх ефективно розвиватися, створювати і використовувати в системі найновіші методи розв'язання сучасних задач з врахуванням всіх вимог користувача.

Разом з тим, програми, створені в MATLAB можна транслювати в коди мов високого рівня, що дозволяє використовувати реалізовані у вигляді *m*-файлів чисельні методи для створення функцій, підпрограм та бібліотечних файлів на мовах C, C++, Fortran та інших, число яких розширюється.

Список літературних джерел

1. Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. – М.: Лаборатория базовых знаний, 2002. – 632 с.
2. Тихонов А.Н., Самарский А.А. Методы решения некорректных задач. – М.: Наука, 1986. – 228 с.
3. Демидович Б.П., Марон И.А. Основы вычислительной математики. – М.: Наука, 1970. – 664 с.
4. Метьюз Дж., Финк К. Численные методы. Использование MatLab. – М.: Издательский дом «Вильямс», 2001. – 583 с.
5. Фельдман Л.П., Петренко А.І., Дмитрієва О.А. Чисельні методи в інформатиці. – К.: Видавнича група ВНУ, 2006. – 480 с.
6. Зелінський К.Х., Ігнатенко В.М., Коц О.П. Комп'ютерні методи прикладної математики. – К.: Академперіодика, 2002. – 480 с.
7. Численные методы: Учеб. Пособие для студентов физ.-мат. Спец. Пед. Институтов/ В.М. Зварыкин, В.Г. Житомирский, М.П. Лапчик. М.: Просвещение, 1990, -176 с.
8. Ляшенко М.Я., Головань М.С. Чисельні методи: Підручник. – К.: Либідь, 1996. – 288 с.
9. Бахвалов Н.С., Лапин А.В., Чижонков Е.В. Численные методы в задачах и упражнениях. – М.: Высшая школа, 2000. – 192 с.
10. Поршнёв С.В. Вычислительная математика. Курс лекций. – СПб.: БХВ-Петербург, 2004. – 320 с.
11. Дьяконов В.П. MATLAB 2006/2007/2008, Simulink 5/6/7 Основы применения – СПб: Солон-Пресс, 2008. – 800 с.
12. Гаврилюк М.А., Галамай Т.Г. Прикладные программы и лабораторный практикум для персонального компьютера. – К.: УМКВО, 1988. – 202 с.
13. Норенков И.П. Основы автоматизированного проектирования. – М.: Изд-во МГТУ им. Баумана, 2002. – 336 с.
14. Мартынов Н.Н., Иванов А.П. Matlab 5.x Вычисления, визуализация, программирование. М.: Кудиц-Образ, 2000. – 336 с.
15. Ортега Д., Пул У. Введение в численные методы решения дифференциальных уравнений. – М.: Наука, 1986. – 228 с.
16. Самарский А.А., Николаев Е.С. Методы решения сеточных уравнений. – М.: Наука, 1978. – 592 с.
17. Чисельні методи. Методичні вказівки та завдання до лабораторних робіт з курсу «Чисельні методи». Частина 2./ Укл. Середенко В.М., Коновалова О.М., Супруненко О.О., Красношлик Н.О. – Черкаси, ЧНУ, 2007. – 52 с.
18. Сильвестр П, Феррари Р. Метод конечных элементов для радиоинженеров и инженеров-электриков. – М.: Мир, 1989. – 229 с.