

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

М. А. Новотарський

АЛГОРИТМИ ТА МЕТОДИ ОБЧИСЛЕНЬ

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для студентів,
які навчаються за спеціальностями
121 «Інженерія програмного забезпечення»,
спеціалізацією «Програмне забезпечення високопродуктивних
комп'ютерних систем та мереж» та
123 «Комп'ютерна інженерія», спеціалізацією
«Комп'ютерні системи та мережі»*

Київ
КПІ ім. Ігоря Сікорського
2019

Рецензенти: *Романюк А. С.*, д. ф.-м. н., проф., зав. відділу теорії функцій
Інституту математики НАН України
Чемерис О. А., д. т. н., с. н. с., заступник директора з наукової
роботи Інституту проблем моделювання в енергетиці
ім. Г. Є. Пухова НАН України

Відповідальний
редактор *Стіренко С. Г.*, д. т. н., проф.

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського
(протокол № 7 від 01.04.2019 р.)
за поданням Вченої ради факультету (протокол № 8 від 04.03.2019 р.)*

Електронне мережеве навчальне видання

Новотарський Михайло Анатолійович, д-р техн. наук, с.н.с.

АЛГОРИТМИ ТА МЕТОДИ ОБЧИСЛЕНЬ

Алгоритми та методи обчислень [Електронний ресурс] : навч. посіб. для студ. спеціальностей 121 «Інженерія програмного забезпечення», спеціалізації «Програмне забезпечення високопродуктивних комп'ютерних систем та мереж» та 123 «Комп'ютерна інженерія», спеціалізації «Комп'ютерні системи та мережі» / М. А. Новотарський; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 4648 Кбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 407 с.

Навчальний посібник створений за матеріалами лекцій з курсу «Алгоритми та методи обчислень». Він поєднує традиційний підхід до вивчення сучасних алгоритмів з основними положеннями теорії алгоритмів, представленої трьома основними універсальними алгоритмічними моделями. Методи обчислень включають методи інтерполяції функцій, чисельне диференціювання та інтегрування, чисельні методи розв'язування нелінійних рівнянь та систем лінійних алгебраїчних рівнянь, ітераційні методи розв'язування лінійних і нелінійних рівнянь, ітераційні методи розв'язування звичайних диференціальних рівнянь та рівнянь з частинними похідними.

Посібник може бути корисним для інженерів та студентів технічних спеціальностей.

© М. А. Новотарський, 2019
© КПІ ім. Ігоря Сікорського, 2019

ЗМІСТ

ВСТУП	13
1. ОСНОВНІ ПОНЯТТЯ ПРО АЛГОРИТМИ	19
1.1. Інтуїтивне поняття алгоритму, різновидності, властивості та способи задавання алгоритмів	19
1.1.1. Поняття алгоритму.....	19
1.1.2. Загальні властивості алгоритмів.....	26
1.1.3. Способи задавання алгоритмів.....	27
1.1.4. Зображення алгоритму у вигляді блок-схеми.....	28
1.1.5. Види алгоритмів.....	31
1.1.6. Приклади задавання алгоритму Евкліда.....	38
1.1.7. Інші способи задавання алгоритмів.....	39
1.2. Міри складності алгоритмів	41
1.2.1. Асимптотична обчислювальна складність.....	44
1.2.2. Асимптотичні оцінки Θ , O та Ω	46
1.2.3. Властивості асимптотичних функцій.....	46
1.3. Класи задач P і NP	47
1.3.1. Алгоритми поліноміальної складності (клас P).....	47
1.3.2. Алгоритми недетермінованої поліноміальної складності (клас NP задач).....	50
1.3.3. Задача про комівояжера.....	51
1.3.4. Зведення задачі до іншої задачі.....	53
1.3.5. Типові NP задачі.....	54
1.4. Алгоритми обчислення похибок	58
1.4.1. Джерела похибок результату.....	59

1.4.2. Задачі, що виникають при роботі з наближеними величинами.....	61
1.4.3. Основні характеристики наближених чисел.....	61
1.4.4. Звідки беруться похибки?.....	64
1.4.5. Правила розрахунків похибки округлення.....	65
1.4.6. Абсолютні похибки арифметичних операцій.....	65
1.4.7. Відносна похибка суми й різниці.....	66
1.4.8. Відносні похибки добутку наближених чисел.....	68
1.4.9. Обчислення з строгим врахуванням похибок.....	69
1.4.10. Значуща цифра.....	71
1.4.11. Правила обчислення без строгого врахування похибок.....	72
1.4.12. Загальна формула для похибки.....	74
1.4.13. Приклади обчислення похибки.....	75
2. ОСНОВИ ТЕОРІЇ АЛГОРИТМІВ.....	79
2.1. Загальні поняття теорії алгоритмів.....	79
2.1.1. Формалізація поняття алгоритму через універсальні алгоритмічні моделі.....	79
2.1.2. Обчислювана функція.....	81
2.1.3. Розв'язність.....	83
2.2. Перша універсальна алгоритмічна модель.	
Рекурсивні функції.....	85
2.2.1. Примітивно рекурсивні функції.....	86
2.2.2. Частково рекурсивні функції.....	90
2.2.3. Оператор мінімізації.....	90
2.2.4. Загально рекурсивні функції.....	96

2.3. Друга універсальна алгоритмічна модель.	
Машина Тьюринга	99
2.3.1. Машина Тьюринга	99
2.3.2. Функції, обчислювані за Тьюрингом	110
2.4. Третя універсальна алгоритмічна модель.	
Нормальні алгоритми Маркова	115
2.4.1. Виникнення теорії нормальних алгоритмів	115
2.4.2. Підстановки Маркова	115
2.4.3. Нормальні алгоритми та їх застосування до слів	116
2.4.4. Нормально обчислювані функції	119
2.4.5. Принцип нормалізації Маркова	124
2.4.6. Співпадіння класу всіх нормально обчислюваних функцій з класом усіх функцій, обчислюваних за Тьюрингом	125
2.4.7. Еквівалентність різних теорій алгоритмів	125
3. МЕТОДИ ОБЧИСЛЕНЬ	127
3.1. Інтерполяція функцій	127
3.1.1. Інтерполяція й задача інтерполяції	127
3.1.2. Постановка задачі інтерполяції	128
3.1.3. Узагальнені многочлени	129
3.1.4. Інтерполяція алгебраїчними многочленами	130
3.1.5. Побудова інтерполяційного многочлена	131
3.2. Інтерполяційний многочлен Лагранжа	134
3.2.1. Постановка задачі для многочлена Лагранжа	134
3.2.2. Теорема про існування та єдиність многочлена Лагранжа ..	134
3.2.3. Скорочена форма запису многочлена	135

3.2.4. Похибка та оцінка похибки многочлена Лагранжа.....	137
3.3. Інтерполяційний многочлен Лагранжа для рівновіддалених вузлів.....	139
3.3.1. Створення многочлена Лагранжа для рівновіддалених вузлів.....	139
3.3.2. Похибка інтерполяційного многочлена Лагранжа для рівновіддалених вузлів.....	143
3.4. Обернена інтерполяція многочленом Лагранжа.....	145
3.5. Інтерполяційний многочлен Ньютона.....	148
3.5.1. Розділені різниці.....	148
3.5.2. Вираження розділених різниць через значення функцій.....	150
3.5.3. Інтерполяційний многочлен Ньютона для довільно заданих вузлів.....	152
3.5.4. Похибка інтерполяційного многочлена Ньютона.....	155
3.5.5. Поняття про скінченні різниці.....	156
3.5.6. Інтерполяційний многочлен Ньютона для рівновіддалених вузлів.....	158
3.5.7. Похибка інтерполяційного полінома Ньютона з рівновіддаленими вузлами.....	160
3.6. Сплайн-інтерполяція.....	161
3.6.1. Послідовність обчислень функції $f(x)$ методом сплайн-інтерполяції.....	165
3.7. Тригонометрична інтерполяція.....	166
3.7.1. Розв'язування задачі тригонометричної інтерполяції.....	167

3.7.2. Формули коефіцієнтів для рівновіддалених вузлів	168
3.8. Чисельне диференціювання	170
3.8.1. Постановка задачі	170
3.8.2. Застосування формули чисельного диференціювання на основі інтерполяційного полінома Ньютона з нерівновіддаленими вузлами	171
3.8.3. Застосування формули чисельного диференціювання на основі інтерполяційного полінома Ньютона з рівновіддаленими вузлами	173
3.9. Методи розв'язування нелінійних рівнянь	178
3.9.1. Постановка задачі	178
3.9.2. Етапи наближеного розв'язування нелінійних рівнянь	178
3.9.3. Метод половинного ділення	182
3.9.4. Метод пропорційних частин (метод хорд)	184
3.9.5. Метод Ньютона (метод дотичних)	192
3.9.6. Видозмінений метод Ньютона	196
3.9.7. Комбінований метод	197
3.9.8. Метод ітерації	200
3.10. Операції з матрицями та векторами	204
3.10.1. Матриці і операції з матрицями	204
3.10.2. Вектори і операції з векторами	206
3.10.3. Різновидності квадратних матриць	210
3.10.4. Визначник квадратної матриці та його властивості	212
3.10.5. Обернена матриця	220
3.10.6. Ортогональні матриці	221
3.10.7. Лінійні простори	224

3.10.8. Базис. Координати.....	225
3.10.9. Норми векторів і матриць	227
3.11. Розв’язування систем лінійних алгебраїчних рівнянь	231
3.11.1. Основні визначення	231
3.11.2. Види систем лінійних алгебраїчних рівнянь.....	232
3.11.3. Сумісні й несумісні системи алгебраїчних рівнянь.....	232
3.11.4. Множина розв’язків системи алгебраїчних рівнянь.....	233
3.11.5. Розширена матриця системи лінійних алгебраїчних рівнянь	233
3.11.6. Ранг матриці (системи).....	233
3.11.7. Базисний мінор	234
3.11.8. Теорема Кронекера-Капеллі.....	236
3.11.9. Системи лінійних алгебраїчних рівнянь із квадратною матрицею	236
3.11.10. Однорідні системи лінійних алгебраїчних рівнянь	237
3.11.11. Фундаментальна система розв’язків системи однорідних лінійних алгебраїчних рівнянь	238
3.11.12. Загальний розв’язок неоднорідної системи лінійних рівнянь	240
3.11.13. Метод виключення Гауса	242
3.11.14. Метод Гауса-Жордана	247
3.11.15. Метод квадратного кореня.....	249
3.11.16. Метод прогонки	252
3.12. Ітераційні методи розв’язування систем алгебраїчних рівнянь	255
3.12.1. Точні й ітераційні методи	255
3.12.2. Загальна схема побудови ітераційних методів	256

3.12.3. Побудова ітераційного процесу	256
3.12.4. Теорема про достатню умову збіжності ітераційного методу	257
3.12.5. Метод простої ітерації (метод Якобі)	257
3.12.6. Метод Гауса-Зейделя	260
3.12.7. Ітераційний параметр	261
3.12.8. Канонічна форма запису однокрокових ітераційних методів	261
3.12.9. Узагальнений розв'язок системи лінійних рівнянь	262
3.12.10. Стаціонарний і нестаціонарний однокрокові методи	263
3.12.11. Метод простої ітерації в координатній формі	263
3.12.12. Модифікація методу ітерацій	269
3.12.13. Достатня умова збіжності процесу ітерації в координатній формі	272
3.12.14. Метод Якобі в координатній формі	272
3.12.15. Метод Гауса-Зейделя в координатній формі	275
3.12.16. Метод релаксації в координатній формі	276
3.13. Розв'язування систем нелінійних рівнянь	280
3.13.1. Постановка задачі	280
3.13.2. Метод простої ітерації	282
3.13.3. Метод Зейделя	287
3.13.4. Метод Ньютона	290
3.14. Методи чисельного інтегрування	297
3.14.1. Однокрокові методи	297
3.14.2. Методи наближеного обчислення інтегралів	297
3.14.3. Методи Ньютона-Котеса. Постановка задачі	297

3.14.4. Метод прямокутників.....	298
3.14.5. Метод трапецій. Постановка задачі.....	300
3.14.6. Формула парабол (Симпсона).....	302
3.14.7. Сплайн-квадратура	305
3.15. Чисельний розв’язок диференціальних рівнянь	307
3.15.1. Загальні положення	307
3.15.2. Частковий розв’язок диференціального рівняння	308
3.15.3. Загальний розв’язок диференціального рівняння	308
3.15.4. Геометрична інтерпретація розв’язків диференціальних рівнянь першого порядку.....	309
3.15.5. Чисельні методи розв’язування диференціальних рівнянь..	310
3.15.6. Однокрокові методи. Метод Ейлера.....	311
3.15.7. Уточнений метод Ейлера	313
3.15.8. Метод Рунге-Кутта	313
3.15.9. Визначення однокрокових методів.....	316
3.15.10. Визначення багатокрокових методів	318
3.15.11. Порівняння методів Адамса та Рунге-Кутта	323
3.15.12. Методи прогнозу й корекції або методи предиктор-коректор.....	323
3.16. Чисельні методи розв’язування крайової задачі для звичайного диференціального рівняння (ЗДР)	328
3.16.1. Постановка задачі	328
3.16.2. Скінченно-різницевий метод розв’язування крайової задачі з граничними умовами першого роду	328
3.16.3. Скінченно-різницевий метод розв’язування крайової задачі з граничними умовами третього роду	330

3.16.4. Апроксимація диференціального рівняння скінченними різницями в загальному вигляді	333
3.16.5. Апроксимація граничних умов. Граничні умови другого роду й змішані	334
3.16.6. Метод прогонки для розв'язування систем із трьохдіагональною матрицею.....	336
3.17. Розв'язування рівнянь з частинними похідними.....	337
3.17.1. Типи рівнянь із частинними похідними.....	338
3.17.2. Перша початково-крайова задача для рівняння теплопровідності	341
3.17.3. Друга початково-крайова задача для рівняння теплопровідності.....	341
3.17.4. Третя початково-крайова задача для рівняння теплопровідності.....	342
3.17.5. Явна схема для першої початково-крайової задачі	343
3.17.6. Неявна схема для першої початково-крайової задачі.....	345
3.17.7. Підсумкова СЛАР для розв'язування методом прогонки першої початково-крайової задачі.....	346
3.17.8. Явна схема для третьої початково-крайової задачі	347
3.17.9. Неявна схема для третьої початково-крайової задачі.....	349
3.17.10. Перша початково-крайова задача для рівняння гіперболічного типу	349
3.17.11. Друга початково-крайова задача для рівняння гіперболічного типу	350
3.17.12. Третя початково-крайова задача для рівняння гіперболічного типу	350
3.17.13. Скінченно-різницева апроксимація рівнянь гіперболічного типу.....	351

3.17.14. Перша початково-крайова задача для рівняння еліптичного типу	352
3.17.15. Третя початково-крайова задача для рівняння еліптичного типу	353
3.18. Сіткові методи розв'язування диференціальних рівнянь	355
3.18.1. Розв'язування задачі Дирихле методом сіток	355
3.18.2. Процес Лібмана	359
3.18.3. Підготовка шаблонів	361
3.18.4. Приклад розв'язування крайової задачі методом сіток	363
3.19. Ітераційні асинхронні методи	367
3.19.1. Базові поняття	367
3.19.2. Метод хаотичних ітерацій	369
3.19.3. Метод асинхронних ітерацій	375
3.19.4. Метод асинхронних ітерацій з нерухомими точками	377
3.20. Метод скінченних об'ємів	382
3.20.1 Базові відомості	382
3.20.2. Розв'язування одновимірної еліптичної задачі методом МСО	383
СПИСОК ЛІТЕРАТУРИ	401

ВСТУП

Навчальний посібник розроблено на основі лекційного курсу «Алгоритми та методи обчислень», який читає автор студентам спеціальностей 121 «Інженерія програмного забезпечення» та 123 «Комп'ютерна інженерія» факультету інформатики та обчислювальної техніки (ФІОТ) Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

Метою вивчення курсу «Алгоритми та методи обчислень» є ґрунтовне вивчення сучасних методів та технологій розробки і оцінювання алгоритмів, фундаментальна підготовка студентів у виборі та використанні методів алгоритмізації, методів обчислень, стійких до похибок, ефективних методів обчислення математичних задач, аналізі одержаних наближених розв'язків, створенні високоефективних алгоритмів і програм, що враховують особливості реалізації обчислень, та сприяння розвитку логічного та аналітичного мислення студентів.

Після засвоєння навчальної дисципліни студенти мають продемонструвати знання основних методів обчислення та відповідних ефективних алгоритмів розв'язування математичних задач на комп'ютері, обчислення значень та апроксимації і інтерполяції функцій, розв'язування систем лінійних алгебраїчних рівнянь та нелінійних рівнянь, числового інтегрування та диференціювання, розв'язування диференціальних рівнянь з вибором методу. Також студенти мають уміти провести аналіз і опрацювання результатів експерименту та обчислити апостеріорні оцінки похибок.

Студент повинен набути вміння вибирати та обґрунтовувати використання на практиці тих чи інших методів обчислень, стійких до похибок і найбільш ефективних при їх практичній реалізації на комп'ютері.

Необхідно набути знань з основних принципів розробки алгоритмів та програмного забезпечення розв'язування задач на комп'ютері, досвіду дослідження обчислювальних алгоритмів, виявлення їх переваг та недоліків, вибору оптимальних алгоритмів розв'язування задач, обробки даних та розробки програм розв'язування задач.

Вивчення курсу допомагає отримати фундаментальні знання в галузі обчислювальних алгоритмів, починаючи з базових понять про обчислювальні алгоритми, принципи їх побудови, оцінки обчислювальної складності та точності і закінчуючи основами теорії алгоритмів, представленої універсальними алгоритмічними моделями, які відображають математичні аспекти даної теорії (рекурсивні алгоритмічні моделі), технічні (машина Тьюринга) та логічні (алгоритми Маркова) аспекти.

Друга частина курсу присвячена вивченню основних методів обчислень, які можуть бути корисними у інженерній практиці. Розглядаються алгебраїчні та тригонометричні способи інтерполяції функцій, чисельне диференціювання та інтегрування, чисельні методи розв'язування нелінійних рівнянь, чисельні методи розв'язування систем лінійних та нелінійних алгебраїчних рівнянь, ітераційні методи розв'язування систем алгебраїчних рівнянь.

Окрему увагу приділено задачам, які пов'язані з розв'язуванням диференціальних рівнянь. Зокрема, розглянуто чисельні методи розв'язування задачі Коші та крайові задачі для звичайних диференціальних рівнянь.

Курс також включає основні підходи до розв'язування диференціальних рівнянь з частинними похідними. Розглянуто сіткові методи, асинхронний ітераційний метод та метод скінченних об'ємів.

Перший розділ даного навчального посібника присвячено розгляду основних понять теорії алгоритмів. Зокрема, наголос робиться на тому, що

інтуїтивне визначення алгоритму істотно залежить від предметної області його застосування. Якщо алгоритми включають арифметичні дії з числовими наборами даних, то такі алгоритми називають чисельними. Як приклад типового чисельного алгоритму наведено алгоритм Евкліда для знаходження найбільшого спільного дільника для двох натуральних чисел. Для реалізації логічних перетворень з метою класифікації, сортування та ін. використовують логічні алгоритми. Прикладом логічного алгоритму є прискорене сортування Хоара. Як правило, прикладні алгоритми включають як елементи чисельних обчислень, так і логічні перетворення. Курс також включає розгляд властивостей алгоритмів та способи їх задавання. Для порівняння алгоритмів застосовують асимптотичні оцінки складності. Наведено основні теореми, властивості та приклади застосування асимптотичних оцінок Θ , O та Ω . Розглянуто клас задач, розв'язування яких потребує алгоритмів, що характеризуються неполіноміальною складністю. Дуже важливо вміти розпізнавати такі задачі. Тому у даному навчальному посібнику наведено основні типи таких задач. На завершення розділу розглянуто алгоритми обчислення похибок, що виникають при обчисленнях.

Другий розділ присвячено поглибленому вивченню природи алгоритмів шляхом знайомства з основними положеннями класичної теорії алгоритмів. Спочатку наведено основні поняття даної теорії, які включають обчислюваність, розв'язність та взаємозв'язок між ними. Розглянута перша універсальна алгоритмічна модель, яка сформована в термінах теорії рекурсивних функцій. В рамках даної теорії прийнятий *конструктивний* підхід, суть якого полягає у тому, що множина досліджуваних функцій будується зі скінченного числа базових очевидно обчислюваних функцій за допомогою операторів рекурсії, суперпозиції та мінімізації.

Друга універсальна алгоритмічна модель розглядає алгоритм як процес функціонування деякого набору технічних вузлів, які здатні

виконувати в кожний окремий дискретний момент часу примітивні операції. Такий підхід дозволяє забезпечити виконання, зокрема, таких основних властивостей послідовних алгоритмів, як детермінованість та елементарність кроків. Основною теоретичною моделлю цього типу є машина Тьюринга, на прикладі роботи якої пояснюються важливі положення згаданого підходу.

Третя універсальна алгоритмічна модель має у своїй основі набір деяких правил модифікації слів, побудованих з використанням попередньо заданого алфавіту. У цьому випадку алгоритм є послідовністю таких модифікацій з результатом роботи у вигляді результуючого слова. Для узагальнення наведено теорему, яка підтверджує той факт, що згадані універсальні моделі є рівносильними і є альтернативними способами представлення єдиної теорії алгоритмів.

Третій розділ присвячено розгляду методів обчислень, які можуть бути успішно реалізовані шляхом створення відповідних алгоритмів для виконання комп'ютерних обчислень. Розгляд таких методів традиційно починається з вивчення методів інтерполяції. Основну увагу приділено методам інтерполяції алгебраїчними многочленами. Наведено вивід інтерполяційного многочлена Лагранжа для рівновіддалених та довільно заданих вузлів. На прикладах продемонстровано способи використання даного многочлена для різних задач інтерполяції, показано приклад застосування оберненого многочлена Лагранжа. Інтерполяційний многочлен Ньютона розглядається як модифікація многочлена Лагранжа. Вивчення його включає варіант многочлена для довільно заданих вузлів та два варіанти даного многочлена для довільно віддалених вузлів з інтерполяцією, що починається з початкових вузлів та інтерполяцією з кінцевих вузлів. Для згаданих алгебраїчних многочленів та їх модифікацій розглянуто теореми, що визначають способи визначення похибки та способи її оцінки на заданому відрізку. В рамках теми інтерполяції функцій

розглянута сплайн-інтерполяція на прикладі інтерполяції сплайном третього порядку та основні підходи до інтерполяції тригонометричними функціями.

У даному розділі розглянуто основні чисельні методи розв'язування нелінійних рівнянь, застосування яких зводиться до ізоляції кореня з подальшим його уточненням. Серед методів уточнення ізольованого кореня на заданому відрізку в даному курсі вивчаються такі методи: метод половинного ділення, метод пропорційних частин, метод Ньютона та його модифікації і метод ітерацій. Для успішного оволодіння методами розв'язування систем лінійних алгебраїчних рівнянь (СЛАР) у навчальному посібнику наведено короткий огляд операцій з матрицями та векторами. Згадані питання використано при вивченні основ теорії СЛАР, що, зокрема, включають огляд видів СЛАР, поняття про множину розв'язків, ранг системи та ін. Розглядаються комп'ютерні версії методів Гауса та Гауса-Жордана, методів квадратного кореня та прогонки для отримання точних розв'язків СЛАР.

Вивчення ітераційних методів розв'язування СЛАР включає розгляд схеми побудови ітераційних методів та умову їх збіжності. При розв'язуванні СЛАР особлива увага приділена методам Гауса-Зейделя, Якобі та методу релаксації, а для розв'язування систем нелінійних рівнянь – методам простої ітерації, Зейделя та Ньютона.

Важливими темами, які розглядаються в рамках вивчення обчислювальних методів, є чисельне диференціювання та інтегрування. Чисельне диференціювання представлено на основі формул, одержаних шляхом застосування варіантів полінома Ньютона як для рівновіддалених вузлів, так і для вузлів з довільним розташуванням. Чисельне інтегрування представлено групою методів Ньютона-Котеса. Розглядаються також сплайн-квадратура та застосування формули парабол.

Курс завершується розглядом видів диференціальних рівнянь та тих задач, які виникають у процесі їх розв'язування. Такий розгляд починається з задачі Коші, розв'язок якої для звичайних диференціальних рівнянь представлено однокроковим методом Ейлера з різними модифікаціями, методом Рунге-Кутта та групою багатокрокових методів. Детально розглянуто всі аспекти розв'язування крайової задачі для звичайного диференціального рівняння. В ході цього розгляду вивчаються види граничних умов та використання їх при апроксимації диференціального рівняння. Описано застосування методу прогонки для розв'язування систем лінійних алгебраїчних рівнянь з трьохдіагональною матрицею, яка зазвичай виникає при апроксимації диференціального рівняння скінченними різницями.

Розв'язування крайових задач для рівнянь з частинними похідними в залежності від виду граничних умов представлено першою, другою та третьою початково-крайовою задачею. Детально розглянутий процес підготовки та розв'язування типових задач ітераційними асинхронними методами та методом скінченних об'ємів.

1. ОСНОВНІ ПОНЯТТЯ ПРО АЛГОРИТМИ

1.1. Інтуїтивне поняття алгоритму, різновидності, властивості та способи задавання алгоритмів

Розв'язування будь-якої задачі з застосуванням комп'ютера необхідно розбити на наступні етапи:

- розробка алгоритму розв'язування задачі;
- складання програми розв'язування задачі алгоритмічною мовою;
- ввід програми в комп'ютер;
- налагодження програми (виправлення помилок),
- виконання програми на комп'ютері;
- аналіз отриманих результатів.

Розглянемо перший етап розв'язування задачі – розробку алгоритму.

1.1.1. Поняття алгоритму

З давніх часів у математиці алгоритм уявляли як формальну інструкцію, яка визначає сукупність операцій і порядок їх виконання для розв'язування всіх задач деякого одного типу.

Для розв'язування ряду однотипних задач іноді доцільно використовувати суто механічні обчислювальні процеси. За їх допомоги шукані величини обчислюють послідовно з даних величин за певними правилами. Опис таких процесів прийнято називати алгоритмами. Взагалі, під алгоритмом інтуїтивно розуміють деякий формальний припис, діючи згідно з яким можна одержати розв'язок задачі.

Кожний зустрічається з алгоритмами зі школи. Правила, за якими виконують арифметичні дії з багатозначними числами, є найпростішими прикладами алгоритмів. Термін «алгоритм» походить від імені

стародавнього узбецького математика Мухаммеда бен Муса аль Хорезмі, який ще в IX столітті сформулював такі правила.

У своєму розвитку математика нагромадила величезну кількість різних алгоритмів. Отримуючи відповідну інтерпретацію в конкретних застосуваннях, вони становлять значну й найбільш істотну частину математичного апарата, використовованого в техніці.

Алгоритм – чіткий опис послідовності дій, які необхідно виконати при розв’язуванні задачі [1].

Можна сказати, що алгоритм описує процес перетворення вхідних даних у результати, оскільки для отримання розв’язку будь-якої задачі необхідно:

увести вхідні дані;

перетворити вхідні дані в результати (вихідні дані);

вивести результати.

Розробка алгоритму розв’язування задачі – це розбиття задачі на послідовно виконувані етапи, причому результати виконання попередніх етапів можуть використовуватися при виконанні наступних. При цьому повинні бути чітко зазначені як зміст кожного етапу, так і порядок виконання етапів. Окремий етап алгоритму є або іншою, більш простою задачею, алгоритм розв’язування якої відомий (розроблений заздалегідь), або повинен бути досить простим і зрозумілим без пояснень.

У наш час виникла потреба у глибокому вивченні й осмисленні алгоритмів, головним чином у зв’язку з проблемою алгоритмічної нерозв’язності. Річ у тім, що при спробах розв’язати ряд задач виникли труднощі, які не вдалося подолати, незважаючи на довготривалі й завзяті зусилля багатьох великих математиків.

Згодом з’ясувалось, що існує коло задач, які алгоритмічно важко розв’язні (так звані *NP*-повні задачі).

Для вирішення проблем, що виникають при розробці алгоритмів, зокрема, для ефективного визначення *NP*-повних задач, у 30-ті роки була створена нова наука – «Теорія алгоритмів».

Чисельні та логічні алгоритми

Алгоритми можуть бути чисельними й логічними.

Алгоритми, які представляють розв'язування задачі у вигляді арифметичних дій над числами, називають *чисельними алгоритмами*.

Типовим прикладом чисельного алгоритму є алгоритм Евкліда для знаходження найбільшого спільного дільника двох заданих додатних цілих чисел a і b [2].

Довідково. Найбільшим спільним дільником (НСД) чисел a і b називають деяке найбільше число c , на яке обидва числа a і b діляться без остачі.

Опис алгоритму Евкліда

Алгоритм Евкліда складається з наступної системи послідовних вказівок:

1. Вводимо числа a і b .
2. Перевіряємо умову $a=b$. Якщо ця умова вірна, то розв'язок знайдений. Обидва числа є НСД одне для одного. Переходимо до пункту 6. Якщо $a \neq b$, то переходимо до 3.
3. Якщо $a < b$, то значення a та b міняємо місцями: $a \leftrightarrow b$.
4. Якщо a ділиться без остачі на b , то число b і є НСД. Переходимо до пункту 6, інакше переходимо до пункту 5.
5. Число a заміняємо на остачу від ділення і переходимо до пункту 3.
6. Вивід результатів.

Приклад 1.1. *Застосування алгоритму Евкліда*

Завдання. Знайти НСД для 30 і 18.

Розв'язок. $30/18 = 1$ (остача 12).

$$18/12 = 1 \text{ (остача 6).}$$

$$12/6 = 2 \text{ (остача 0).}$$

Кінець: НСД – це дільник. НСД (30, 18) = 6.

Отже, після п'ятої вказівки потрібно щоразу повертатися до третьої, доки не буде виконана четверта вказівка. Хоча заздалегідь не відомо, скільки буде потрібно таких циклічних переходів, але ясно, що для будь-яких двох чисел мета буде досягнута за скінченну кількість кроків.

Логічні алгоритми, як правило, полягають у маніпулюванні даними з метою встановлення або визначення деякої закономірності. До логічних алгоритмів можна віднести алгоритми класифікації, сортування, планування та ін. [3].

Приклад 1.2. *Логічний алгоритм.* Найпростішим прикладом логічного алгоритму є алгоритм сортування. Розглянемо, наприклад, сортування Хоара.

I етап алгоритму Хоара

Номер кроку	$i \rightarrow$					$\leftarrow j$	Примітки
	0	1	2	3	4	5	
	40	11	83	21	75	64	Початковий список
1	40					64	$k_0 < k_j$ ОК
2	40				75		$k_0 < k_j$ ОК
3	40			21			$k_0 > k_j$ Обмін
	21			40			$k_0 > k_i$ ОК
3	21	11	83	40	75	64	Примітки
4		11		40			$k_0 > k_i$ ОК
5			83	40			$k_0 < k_i$ Обмін
			40	83			$k_0 > k_i$ ОК
6	21	11	40	83	75	64	Кінець етапу

II етап алгоритму Хоара

Номер кроку	$i_1 \rightarrow$		Примітки	$i_2 \rightarrow$				Примітки
	0	1		0	1	2	3	
	21	11	Поч. спис.	40	83	75	64	Поч. спис.
7	21	11	$k_0 > k_j$ Обмін	40	83		64	$k_0 > k_j$ Обмін
	11	21	$k_0 > k_i$ ОК	40	64		83	$k_0 > k_i$ ОК
8				40		75	83	$k_0 > k_i$ ОК
9	11	21	Кінець алг.	40	64	75	83	Кінець алг.

Рис. 1.1. Алгоритм сортування Хоара

Прийmemo перший елемент послiдовностi за базовий ключ, видiлимо його червоним кольором i позначимо $k_0 = 40$. Установимо два покажчики i та j , з яких i починає вiдлiк лiворуч ($i=1$), а j – праворуч ($j=n$). Порiвнюємо базовий ключ k_0 i поточний ключ k_j . Якщо $k_0 < k_j$, то встановлюємо $j := j - 1$ й проводимо наступне порiвняння k_0 й k_j . Продовжуємо зменшувати j доти, поки не досягнемо умови $k_0 > k_j$. Пiсля цього мiняємо мiсцями ключi k_0 й k_j (див. крок 3 на рис. 1.1).

Тепер починаємо змiнювати iндекс $i := i + 1$ i порiвнювати елементи k_i й k_0 . Продовжуємо збiльшення i доти, поки не досягнемо умови $k_i > k_0$, пiсля чого вiдбувається обмiн k_i i k_0 (див. крок 5). Знову вертаємося до iндексу j , зменшуємо його. Почергово зменшуючи j та збiльшуючи i , продовжуємо цей процес iз обох кiнцiв до середини доти, поки не одержимо $i = j$ (див. крок 6).

Уже на першому етапi наявнi два факти:

по-перше, базовий ключ $k_0 = 40$ зайняв своє постiйне мiсце у послiдовностi для сортування;

по-друге, усi елементи лiворуч вiд k_0 будуть меншими за нього, а праворуч – бiльшими за нього.

Таким чином, по закiнченнi першого етапу маємо:

21, 11, 40, 83, 75, 64

Права частина

Лiва частина

Етап 7 починають з вибору базових ключiв окремо для лiвої й правої частини. Потiм виконують незалежно два сортування: сортування лiвої частини й сортування правої частини.

Послідовні та паралельні алгоритми [4]

Послідовний алгоритм – це алгоритм, у якому кожний наступний етап використовує результати деякої підмножини попередніх етапів. Традиційний підхід одержання паралельного алгоритму з послідовного полягає в пошуку незалежних ділянок з метою організації їх паралельного виконання.

Приклад 1.3. Послідовний алгоритм

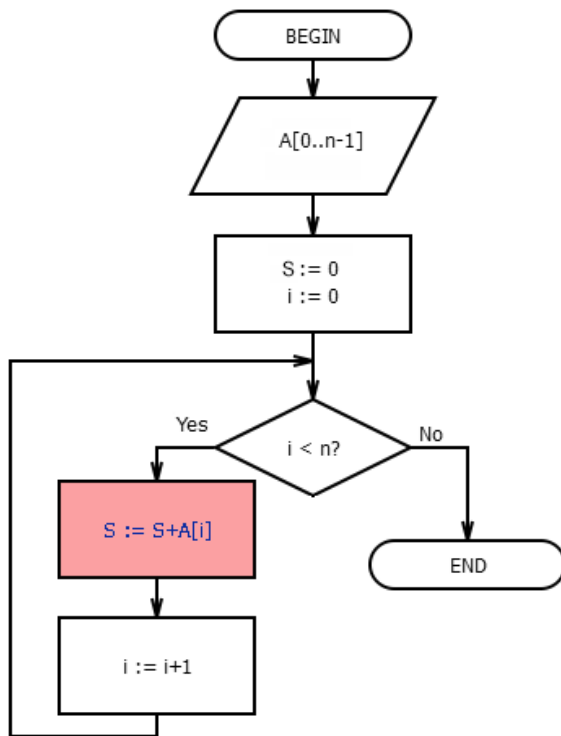


Рис. 1.2. Послідовний алгоритм додавання n чисел

Нехай даний звичайний алгоритм підсумовування (додавання) n чисел, коли на кожному кроці до часткової суми додають черговий доданок (рис. 1.2). Такий алгоритм є виключно послідовним. Критичною операцією даного алгоритму є додавання чергового числа до проміжної суми: $S := S + A[i]$.

Додавання даного числа може виконуватися тільки після завершення операції підсумовування з попереднім.

Даний простий приклад показує, що

не завжди для одержання паралельного алгоритму можна використовувати методику розпаралелювання, що полягає в пошуку незалежних ділянок. Якщо розпаралелювання виявилось нерезультативним, то це ще не означає, що принципово не можна одержати результат, використовуючи паралельні обчислення.

Паралельний алгоритм – алгоритм, який може бути реалізований по частинах на різних обчислювальних засобах з наступним об'єднанням часткових результатів для одержання глобального розв'язку задачі [5].

Приклад 1.4. Паралельний алгоритм

Розглянемо приклад побудови паралельного варіанта алгоритму підсумовування n чисел (рис. 1.3). Розіб'ємо всі доданки на пари й здійснимо підсумовування двох чисел усередині кожної пари. Усі ці операції незалежні. Отримані часткові суми також розіб'ємо на пари й знову здійснимо підсумовування двох чисел усередині кожної пари. Знову всі операції незалежні. Уся сума буде отримана через $\log_2 n$ кроків.

У даному алгоритмі вже є значний ресурс паралелізму, хоча він не рівномірний. На першому часовому кроці можуть бути використані $n/2$ процесорів, на другому $n/4$ і т. д. Названий цей алгоритм процесом здвоювання.

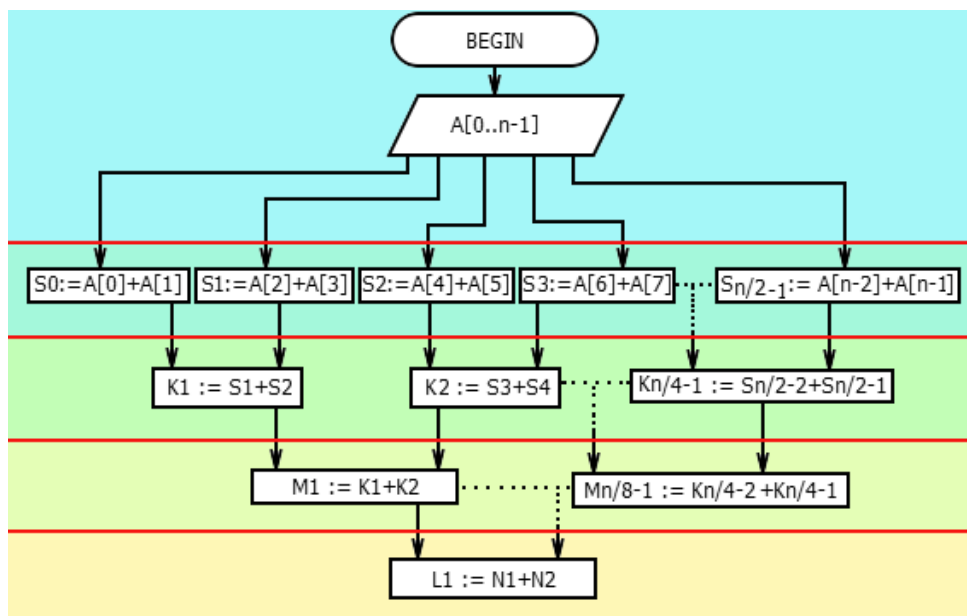


Рис. 1.3. Паралельний алгоритм додавання n чисел

Зазначимо, що обидва алгоритми ґрунтуються на реалізації математично еквівалентних виразів підсумовування чисел, але вони мають

різні властивості, принаймні, з точки зору паралельних обчислень. Насправді, у них багато й інших відмінностей: вони по-різному реагують на помилки округлення, по-різному використовують пам'ять і т. п. Тому ці алгоритми слід вважати принципово різними, незважаючи на те, що вони математично еквівалентні!

Саме цей факт став причиною активізації досліджень в області розробки нових обчислювальних методів, орієнтованих на реалізацію в паралельних обчислювальних системах, і розвитку теорії паралельних алгоритмів, які базуються на паралельних методах.

1.1.2. Загальні властивості алгоритмів

Досвід розробки й застосування алгоритмів підказує ряд загальних властивостей будь-яких алгоритмів:

1. **Дискретність алгоритму:** будь-який алгоритм можна розглядати як процес послідовної побудови величин, що йде в дискретному часі, виконуючи набір певних інструкцій, який називають програмою.

2. **Масовість алгоритму:** алгоритм придатний для розв'язування не конкретної визначеної однієї задачі, а цілого класу однотипних задач. У цьому аспекті таблиця множення не є алгоритмом, а множення стовпчиком багатозначних чисел – алгоритм. Наприклад, в алгоритмі Евкліда числа a і b вибирають з нескінченної зліченної множини цілих чисел.

3. **Детермінованість алгоритму:** після виконання чергового етапу однозначно визначено, що робити на наступному етапі.

4. **Елементарність кроків алгоритму:** розв'язування задачі розбивають на етапи, кожний з яких повинен бути простим і локальним. Це означає, що відповідна операція повинна бути елементарною для виконавця алгоритму (людини або машини). Наприклад, операції, наявні в алгоритмі Евкліда, – порівняння, ділення та перестановка чисел – є досить

стандартними й звичними. У той же час сам алгоритм Евкліда може бути елементарною операцією більш складного алгоритму.

5. Результативність алгоритму: алгоритм через скінченну кількість кроків повинен привести до зупинки, яка свідчить про досягнення необхідного результату. Так, при пошуку шляху в лабіринті зупинка настає або на досягнутому майданчику, або при поверненні до місця старту, якщо зазначена мета не досягнута. Зокрема, кожен, хто пред'являє алгоритм розв'язування деякої задачі, зобов'язаний показати, що алгоритм зупиняється після скінченної кількості кроків. Ще говорять, що алгоритм збігається для будь-якого x з області визначення функції $f(x)$. Однак перевірити результативність (збіжність алгоритму) набагато важче, ніж інші його властивості.

1.1.3. Способи задавання алгоритмів

Існує досить велика кількість формальних засобів, що дозволяють однозначно задавати сучасні алгоритми. Однак у рамках даного курсу будемо розглядати тільки традиційні, найпоширеніші способи:

1. Спосіб задавання алгоритму у вербальній (словесній) формі

Це найдавніший спосіб задавання алгоритмів. Перші алгоритми формулювали винятково словесно. Прикладом словесної форми задавання може служити алгоритм Евкліда, розглянутий вище. Як правило, словесний опис складається з пунктів, у кожному з яких чітко формулюють найпростішу дію. Потім дають вказівку про подальші дії.

2. Спосіб задавання алгоритму в графічній формі

Блок-схема – це графічне зображення алгоритму. При її побудові зміст кожного кроку алгоритму записують у довільній формі усередині блоку, представленого геометричною фігурою. Порядок виконання кроків вказують за допомогою стрілок, що з'єднують блоки. Використання різних геометричних фігур відображає різний характер виконуваних дій.

3. Спосіб задавання алгоритму у вигляді псевдокоду

Псевдокод – це опис алгоритмів умовною алгоритмічною мовою, що включає елементи мови програмування, фрази природної мови, загальноприйняті математичні позначення й ін. Такий спосіб опису часто використовується в навчальній і науковій літературі з метою забезпечення компактного представлення алгоритму.

4. Спосіб задавання алгоритму у вигляді комп'ютерної програми

Для задавання алгоритму у вигляді програми використовують конструкції конкретної мови програмування. Представлений у вигляді програми алгоритм після компіляції перетворюється на файл, що виконується комп'ютером.

1.1.4. Зображення алгоритму у вигляді блок-схеми

Блок-схемою називають наочне графічне зображення алгоритму, у якому окремі етапи алгоритму зображують за допомогою різних геометричних фігур – блоків, а зв'язки між етапами (послідовність виконання етапів) вказують за допомогою стрілок, які з'єднують ці фігури. Конфігурація й розміри блоків, а також порядок графічного оформлення блок-схем регламентовані ДЕРЖСТАНДАРТ 10.002-80 і ДЕРЖСТАНДАРТ 10.003-80 «Схеми алгоритмів і програм». Блоки супроводжують написами.

У таблиці 1.1 наведені найбільш часто використовувані блоки, зображені елементи зв'язків між ними й дано коротке пояснення до них. Блоки й елементи зв'язків називають елементами блок-схем.

Представлених у таблиці елементів цілком достатньо для зображення алгоритмів, які необхідні при виконанні студентських робіт.

При з'єднанні блоків слід використовувати тільки вертикальні й горизонтальні лінії потоків.

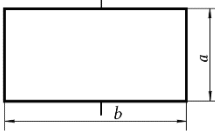
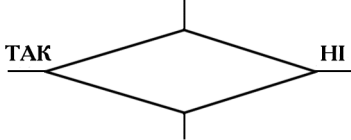
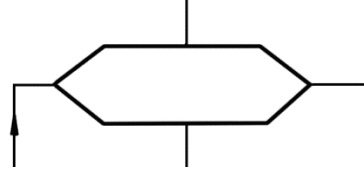
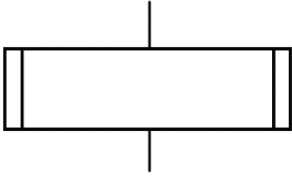
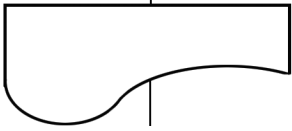
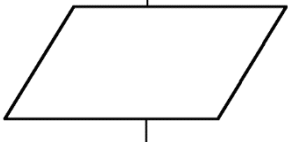
Горизонтальні потоки, що мають напрямок праворуч ліворуч (справа наліво), і вертикальні потоки, що мають напрямок знизу вгору, повинні бути обов'язково позначені стрілками.


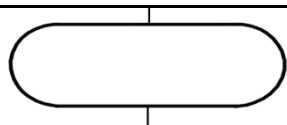
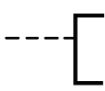
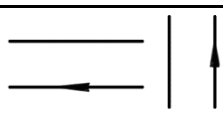
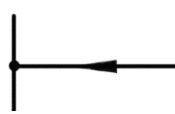

Інші потоки можуть бути позначені або залишені непозначеними.

Лінії потоків мають бути паралельні лініям зовнішньої рамки або границям аркуша.

Таблиця 1.1

Основні елементи блок-схем

Назва	Елемент	Коментар
Процес		Обчислювальна дія або послідовність обчислювальних дій
Розв'язування		Перевірка умови
Модифікація		Заголовок циклу
Визначений процес		Звертання до процедури
Документ		Вивід даних, друк даних
Введення/Виведення		Ввід/Вивід даних

Назва	Елемент	Коментар
З'єднувач		Розрив лінії потоку
Початок, Кінець		Початок, кінець, пуск, зупинка, вхід і вихід у допоміжних алгоритмах
Коментар		Використовують для розміщення написів
Горизонтальні й вертикальні потоки		Лінії зв'язків між блоками, напрямки потоків
Злиття		Злиття ліній потоків
Міжсторінковий з'єднувач		Перехід на інший лист

Відстань між паралельними лініями потоків повинна бути не менше 3 мм, між іншими елементами схеми – не менше 5 мм.

Горизонтальний і вертикальний розміри блоку мають бути кратні 5 (ділитися на 5 націло). Співвідношення горизонтального й вертикального розмірів блоку $b/a = 1.5$ є основним. При ручному виконанні креслення блоку припустиме співвідношення $b/a = 2$.

Блоки «Початок», «Кінець» і «З'єднувач» мають висоту $a/2$, тобто вдвічі меншу за основну висоту блоків.

Для розміщення блоків рекомендується поле аркуша розбивати на горизонтальні й вертикальні (для схем, що розгалужуються) зони.

Для зручності опису блок-схеми кожний її блок слід пронумерувати, використовувати наскрізну нумерацію блоків. Номер блоку розташовують у розриві в лівій верхній частині рамки блоку.

1.1.5. Види алгоритмів

По характеру зв'язків між блоками розрізняють алгоритми лінійної, циклічної структури, та структури, що розгалужується.

Лінійний алгоритм

Лінійний алгоритм – це алгоритм, у якому блоки виконуються послідовно зверху вниз від початку до кінця.

На рис. 1.4 наведений приклад блок-схеми алгоритму обчислення периметра P і площі S квадрата зі стороною довжини A .

Блок-схема алгоритму складається із шести блоків. Виконання алгоритму починається із блоку 1 «Старт». Цей блок символізує включення комп'ютера, налаштування його на виконання алгоритму й виділення пам'яті під усі змінні, які задіяні в алгоритмі. В алгоритмі рис. 1.4 таких змінних три: A , P , S . Отже, під кожен зі змінних алгоритмом буде виділено по одній комірці пам'яті. На цьому блок 1 буде відпрацьований.

Як видно з рис. 1.4, блок 1 зв'язаний вертикальною лінією потоку з блоком 2. Ця лінія не має стрілки, що вказує напрямок потоку. Отже, цей потік спрямований униз. Таким чином, після виконання блоку 1 керування буде передано на блок 2. Блок 2 «Введення даних» (див. табл. 1.1) показує, що змінній A слід присвоїти значення. Це означає, що в комірку, відведену комп'ютером під цю змінну, потрібно помістити константу. На реальному комп'ютері ця константа може бути введена у різний спосіб. Спосіб залежить від того, як запрограмований даний фрагмент. Можна, наприклад, передбачити введення константи з клавіатури або одержати його із задалегідь підготовленого файлу. Можливо, ця константа буде отримана через зовнішні джерела даних, наприклад, від фізичної установки, підключеної до комп'ютера.

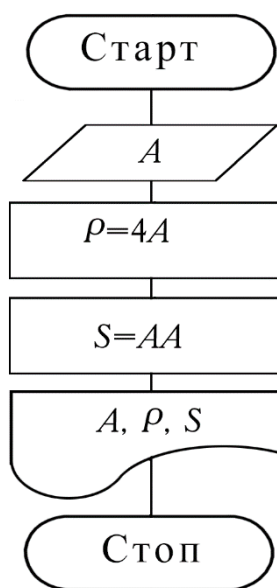


Рис. 1.4. Приклад лінійного алгоритму

Для даного прикладу спосіб передачі константи не має значення, важливо лише те, що при виконанні блоку 2 у комірку з адресою A буде занесена конкретна константа. Нехай такою константою є число 5.

Далі керування по лінії потоку передається до блоку 3 «Процес». У цьому блоці при виконанні розміщеної в ній команди число 4 множиться на константу, поміщену в комірку A (тобто 5), і результат (тобто 20) присвоюється змінній P (тобто константа 20 записується в комірку за адресою P). Після виконання цих операцій керування передається до блоку 4.

У блоці 4 аналогічним чином проводиться множення значень змінної A і результат (константа 25) присвоюється змінній S (у комірку за адресою S буде занесена константа 25). Після цього виконується перехід до блоку 5.

При виконанні команд блоку 5 виводяться (наприклад, на екран, папір, у зовнішній файл і т. д.) значення змінних A , P , S , які збереглися у відповідних комірках до цього моменту. Зрозуміло, що для конкретного прикладу $A = 5$ будуть виведені константи 5, 20, 25, тобто довжина сторони

квадрата, його периметр і площа. Далі керування передається останньому блоку 6.

У блоці 6 «Стоп» проводиться звільнення комірок пам'яті, які були зарезервовані під змінні A , P , S , і алгоритм закінчує роботу.

Алгоритми, що розгалужуються

На практиці алгоритми лінійної структури трапляються вкрай рідко. Частіше необхідно організувати процес, який залежно від деяких умов проходить по тій або іншій гілці алгоритму. Такий алгоритм називають алгоритмом, що розгалужується.

У блок-схемах розгалуження починається на виходах елемента «Розв'язок», за допомогою якого в алгоритмі виконується перевірка будь-якої умови. Кількість гілок тим більша, чим більше умов, що перевіряються.

Для пояснення розглянемо розв'язування задачі знаходження значення функції $z = x/y$.

На перший погляд здається, що алгоритм розв'язування цієї задачі має лінійну структуру. Однак, якщо врахувати, що ділити на нуль не можна через переповнення комірки, то алгоритм потребує ускладнення.

По-перше, потрібно з обчислень виключити варіант $x = 0$.

По-друге, потрібно проінформувати користувача алгоритму про помилку, яка виникла.

Якщо цього не зробити, то при обчисленнях може виникнути аварійний вихід до одержання результату. У професійній практиці аварійні завершення вкрай небажані, оскільки до цього моменту вже може бути накопичена певна кількість результатів, які виявляться неопрацьованими й просто пропадуть. Можна привести інші приклади, коли аварійна зупинка комп'ютера може спричинити набагато серйозніші наслідки.

Розв'язок задачі представлений блок-схемою на рис. 1.5.

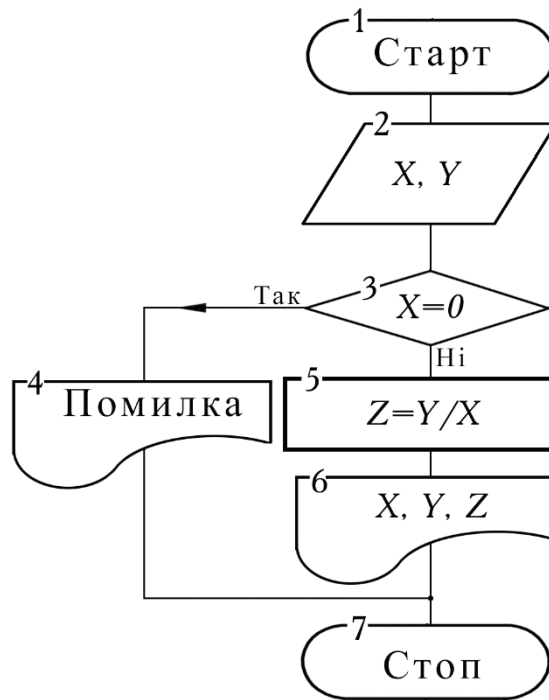


Рис. 1.5. Алгоритм, що розгалужується

Блок-схема складається з 7 блоків. Після початку роботи алгоритм у блоці 2 вимагає введення аргументів X і Y . Потім у блоці 3 проводиться перевірка умови $X=0$. Тут алгоритм перевіряє, чи дорівнює нулю константа, уведена в комірку з адресою X . Результатом такої перевірки є відповідь «Так» або «Ні». Залежно від цієї відповіді виконання алгоритму піде по одній або іншій гілках. Якщо результат перевірки виявиться негативним, то на X можна ділити, і керування передається блоку 4.

У блоці 4 буде отриманий результат Z , потім у блоці 6 значення всіх трьох змінних будуть надруковані й у блоці 7 алгоритм закінчить роботу. Якщо ж відповідь виявиться позитивною, то керування буде передано блоку 4. Виконуючи команду блоку 4, алгоритм виведе повідомлення «Помилка» і потім закінчить роботу в тому ж блоці 7.

Циклічний алгоритм

Часто при розв'язуванні задач доводиться повторювати виконання операцій по тих самих залежностях при різних значеннях змінних, що

входять у ці залежності, і робити багаторазовий прохід по тих самих ділянках алгоритму. Такі ділянки називають циклами.

Цикл – різновид керуючої конструкції, призначеної для організації багаторазового виконання одних і тих самих ділянок алгоритму при різних значеннях змінних.

Алгоритми, що містять цикли, називають *циклічними*.

Використання циклів суттєво скорочує обсяг алгоритму. Етапи організації циклу:

- підготовка (ініціалізація) циклу;
- виконання обчислень циклу (тіло циклу);
- модифікація параметрів;
- перевірка умови закінчення циклу.

Розрізняють цикли з наперед відомою і наперед невідомою кількістю проходів.

Цикл називають *детермінованим*, якщо число повторень тіла циклу заздалегідь відомо або визначено.

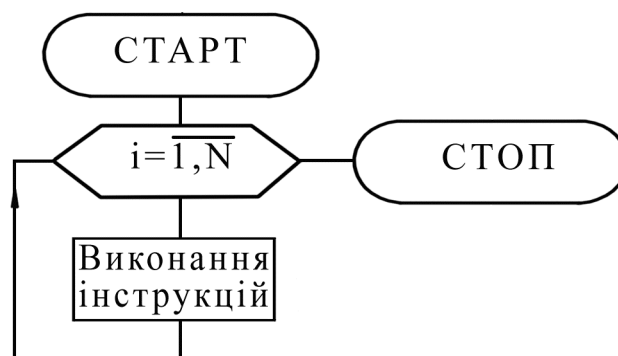


Рис. 1.6. Алгоритм з детермінованим циклом

Цикл називають ітераційним, якщо число повторень тіла циклу заздалегідь невідомо, а залежить від значень параметрів (деяких змінних), що беруть участь в обчисленнях.

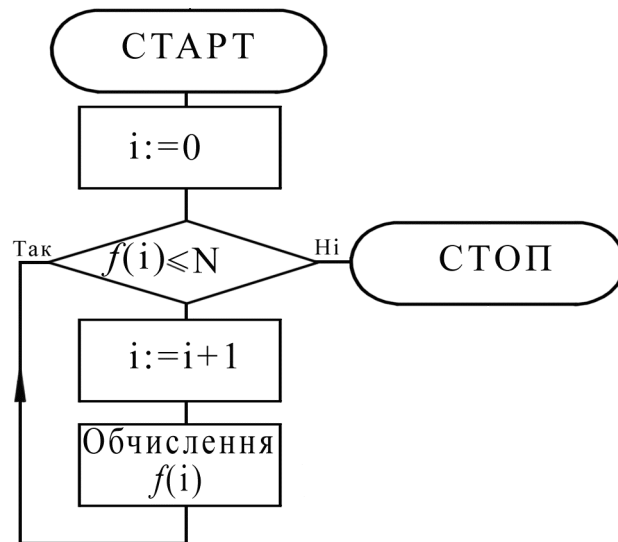


Рис. 1.7. Алгоритм із ітераційним циклом

Приклад 1.5. Розглянемо приклад алгоритму з циклом, що має наперед невідому кількість проходів. Для цього розв’яжемо наступну задачу.

Указати найменшу кількість членів ряду натуральних чисел $1, 2, 3, \dots$, сума яких більша за число K .

Блок-схема алгоритму розв’язування цієї задачі наведена на рис. 1.8. Вона складається з восьми блоків.

Після початку роботи в блоці 2 вводиться значення числа K . Далі в блоці 3 змінна i одержує значення 1, тобто значення, з якого почнеться відлік натуральних чисел. Змінна S , призначена для накопичення суми цих чисел, перед початком підсумовування одержує значення 0. Після цього керування передається блоку 5.

У ньому при виконанні команди $S = S + i$ проводиться додавання вмісту комірок S та i , а результат записується в комірку S . Оскільки до операції додавання було $S = 0$, $i = 1$, то після виконання операції $S = 1$. При записі нового значення старий вміст комірки S (нуль) стирається, а на його місце записується число 1.

Зауважимо, що якби до цієї операції в блоці 3 не була виконана команда $S = 0$ (записати нуль у комірку S), то при обчисленні суми $S+1$ виникла б помилка, оскільки із комірки S була б взята константа, яка з'явилася там після розподілу пам'яті.

Після підсумовування першого члена послідовності в блоці 6 виконується перевірка умови про перевищення сумою S заданого числа K .

Якщо умова б не виконається, то проводиться перехід до блоку 4, де при виконанні операції значення змінної збільшується на 1 і дорівнює 2. Тепер алгоритм знову повернеться до блоку 5 і до старого значення суми додасть новий член 2. Після цього сума дорівнюватиме 3.

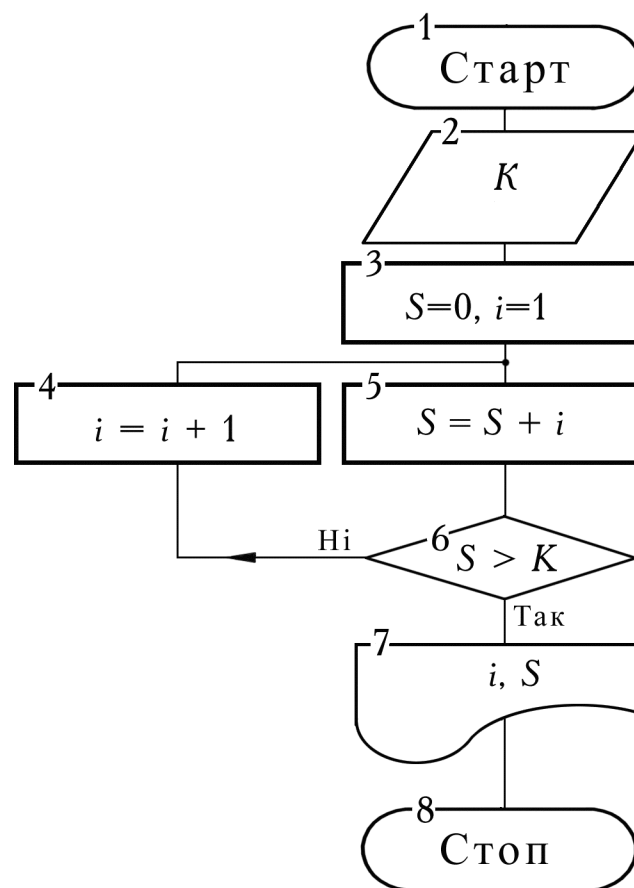


Рис. 1.8. Приклад циклічного алгоритму

У блоці 6 знову перевіряється умова одержання необхідної суми і т. д. Ланцюжок блоків 4–5 буде оброблятися знову й знову до того моменту, коли

одного разу при певному значенні змінної i , нарешті, виконається умова $S > K$, тобто сума, що накопичується в такому циклі, уперше перевищить задане значення K . Змінна i , значення якої при черговому проході ланцюжка цих блоків збільшується на 1, відіграє роль лічильника цього циклу.

Далі проводиться перехід до блоку 7, де надрукується значення кількості членів ряду (взяте й надрукуване число із комірки i , яке там зберігається в момент виконання умови), суми S та в блоці 8 алгоритм закінчить роботу.

1.1.6. Приклади задавання алгоритму Евкліда

Блок-схема алгоритму Евкліда

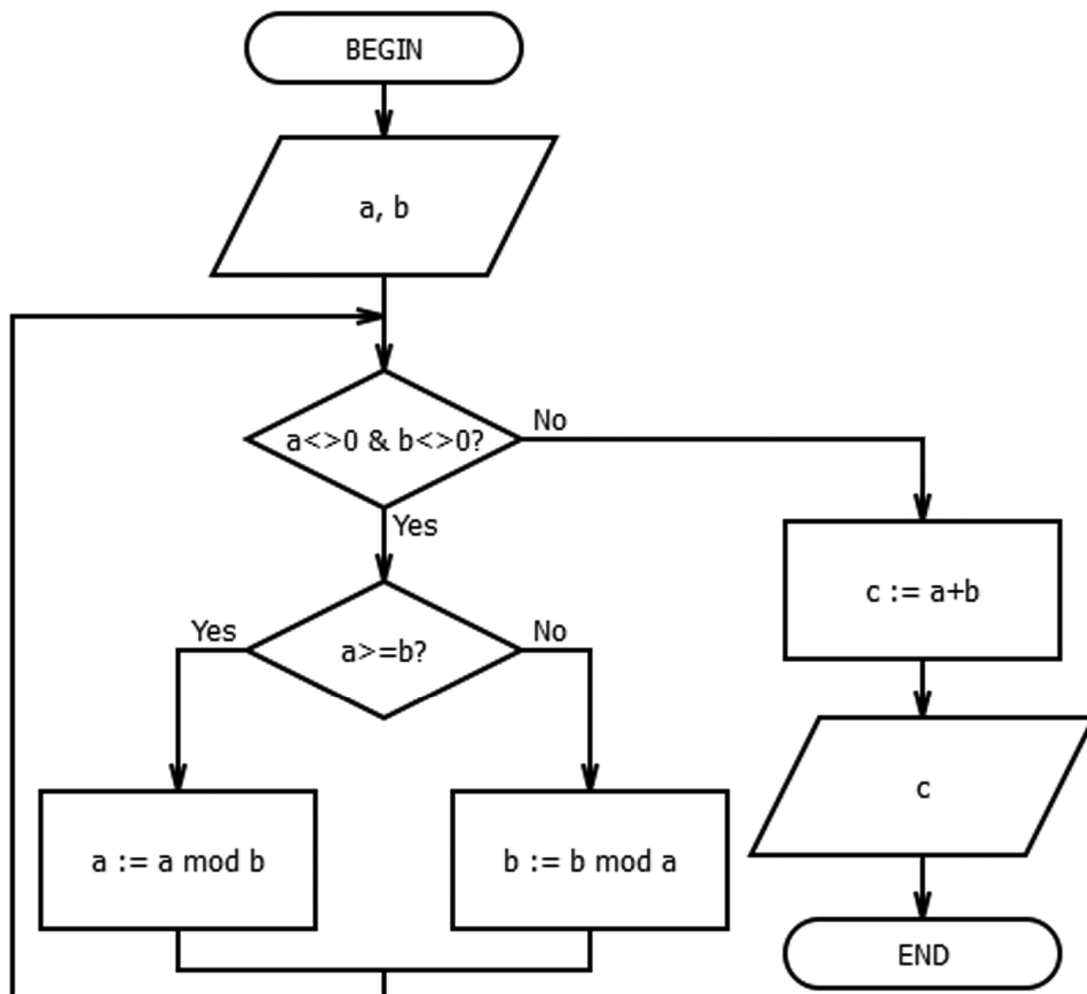


Рис. 1.9. Блок-схема алгоритму Евкліда

Псевдокод алгоритму Евкліда

1. *Input* a, b .
2. If $a < b$ then $a \leftrightarrow b$ else знайти q, r з $a = bq + r$.
3. If $r = 0$ then $c := b$ else $a := b; b := r; \text{goto } 2$.

Приклад 1.6. Програми за алгоритмом Евкліда

<pre>C++ while (b) { a %= b; swap (a, b); } gcd = a;</pre>	<pre>Python def gcd(a, b): return b and gcd(b, a%b) or a</pre>
<pre>Java public class GCD { public static int gcd(int a, int b) { while (b != 0) { int tmp = a%b; a = b; b = tmp; } return a; } }</pre>	

1.1.7. Інші способи задавання алгоритмів

Мережі Петрі

Мережу Петрі задають у вигляді кортежу [6]:

$$\Phi = (P, T, F, M),$$

де $P = \{p_1, p_2, \dots, p_n\}$ – скінченна множина позицій, $n > 0$;

$T = \{t_1, t_2, \dots, t_m\}$ – скінченна множина переходів, $m > 0$;

$F \subseteq \{P \times T\} \cup \{T \times P\}$ – скінченна множина ребер, які з'єднують

переходи й позиції;

M – вектор маркувань.

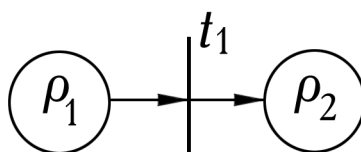


Рис. 1.10. Найпростіша мережа Петрі

На рис. 1.11 наведена мережа Петрі з такими параметрами:

$F = I \cup O$, $I \subset \{P \times T\}$ – підмножина вхідних ребер;

$O \subset \{T \times P\}$ – підмножина вихідних ребер;

$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9\}$

$T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$

$I = \{(p_1, t_1), (p_2, t_2), (p_3, t_3), (p_6, t_4), (p_4, t_4), (p_8, t_4), (p_5, t_5), (p_7, t_6)\}$

$O = \{(t_1, p_2), (t_1, p_3), (t_1, p_4), (t_2, p_5), (t_3, p_6), (t_4, p_7), (t_5, p_8), (t_6, p_9)\}$

$$F = \begin{pmatrix} & p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 & p_8 & p_9 \\ t_1 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ t_2 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ t_3 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ t_4 & 0 & 0 & 0 & 1 & 0 & 1 & -1 & 1 & 0 \\ t_5 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ t_6 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & -1 \end{pmatrix}.$$

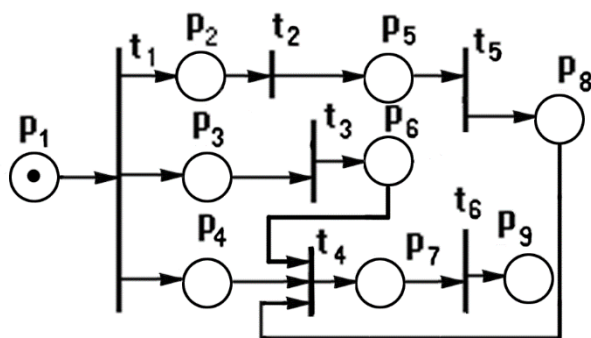


Рис. 1.11. Приклад мережі Петрі

Динаміка мережі Петрі

Динаміка мереж Петрі базується на двох основних правилах.

Правило активації

Перехід $t \in T$ активується відповідним маркуванням μ його вхідних позицій тоді й тільки тоді, коли вони містять задану кількість міток.

Активация переходів створює передумови їх спрацювання, але не кожний активований перехід може спрацювати.

Правило спрацювання

Спрацювання переходу t , активованого маркуванням $\mu^{(\tau)}$, породжує нове маркування мережі $\mu^{(\tau+1)}$ шляхом вилучення міток із вхідних позицій і розміщення деякої кількості міток на вихідних позиціях.

Контрольні запитання

1. Опишіть загальні властивості послідовних алгоритмів.
2. Нехай необхідно побудувати такі алгоритми:
 - Обчислення суми перших n елементів геометричної прогресії.
 - Сортування масиву вставками.
 - Пошук максимального числа в тривимірному масиві.
 - Знаходження коренів нелінійного рівняння 5-го степеня.

Визначте, які з цих алгоритмів є чисельними, а які – логічними.

3. Чи можна побудувати паралельний алгоритм додавання N чисел?

4. Нехай потрібно обчислити вираз:

$$f(x, y) = \begin{cases} y/x, & x > y, \\ y/x, & x \leq y \end{cases} - 10 \leq x \leq 10, -5 \leq y \leq 5, \text{ де } x, y - \text{цілі числа.}$$

Запишіть псевдокод алгоритму обчислення виразу та задайте алгоритм графічно.

5. Алгоритм Евкліда є чисельним чи логічним алгоритмом? Запишіть послідовність дій алгоритму Евкліда при $a = 8$ і $b = 12$.

1.2. Міри складності алгоритмів

Для розв'язування однієї й тієї ж задачі можна застосувати різні алгоритми. Ефективність роботи кожного з них визначається різними факторами і залежить від:

- кваліфікації програміста,
- потужності обчислювальних засобів,
- застосовуваних методів обчислень.

При аналізі алгоритму визначається кількість «часу», необхідного для його виконання.

Критерій часової оцінки – це не реальне число секунд або інших проміжків часу, а приблизне число операцій, виконуваних алгоритмом. Число операцій визначає відносний час виконання алгоритму. Таким чином, іноді ми будемо називати «часом» середню кількість операцій, необхідних для виконання алгоритму. Фактична кількість секунд, необхідних для виконання алгоритму на комп'ютері, непридатна для аналізу, оскільки нас цікавить тільки відносна ефективність алгоритму, що вирішує конкретну задачу. Дійсно, алгоритм не покращується, якщо його перенести на більш швидкий комп'ютер, і не погіршується, якщо його виконувати на більш повільному комп'ютері.

Крім того, фактична кількість операцій алгоритму на тих або інших даних, що вводяться, не надає вичерпної інформації про якість алгоритму або мало його характеризує. Замість цього важливою характеристикою є залежність числа операцій конкретного алгоритму від обсягу вхідних даних. Ми можемо порівняти два алгоритми за швидкістю зростання числа операцій у залежності від зростання обсягу вхідних даних. Саме швидкість зростання відіграє ключову роль.

При аналізі алгоритмів враховують складність алгоритмів за часом, однак потрібно враховувати й те, який обсяг пам'яті потрібний для роботи того або іншого алгоритму. На ранніх етапах розвитку комп'ютерів цей аналіз носив принциповий характер. Досить часто доводилося вибирати більш повільний алгоритм, якщо він вимагав менше пам'яті. Розроблювачі сучасних програм не відчують потреби в економії пам'яті, у результаті

чого комп'ютер морально застаріває задовго до його фізичної непридатності.

Швидкість росту алгоритму визначається швидкістю зростання числа операцій при зростанні обсягу вхідних даних. Нас цікавить тільки загальний характер поведінки алгоритму, а не подробиці цієї поведінки. Отже, при аналізі алгоритмів нас буде цікавити швидше за все клас функції швидкості росту, до якого належить алгоритм, ніж точна кількість виконуваних ним операцій [7].

Деякі поширені класи функцій наведені в таблиці. У цій таблиці наведені значення функцій з даного класу на широкому діапазоні значень аргументу. Видно, що при невеликих обсягах вхідних даних значення функцій мало відрізняються, однак при зростанні цих обсягів різниця суттєво зростає. Крім того, швидкодіючі функції домінують над функціями з більш повільним ростом. Тому якщо ми виявимо, що складність алгоритму є сумою двох або декількох таких функцій, то будемо часто відкидати всі функції, крім тих, які зростають найшвидше. Якщо, наприклад, буде встановлено, що алгоритму потрібно $x^3 - 30x$ операцій, то будемо вважати, що складність алгоритму росте як x^3 , оскільки вже при 100 вхідних даних різниця між x^3 і $x^3 - 30x$ становить лише 0,3%.

Таблиця 1.2

Таблиця класів росту функцій

n	$\log_2 n$	n^2	n^3	2^n	n!
1	0	1	1	2	1
2	1	4	8	4	2
5	2.3	25	125	32	120
10	3.3	100	1000	1024	362880
15	3.9	225	3375	32768	-----
20	4.3	400	8000	1048576	-----
30	4.9	900	27000	1073741824	-----

Швидкість росту складності алгоритму відіграє важливу роль, швидкість росту визначається старшим членом формули. Відкинувши всі молодші члени, ми одержуємо те, що називають порядком функції або алгоритму, швидкістю росту складності якого вона є. Алгоритми можна згрупувати за швидкостями росту їх складностей. Введемо насамперед деякі поняття, пов'язані з асимптотичною оцінкою функцій.

1.2.1. Асимптотична обчислювальна складність

Хоча в багатьох випадках ці оцінки використовують неформально, корисно почати з точних визначень.

Зазначимо, що у даному розділі функції відображають цілі числа в цілі, у тому випадку, якщо з контексту це незрозуміло [8].

Оцінка Θ . Якщо $f(n)$ й $g(n)$ – деякі функції, то запис $f(n) = \Theta(g(n))$ означає, що знайдуться такі $c_1, c_2 > 0$ й таке n_0 , що $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ для всіх $n \geq n_0$.

У цьому випадку говорять, що $g(n)$ є *асимптотично точною оцінкою* для $f(n)$.

Однак це позначення слід використовувати з обережністю. Встановивши, що $f_1(n) = \Theta(g(n))$ й $f_2(n) = \Theta(g(n))$, не слід робити висновок, що $f_1(n) = f_2(n)$!

Визначення $\Theta(g(n))$ припускає, що функції $f(n)$ й $g(n)$ асимптотично невід'ємні, тобто невід'ємні для достатньо великих значень n . Зазначимо, що якщо f й g строго додатні, то можна виключити n_0 з визначення (змінивши константи c_1 й c_2 так, щоб для малих n нерівність також виконувалася).

Це відношення симетричне: якщо $f(n) = \Theta(g(n))$, то $g(n) = \Theta(f(n))$.

Приклад 1.7. Доведення асимптотичної складності

Перевіримо, що $\frac{n^2}{2} - 3n = \Theta(n^2)$. Згідно з визначенням, потрібно

вказати додатні константи c_1 , c_2 і число n_0 так, щоб нерівності

$$c_1 n^2 \leq \frac{n^2}{2} - 3n \leq c_2 n^2$$

виконувалися для всіх $n \geq n_0$. Розділимо згадані нерівності на n^2 :

$$c_1 \leq \frac{1}{2} - \frac{3}{n} \leq c_2$$

Видно, що для виконання другої нерівності $\frac{1}{2} - \frac{3}{n} \leq c_2$ достатньо прийняти, що $c_2 = \frac{1}{2}$. Перша нерівність $c_1 \leq \frac{1}{2} - \frac{3}{n}$ буде виконана, якщо, наприклад, $n_0 = 7$ і $c_1 = \frac{1}{14}$.

Приклад 1.8. Доведення нерівності асимптотичної складності Використання формального визначення: покажемо, що $6n^3 \neq \Theta(n^2)$. Насправді, припустимо, що найдуться такі c_2 й n_0 , що $6n^3 \leq c_2 n^2$ для всіх $n \geq n_0$. Розділивши нерівність $6n^3 \leq c_2 n^2$ на n^2 , одержуємо $6n \leq c_2$, звідси $n \leq \frac{c_2}{6}$. Яке б велике додатне c_2 ми б не взяли, завжди знайдеться таке додатне n , значення якого перевищить $\frac{c_2}{6}$.

Відшуковуючи асимптотично точну оцінку для суми, ми можемо відкидати члени меншого порядку, які при більших n стають малими в порівнянні з основним доданком. Відзначимо також, що коефіцієнт при старшому члені не впливає (він може вплинути тільки на вибір констант c_1 і c_2). Наприклад, розглянемо квадратичну функцію $f(n) = an^2 + bn + c$, де a , b і c – деякі константи, й $a > 0$. Відкидаючи члени молодших порядків і коефіцієнт при старшому члені, знаходимо, що $f(n) = \Theta(n^2)$. Щоб переконатися в цьому формально, можна

прийняти, що $c_1 = \frac{a}{4}$, $c_2 = \frac{7a}{4}$ і $n_0 = 2 \max\left(\frac{|b|}{a}, \sqrt{\frac{|c|}{a}}\right)$.

Взагалі, для будь-якого полінома $p(n)$ степеня d з додатним старшим коефіцієнтом маємо $p(n) = \Theta(n^d)$.

Згадаємо важливий окремий випадок використання Θ -позначень:

$\Theta(1)$ позначає обмежену функцію, відділену від нуля якоюсь додатною константою при достатньо великих значеннях аргументу.

1.2.2. Асимптотичні оцінки Θ , O та Ω

Запис $f(n) = \Theta(g(n))$ містить дві оцінки: верхню й нижню. Їх можна розділити.

Якщо $f(n)$ й $g(n)$ – деякі функції, то

- запис $f(n) = O(g(n))$ означає, що знайдеться така константа $c > 0$ й таке n_0 , що $f(n) \leq cg(n)$ для всіх $n \geq n_0$;

- запис $f(n) = \Omega(g(n))$ означає, що знайдеться така константа $c > 0$ й таке n_0 , що $0 \leq cg(n) \leq f(n)$ для всіх $n \geq n_0$.

Теорема 1.1.

1. Для будь-яких двох функцій $f(n)$ і $g(n)$ властивість $f(n) = \Theta(g(n))$ справджується тоді і тільки тоді, коли $f(n) = O(g(n))$ й $f(n) = \Omega(g(n))$
2. Для будь-яких двох функцій $f(n)$ і $g(n)$ властивість $f(n) = O(g(n))$ справджується тоді й тільки тоді, коли $g(n) = \Omega(f(n))$.

Оскільки $an^2 + bn + c = \Theta(n^2)$ (при $a > 0$). Тому $an^2 + bn + c = O(n^2)$.

Працюючи із символами O , Θ і Ω ми маємо справу з *однобічними* рівностями – ці символи можуть стояти тільки праворуч від знака $=$.

1.2.3. Властивості асимптотичних функцій

Введені нами визначення мають деякі властивості транзитивності, рефлексивності й симетричності.

Транзитивність

$f(n) = \Theta(g(n))$ і $g(n) = \Theta(h(n))$ спричиняє $f(n) = \Theta(h(n))$

$f(n) = O(g(n))$ і $g(n) = O(h(n))$ спричиняє $f(n) = O(h(n))$

$f(n) = \Omega(g(n))$ і $g(n) = \Omega(h(n))$ спричиняє $f(n) = \Omega(h(n))$

Рефлексивність

$f(n) = \Theta(f(n))$, $f(n) = O(f(n))$, $f(n) = \Omega(f(n))$

Симетричність

$f(n) = \Theta(g(n)) \Leftrightarrow g(n) = \Theta(f(n))$

Контрольні запитання

1. Які класи росту функцій вам відомі?
2. Доведіть справедливість нерівності для верхньої оцінки складності: $2n^2 \neq O(n)$.
3. Доведіть справедливість рівності для оцінки складності: $2n^2 + n = O(n^2)$.
4. Наведіть приклад алгоритму з алгоритмічною складністю $O(n^2)$.
5. Яку функцію задає нерівність $0 \leq c_1g(n) \leq f(n) \leq c_2g(n)$?

1.3. Класи задач P і NP

1.3.1. Алгоритми поліноміальної складності (клас P)

Більшість алгоритмів мають поліноміальний порядок складності.

Складність $O(n)$. Іноді час роботи виявляється лінійним $O(n)$, як при послідовному пошуку: при подовженні списку даних удвічі алгоритм працює вдвічі довше. В алгоритмах послідовного пошуку нас цікавить процес перегляду списку в пошуках деякого елемента, називаного цільовим. При послідовному пошуку передбачається, що список не відсортований. Наприклад, ключове значення може бути номером співробітника, прізвищем або будь-яким іншим унікальним ідентифікатором. Алгоритм послідовного пошуку послідовно переглядає по одному елементу списку, починаючи з першого, доти, поки не знайде потрібний елемент. Очевидно,

що чим далі в списку перебуває конкретне значення ключа, тим більше часу піде на його пошук.

Складність $O(n^2)$. Дуже часто трапляються алгоритми складності $O(n^2)$ – таку складність мають деякі алгоритми сортування: якщо довжину вхідного списку подвоїти, то час роботи алгоритму зросте у 4 рази. Усі вісім існуючих алгоритмів сортування демонструють широкий спектр можливих варіантів поведінки. *Сортування вставками* сортує список, вставляючи черговий елемент у потрібне місце вже відсортованого списку. *Бульбашкове сортування* порівнює елементи попарно, переставляючи між собою елементи тих пар, порядок у яких порушений. *Сортування Шелла* є багатопрохідним сортуванням, при якому список розбивається на підсписки, кожний з яких сортується окремо, причому на кожному проході кількість підсписків зменшується, а їх довжина зростає.

Розглянемо найбільш типовий варіант сортування – бульбашкове сортування. Алгоритм бульбашкового сортування здійснює кілька проходів за списком. При кожному проході відбувається порівняння сусідніх елементів. Якщо порядок сусідніх елементів неправильний, вони міняються місцями. Кожний прохід починається з початку списку. Спочатку порівнюються 1 і 2 елементи, потім 2 і 3, потім 3 і 4 і т. д. Елементи з неправильним порядком у парі переставляються. При виявленні на першому проході найбільшого елемента списку він буде переставлятися з усіма наступними, поки не дійде до кінця списку. Тому при другому проході немає необхідності порівняння з останнім елементом. При другому проході другий за величиною елемент списку опуститься на другу позицію з кінця і т. д. Зазначимо, що при кожному проході ближче до свого місця просувається відразу кілька елементів, хоча гарантовано займає остаточне положення лише один.

Скільки порівнянь виконується в найгіршому випадку? На першому проході буде виконано $n - 1$ порівнянь сусідніх значень, на другому $n - 2$ порівняння. Подальше дослідження показує, що при кожному черговому проході число порівнянь зменшується на 1. Тому складність у найгіршому випадку задають формулою

$$W(n) = \sum_{i=n-1}^1 i = \sum_{i=1}^{n-1} i = \frac{(n-1)n}{2} = \frac{n^2 - n}{2} \approx \frac{n^2}{2} = O(n^2)$$

Складність стандартного алгоритму матричного множення дорівнює $O(n^3)$ й при збільшенні розмірів матриць удвічі такий алгоритм працює у 8 разів довше.

Матриця – математичний об’єкт, еквівалентний двовимірному масиву. Якщо число стовпців у першій матриці збігається із числом рядків у другій, то ці дві матриці можна перемножити. Для обчислення добутку двох матриць кожний рядок першої почленно множиться на кожний стовець другої. Потім обчислюється сума таких добутків і записується у відповідну клітинку результату.

Стандартний алгоритм множення матриці розміром $a \times b$ на матрицю розміром $b \times c$ виконує abc множень і $a(b-1)c$ додавань.

Однак дослідникам удалося виявити інші алгоритми, що множать матриці більш ефективно, зокрема, алгоритм Виноградова й алгоритм Штрассена.

Алгоритм Штрассена працює з квадратними матрицями. Насправді він настільки ефективний, що іноді доцільно розширити матриці до квадратних, і при цьому він все одно дає кращий результат. Аналіз загального випадку показує, що число множень при множенні двох $n \times n$ матриць приблизно дорівнює $n^{2.81}$, а число додавань $6n^{2.81} - 6n^2$.

Зводячи три результати до купи, маємо наступну таблицю:

Таблиця 1.3

Кількість операцій у алгоритмах множення матриць

	Множення	Додавання
Стандартний алгоритм	n^3	$n^3 - n^2$
Алгоритм Виноградова	$\frac{n^3 + 2n^2}{2}$	$\frac{3n^3 + 4n^2 - 4n}{2}$
Алгоритм Штрассена	$n^{2.81}$	$6n^{2.81} - 6n^2$

Усі розглянуті алгоритми мають поліноміальну складність. Найбільш часомістким є алгоритм множення матриць, його складність $O(n^3)$. Головне, однак, те, що ми могли знайти такий розв'язок задач за прийнятний проміжок часу. Усі ці задачі належать до класу P – класу задач поліноміальної складності. Такі задачі називають також практично розв'язними.

1.3.2. Алгоритми недетермінованої поліноміальної складності (клас NP задач)

Крім практично розв'язних задач, що належать до класу P – класу задач поліноміальної складності, існує й інший клас задач: вони практично нерозв'язні й ми не знаємо алгоритмів, здатних розв'язати їх за прийнятний час. Ці задачі утворюють клас NP – недетермінованої поліноміальної складності.

Відзначимо тільки, що складність усіх відомих детермінованих алгоритмів, що вирішують ці задачі, або експоненціальна, або факторіальна. Складність деяких з них дорівнює 2^n , де n – кількість вхідних даних. У цьому випадку при додаванні до списку вхідних даних одного елемента час роботи алгоритму подвоюється. Якщо для розв'язування такої задачі на вході з 10 елементів алгоритму було потрібно 1024 операцій, то на вході

з 11 елементів число операцій становитиме 2048. Це значне зростання часу при невеликому збільшенні кількості вхідних даних.

Термін «недетерміновані поліноміальні», що характеризує задачі із класу NP , пояснюється наступним двокроковим підходом до їх розв'язування. На першому кроці є недетермінований алгоритм, що генерує можливий розв'язок такої задачі – щось подібне до спроби вказати розв'язок; іноді така спроба виявляється успішною, і ми одержуємо оптимальну або близьку до оптимальної відповідь, але частіше відповідь далека від оптимальної. На другому кроці перевіряється, чи дійсно відповідь, отримана на першому кроці, є розв'язком початкової задачі. Кожний із цих кроків окремо вимагає поліноміального часу. Проблема, однак, у тому, що ми не знаємо, скільки разів нам доведеться повторити обидва кроки, щоб одержати шуканий розв'язок. Хоча обидва кроки є поліноміальними, кількість звертань до них може бути експоненціальною або факторіальною.

1.3.3. Задача про комівояжера

До класу NP належить *задача про комівояжера*. Нам заданий набір міст і «вартість» подорожі між будь-якими двома з них. Потрібно визначити такий порядок, при якому слід відвідати всі міста (по одному разу) і повернутися у початкове місто, щоб загальна вартість подорожі виявилася мінімальною. Цю задачу можна застосувати, наприклад, для визначення порядку ефективного збору сміття з баків на вулицях міста або вибору найкоротшого шляху поширення інформації із усіх вузлів комп'ютерної мережі. Вісім міст можна впорядкувати 40 320 можливими способами, а для десяти міст це число зростає вже до 3 628 800. Пошук найкоротшого шляху вимагає перебору всіх цих можливостей. Припустимо, що в нас є алгоритм, здатний підрахувати вартість подорожі через 15 міст у зазначеному порядку. Якщо за секунду такий алгоритм здатний пропустити через себе

100 варіантів, то йому буде потрібно більше чотирьох століть, щоб дослідити всі можливості й знайти найкоротший шлях. Навіть якщо в нас є 400 комп'ютерів, для обчислень знадобиться рік, хоча ми маємо лише 15 міст. Для 20 міст мільярд комп'ютерів працюватимуть паралельно протягом дев'яти місяців, щоб знайти найкоротший шлях. Ясно, що швидше й дешевше подорожувати хоч як-небудь, ніж чекати, поки комп'ютери дадуть оптимальний розв'язок.

Чи можна знайти найкоротший шлях, не переглядаючи їх всі? Дотепер нікому не вдалося створити алгоритм, який не займається, по суті, переглядом усіх шляхів. Коли число міст невелике, задача вирішується швидко, однак це не означає, що так буде завжди, а нас цікавить саме розв'язок загальної задачі.

Задача про комівояжера, звичайно, дуже схожа на задачі про графи. Кожне місто можна представити вершиною графа, наявність шляху між двома містами – ребром, вартість подорожі між ними – вагою цього ребра. Звідси можна зробити висновок, що алгоритм пошуку найкоротшого шляху вирішує й задачу комівояжера, однак це не так. Які ж дві умови задачі про комівояжера відрізняють її від задачі про найкоротший шлях? По-перше, ми повинні відвідати всі міста, а алгоритм пошуку найкоротшого шляху дає лише шлях між двома заданими містами. Якщо вибрати шлях з найкоротших відрізків, видаваних алгоритмом пошуку найкоротших шляхів, то він буде проходити через деякі міста по декілька разів. Друга відмінність полягає у вимозі повернення у початкову точку, така вимога відсутня у пошуку найкоротшого шляху.

З огляду на те, наскільки велике число можливих упорядкувань вершин, детермінований алгоритм, що порівнює всі можливі способи упорядкування, працює надто довго. Щоб показати, що ця задача належить до класу NP , необхідно зрозуміти, як її можна розв'язати за допомогою описаної вище двокрокової процедури. У задачі про комівояжера на

першому кроці випадковим чином генерується деяке упорядкування міст. Оскільки це недетермінований процес, щоразу виходитиме новий порядок. Очевидно, що процес генерації можна реалізувати за поліноміальний час: ми можемо зберігати список міст, генерувати випадковий номер, вибирати зі списку місто з цією назвою і видаляти його зі списку, щоб він не з'явився другий раз. Така процедура виконується за $O(n)$ операцій, де n – число міст. На другому кроці відбувається підрахунок вартості подорожі по містах у зазначеному порядку. Для цього потрібно просто додати вартості подорожі між послідовними парами міст у списку, що також вимагає $O(n)$ операцій. Обидва кроки поліноміальні, тому задача про комівояжера належить до класу NP . Часомісткою робить її саме необхідне число ітерацій цієї процедури.

Слід зазначити, що таку двокрокову процедуру можна було застосувати до кожної із задач, що розглядалися нами раніше. Наприклад, сортування списку можна виконувати, генеруючи довільний порядок елементів початкового списку й перевіряючи, чи не є цей порядок зростаючим.

Чи не відносить це міркування задачу сортування до класу NP ? Звичайно, відносить. Різниця між класом P и класом NP у тому, що в першому випадку наявним є детермінований алгоритм, що вирішує задачу за поліноміальний час, а в другому ми такого алгоритму не знаємо.

1.3.4. Зведення задачі до іншої задачі

Один зі способів розв'язування задач полягає в тому, щоб звести, або редукувати, одну задачу до іншої. Тоді алгоритм розв'язування другої задачі можна перетворити таким чином, щоб він вирішував першу. Якщо перетворення виконується за поліноміальний час і друга задача вирішується

за поліноміальний час, то й наша нова задача також вирішується за поліноміальний час.

Пояснимо наше міркування прикладом.

Нехай перше завдання полягає в тому, щоб повернути значення «так» у випадку, якщо одна з даних булевих змінних має значення «істина», і повернути «ні» у протилежному випадку.

Друга задача полягає в тому, щоб знайти максимальне значення в списку цілих чисел. Кожна з них допускає простий зрозумілий розв'язок, але припустимо на хвилину, що ми знаємо розв'язок задачі про пошук максимуму, а задачу про булеві змінні вирішувати не вміємо.

Ми прагнемо звести задачу про булеві змінні до задачі про максимум цілих чисел. Напишемо алгоритм перетворення набору значень булевих змінних у список цілих чисел, який значенню «false» ставить у відповідність число 0, а значенню «true» – число 1.

Потім скористаємося алгоритмом пошуку максимального елемента в списку. По тому, як складався список, робимо висновок, що цей максимальний елемент може бути або нулем, або одиницею. Таку відповідь можна перетворити у відповідь у задачі про булеві змінні, повертаючи «так», якщо максимальне значення дорівнює 1, і «ні», якщо воно дорівнює 0.

Пошук максимального значення виконується за лінійний час, а редукція першої задачі до другої теж вимагає лінійного часу, тому задачу про булеві змінні теж можна розв'язати за лінійний час.

1.3.5. Типові *NP* задачі

Кожна із задач у 1.3.5 є або оптимізаційною, або задачею про прийняття рішення. Метою оптимізаційної задачі зазвичай є конкретний результат, що представляє собою мінімальне або максимальне значення. У задачі про прийняття рішення зазвичай задають деяке граничне значення, і нас цікавить, чи існує розв'язок, більший (у задачах максимізації) або

менший (у задачах мінімізації) зазначеної границі. Відповіддю в задачах оптимізації є отриманий конкретний результат, а в задачах про прийняття рішень – «так» або «ні».

У 1.3.3 розглянутий оптимізаційний варіант задачі про комівояжера. Це задача мінімізації, і нас цікавив шлях мінімальної вартості. У варіанті ухвалення рішення ми могли б запитати, чи існує шлях комівояжера з вартістю, яка є меншою за задану константу C . Ясно, що відповідь у задачі про ухвалення рішення залежить від обраної границі. Якщо ця границя дуже велика (наприклад, вона перевищує сумарну вартість усіх доріг), то відповідь «так» одержати нескладно. Якщо ця границя надто мала (наприклад, вона менша за вартість шляху між будь-якими двома містами), то відповідь «ні» також дається легко. У інших проміжних випадках час пошуку відповіді дуже великий й порівнянний з часом розв'язування оптимізаційної задачі. Тому ми розглядатимемо як задачі оптимізації, так і задачі прийняття рішень, використовуючи ту з них, яка точніше відповідає нашим поточним цілям.

Наведемо опис шести NP задач – як в оптимізаційному варіанті, так і у варіанті прийняття рішення.

Розфарбування графа

Граф $G = (V, E)$ є набором вершин, або вузлів, V і набором ребер E , що з'єднують вершини попарно. Будемо розглядати тільки *неорієнтовані* графи. Вершини графа можна розфарбувати в різні кольори, які зазвичай позначають цілими числами. Нас цікавлять такі розфарбування, у яких кінці кожного ребра пофарбовані різними кольорами. Очевидно, що в графі з N вершинами можна пофарбувати вершини в N різних кольорів, але чи можна обійтися меншою кількістю кольорів?

У задачі оптимізації нас цікавить мінімальне число кольорів, необхідних для розфарбування вершин графа [9].

У задачі ухвалення рішення нас цікавить, чи можна розфарбувати вершини в 3 або менше кольорів.

У задачі про розфарбування графа є практичні додатки. Якщо кожна вершина графа позначає курс, який читають у коледжі, і вершини з'єднуються ребром, якщо ці курси читають для одного і того ж студента, то отриманий граф досить складний. Якщо припустити, що кожному студенту читають 5 курсів, то на нього припадає 10 ребер. Припустимо, що на 3500 студентів припадає 500 курсів. Тоді в графі буде 500 вершин і 35000 ребер. Якщо на іспити відведено 20 днів, то це означає, що вершини графа потрібно розфарбувати в 20 кольорів, щоб у жодного студента не було по два іспити на день.

Розробка безконфліктного розкладу іспитів еквівалентна розфарбуванню графів. Однак задача розфарбування графів належить до класу NP , тому розробка безконфліктного розкладу за прийнятний час неможлива. Крім того, при плануванні іспитів зазвичай потрібно, щоб у студента було не більше двох іспитів на день, а екзамени з різних частин курсу призначаються в один день. Очевидно, що розробка «досконалого» плану іспитів неможлива, і тому необхідна інша техніка для одержання, принаймні, непоганих розкладів.

Розкладка по ящиках

Нехай у нас є кілька ящиків одиничної ємності й набір об'єктів різних розмірів s_1, \dots, s_N . У задачі оптимізації нас цікавить найменша кількість ящиків, необхідна для розкладки всіх об'єктів, а в задачі ухвалення рішення – чи можна упакувати всі об'єкти у B або меншу кількість ящиків.

Ця задача виникає при запису інформації на диск або у фрагментовану пам'ять комп'ютера, при ефективному розподілі вантажу на кораблях, при вирізанні частин зі стандартних порцій матеріалу на замовлення клієнтів. Якщо, наприклад, у нас є великі металеві пластини й список замовлень на

менші пластини, то, природно, ми прагнемо розподілити замовлення якомога щільніше, зменшивши втрати й збільшивши прибуток.

Упакування рюкзака

У нас є набір об'єктів об'ємом s_1, \dots, s_N вартості w_1, \dots, w_N . У задачі оптимізації ми прагнемо впакувати рюкзак об'ємом K так, щоб його вартість була максимальною. У задачі ухвалення рішення нас цікавить, чи можна досягти того, щоб сумарна вартість упакованих об'єктів була, щонайменше, W .

Ця задача виникає при виборі стратегії вкладення грошей: об'ємом тут є обсяг різних вкладень, вартістю – передбачувана величина доходу, а об'єм рюкзака визначається розміром (обсягом) запланованих капіталовкладень.

Задача про суми елементів підмножин

Нехай у нас є множина об'єктів різних розмірів s_1, \dots, s_N і деяка додатна верхня границя L . У задачі оптимізації нам необхідно знайти набір об'єктів, сума розмірів яких найбільш близька до L і не перевищує цієї верхньої границі. У задачі ухвалення рішення потрібно встановити, чи існує набір об'єктів із сумою розмірів L . Це спрощена версія задачі про упаковання рюкзака.

Задача про істинність КНФ-виразу

Кон'юнктивна нормальна форма (КНФ) є послідовністю булевих виразів, пов'язаних між собою операторами AND (позначуваними \wedge), причому кожен вираз є мономом від булевих змінних або їх заперечень, пов'язаних операторами OR (які позначають через \vee).

Приклад 1.9. Булевий вираз у кон'юнктивній нормальній формі (заперечення позначають рискою над іменем змінної):

$$(a \vee b) \wedge (\bar{a} \vee c) \wedge (a \vee b \vee \bar{c} \vee d) \wedge (b \vee \bar{c} \vee \bar{d}) \wedge (a \vee \bar{b} \vee \bar{c} \vee \bar{d} \vee e).$$

Задача про істинність булевого виразу в кон'юнктивній нормальній формі ставиться тільки у варіанті прийняття рішення: чи існують у змінних, що входять у вираз, такі значення істинності, підстановка яких робить увесь

вираз дійсним. Як число змінних, так і складність виразу не обмежені, тому число комбінацій значень істинності може бути дуже великим.

Задача планування робіт

Нехай у нас є набір робіт і ми знаємо час, необхідний для завершення кожної з них, t_1, t_2, \dots, t_N , терміни d_1, d_2, \dots, d_N , до яких ці роботи повинні бути обов'язково завершені, а також штрафи p_1, p_2, \dots, p_N , які будуть накладені при незавершенні кожної роботи у встановлений термін. Задача оптимізації вимагає встановити порядок робіт, який мінімізує штрафи, що накладаються. У задачі прийняття рішень ми запитуємо, чи існує порядок робіт, при якому величина штрафу буде не більше, ніж P .

Контрольні запитання

1. Яку обчислювальну складність має алгоритм сортування Шелла?
2. Яка обчислювальна складність алгоритму множення матриць?
3. Чому алгоритм розв'язування задачі про комівояжера має недетерміновану поліноміальну обчислювальну складність?
4. Який типовий підхід до розв'язування задач NP -класу вам відомий?
5. Назвіть відомі вам задачі, розв'язування яких приводить до алгоритмів з недетермінованою поліноміальною обчислювальною складністю.

1.4. Алгоритми обчислення похибок

При розв'язуванні задач на ЕОМ практично неможливо одержати точний розв'язок. Одержуваний чисельний розв'язок майже завжди містить похибку, тобто є наближенням [10].

У цьому розділі розглянемо: джерела виникнення похибок при розв'язуванні задачі, основні правила задавання наближених величин, оцінки похибок обчислень деяких функцій.

Приклад 1.10. *Наближене визначення величин:*

$$\frac{1}{3} \approx 0.3333, \quad \pi \approx 3.14, \quad \sqrt{2} = 1.4142$$

1.4.1. Джерела похибок результату

Похибки розв'язку задач на ЕОМ пояснюються наступними причинами.

Похибка математичної моделі пов'язана з невідповідністю моделі фізичній реальності. При моделюванні може не бути врахованим: тертя, опір середовища, температура, сили поверхневого натягу.

Якщо математична модель обрана недостатньо ретельно, то, які б методи ми не застосовували для обчислень, усі результати будуть ненадійні або неправильні. Прикладом помилкової моделі є модель вічного двигуна.

Модель вічного двигуна

Вічний двигун першого роду – це ідеальний двигун, який після запуску має працювати постійно і не вимагати додаткового надходження енергії [11]. На рис. 1.12 показана одна з перших конструкцій вічного двигуна. За всю історію людство накопичило певну кількість різних конструкцій двигуна першого роду, проте жодна не виконує покладені на неї завдання.

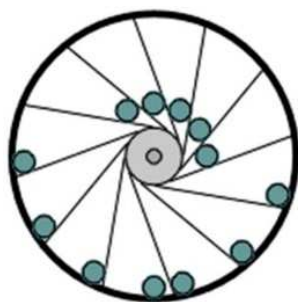


Рис. 1.12. Вічний двигун

Причиною цього є той факт, що всі такі моделі вступають у протиріччя з законом збереження енергії.

Похибка вхідних (початкових) даних

Значення величин, що входять в умову задачі, одержують у результаті вимірювань, а тому вони мають наближений характер.

На рис. 1.13 показано приклад практичного вимірювання лінійкою розміру поштової марки, який не може бути отриманий з високою точністю.



Рис. 1.13. Похибка вимірювання

Це *непереборна похибка*, але цю похибку можливо й необхідно оцінити для вибору алгоритму розрахунків і точності обчислень.

Похибки експерименту поділяють на систематичні, випадкові, грубі. Ідентифікація таких помилок можлива при статистичному аналізі результатів експерименту.

Похибка методу

Похибка методу – це наслідок прийнятих методик розрахунку, які використовують наближені формули.

Чисельний алгоритм має дискретний характер. Це означає, що замість точного розв’язку задачі метод знаходить розв’язок іншої задачі, близької до шуканої.

Похибка методу – основна характеристика будь-якого чисельного алгоритму. Похибка методу повинна бути у 2–5 разів меншою за непереборну похибку.

Обчислення похибки многочлена Лагранжа

$$R_n(x) = \frac{w_{n+1}(x)}{(n+1)!} f^{(n+1)}(\xi)$$

$$w_{n+1}(x) = (x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_i)(x - x_{i+1}) \dots (x - x_n)$$

Похибка округлення

Похибка округлення пов'язана з використанням в обчислювальних машинах чисел з скінченною точністю представлення. При реалізації чисельних алгоритмів комп'ютери оперують скінченними десятковими дробами, тобто наближеними числами.

Отже, дані і результати проміжних операцій округлюються, що призводить до накопичення похибки.

1.4.2. Задачі, що виникають при роботі з наближеними величинами

1. Математично виражають похибки наближених величин.
2. Оцінюють похибку результату, коли відома похибка вхідних даних.
3. Знаходять похибку вхідних даних з метою одержання заданої похибки результату.
4. Погоджують похибки різних вхідних даних для того, щоб не виконувати зайвої роботи при обчисленні, якщо інші дані занадто грубі.
5. Стежать в процесі обчислень за похибкою проміжних результатів для того, щоб, з одного боку, забезпечити необхідну точність остаточного результату, а з іншого боку, за можливості спростити обчислення.

1.4.3. Основні характеристики наближених чисел

Основними характеристиками наближених чисел є абсолютна та відносна похибки. Додатковими характеристиками наближених чисел є межа абсолютної та межа відносної похибки.

Означення абсолютної похибки. Абсолютною похибкою $\Delta_a x$ наближеного значення величини називають модуль різниці між її точним значенням a та наближеним значенням x .

$$\Delta_a x = |a - x|$$

Приклад 1.11. *Визначення абсолютної похибки*

$$\pi \approx 3.14 \rightarrow \Delta_{\pi} = |\pi - 3.14| = |3.1415926... - 3.14| = 0.0015926...$$

$$\pi \approx 3.1415 \rightarrow \Delta_{\pi} = |\pi - 3.1415| = |3.1415926... - 3.1415| = 0.0000926...$$

Межа абсолютної похибки

На практиці абсолютна похибка наближення часто не може бути обчислена, оскільки невідоме точне значення величини. Проте майже завжди можна вказати число, яке не менше від абсолютної похибки, тобто оцінити цю похибку зверху. У нашому прикладі можлива, зокрема, така оцінка похибки зверху: $|\pi - 3.14| < 0.2$.

Нехай для механізму необхідна кругла деталь діаметром 24 мм. Відомо, що він працює задовільно, якщо діаметр d деталі відрізняється від заданого не більше ніж на $h_a = \pm 0.5$ мм. Отже, $d = 24 \pm h_a = 24 \pm 0.5$ мм, або $d \in [23.5; 24.5]$.

Означення межі абсолютної похибки. Число h_a , яке не менше від абсолютної похибки $\Delta_a x$, називають межею абсолютної похибки.

Абсолютна похибка має таку саму розмірність, що і розглядувана величина.

Означення відносної похибки. Відносна похибка σ_a дорівнює відношенню абсолютної похибки обчислення $\Delta_a x$ до модуля наближеного значення величини x :

$$\sigma_a x = \frac{\Delta_a x}{|x|}.$$

Відносну похибку часто виражають у відсотках.

Відносна похибка є безрозмірною величиною.

Означення приведеної (нормалізованої) відносної похибки. Приведена похибка $\delta_a x$ дорівнює відношенню абсолютної похибки $\Delta_a x$ до

$$\text{максимального значення } x_{\max}: \delta_a x = \frac{\Delta_a x}{|x_{\max}|}.$$

Далі будемо її називати: h_x – межа відносної похибки.

Приклад 1.12. Обчислення відносної похибки

Розглянемо прямокутник зі сторонами 12 см і 5 см (рис. 1.14).

Після вимірювання його діагоналі лінійкою отримали результат 12.9 см. Яка відносна похибка цього наближення?

$$a = \sqrt{12^2 + 5^2} = \sqrt{144 + 25} = \sqrt{169} = 13 \text{ см}, \quad x = 12.9 \text{ см}$$

$$\Delta_a x = |13 - 12.9| = 0.1 \text{ см}$$

$$\sigma_a x = \frac{0.1}{12.9} = 0.00775$$

$$\text{або } \sigma_a x = \frac{1}{129} \cdot 100\% \approx 0.78\%$$

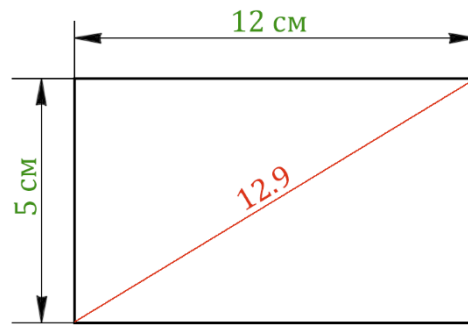


Рис. 1.14. Прямокутник 12 × 5 см

Означення межі відносної похибки. Число E_a , яке не менше за відносну похибку, називають межею відносної похибки: $E_a > \sigma_a x$.

Оскільки $\sigma_a x = \frac{\Delta_a x}{|x|} < \frac{h_a}{|x|}$, то число $\frac{h_a}{|x|}$ називають межею відносної

похибки: $E_a = \frac{h_a}{|x|}$.

Приклад 1.13. Визначення точності вимірювання

При вимірюванні довжини l і діаметра d деякого провідника одержали значення $l = 50 \pm 0.1$ м та $d = 2 \pm 0.1$ мм. Яке з вимірювань точніше?

Розв'язок. $E_l = \frac{0.1}{50} \cdot 100\% = 0.2\%$ $E_d = \frac{0.1}{2} \cdot 100\% = 5\%$

Отже, вимірювання довжини більш точне.

1.4.4. Звідки беруться похибки?

Нехай є реальний маятник, що робить згасаючі коливання, який починає рух у момент $t = t_0$. Потрібно знайти кут відхилення φ від вертикалі в момент t_1 . Рух маятника можливо описати наступним диференціальним рівнянням:

$$l \frac{d^2 \varphi}{dt^2} + \mu \frac{d\varphi}{dt} + g \sin \varphi = 0, \quad \text{де } l - \text{довжина маятника, } g - \text{прискорення}$$

вільного падіння, μ – коефіцієнт тертя, φ – величина кута відхилення маятника від вертикалі.

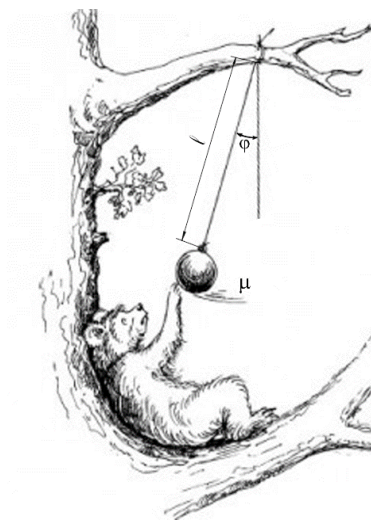


Рис. 1.15. Фізичний маятник

Похибки математичної моделі реального маятника

1. **Реальне тертя** характеризується незначною нелінійністю відносно швидкості маятника (*похибка моделі*).

2. **Довжина** маятника l може бути виміряна з деякою точністю, обумовленою вимірювальним приладом (*похибка вхідних даних*).

3. **Прискорення** вільного падіння g має різні значення в різних точках планети (*похибка вхідних даних*).

4. **Коефіцієнт тертя** μ також може бути заданий з деякою точністю (*похибка вхідних даних*).

5. **Чисельний метод** розв'язування диференціального рівняння має **похибку** (*похибка методу*).

1.4.5. Правила розрахунків похибки округлення

Додавання й віднімання наближених чисел. Розглянемо два наближених числа x і y , для яких відомі точні значення a і b і відповідно абсолютні похибки $\pm\Delta_a x$ й $\pm\Delta_b y$.

$$\pm\Delta_a x = x - a,$$

$$\pm\Delta_b = y - b.$$

Операція суми $s = x + y$ виконується з похибкою $\pm\Delta s$.

Операція різниці $r = x - y$ виконується з похибкою $\pm\Delta r$.

Оскільки знаки похибок доданків нам невідомі, то, для забезпечення достовірності кінцевого результату, ми повинні взяти найгірший випадок, коли похибки додаються.

1.4.6. Абсолютні похибки арифметичних операцій

1. *Абсолютна похибка суми* наближених чисел дорівнює сумі абсолютних похибок доданків з однаковими знаками.

$$x + y \Rightarrow \Delta s = +\Delta_a x + \Delta_b y$$

$$x + y \Rightarrow -\Delta s = (-\Delta_a x) + (-\Delta_b y)$$

2. Абсолютна похибка різниці наближених чисел дорівнює різниці абсолютних похибок доданків із протилежними знаками.

$$\begin{aligned}x - y &\Rightarrow \Delta r = (+\Delta_a x) - (-\Delta_b y) \\x - y &\Rightarrow -\Delta r = (-\Delta_a x) - (+\Delta_b y)\end{aligned}$$

3. Абсолютна похибка добутку:

$$xy \Rightarrow \Delta(xy) \approx y \cdot \Delta_a x + x \cdot \Delta_b y,$$

4. Абсолютна похибка частки:

$$x / y \Rightarrow \Delta(x/y) \approx \frac{\Delta_a x}{y} - \frac{x \Delta_b y}{y^2},$$

$$\text{де } |a - x| = \Delta_a x, \quad |b - y| = \Delta_b y.$$

1.4.7. Відносна похибка суми й різниці

Нехай $\sigma_a x = \frac{\pm \Delta_a x}{|a|}$ – відносна похибка наближеного числа x

та $\sigma_b y = \frac{\pm \Delta_b y}{|b|}$ – відносна похибка наближеного числа y .

Відносні похибки суми й різниці наближених чисел x і y визначають з виразів:

$$\sigma_+ = \frac{\pm \Delta_a x \pm \Delta_b y}{|x + y|} \quad \sigma_- = \frac{\pm \Delta_a x \mp \Delta_b y}{|x - y|}$$

Визначення відносної похибки суми. Відносну похибку суми визначають через відносну похибку доданків: $c = a + b \rightarrow z = x + y$

$$x = a \pm \Delta_a x = a \left(1 + \frac{\pm \Delta_a x}{a} \right) = a (1 \pm \sigma_a x)$$

$$y = b \pm \Delta_b y = b \left(1 + \frac{\pm \Delta_b y}{b} \right) = b(1 \pm \sigma_b y)$$

$$c(1 \pm \sigma_c z) = a(1 \pm \sigma_a x) + b(1 \pm \sigma_b y)$$

Поділимо на $(a + b)$ ліву та праву частину виразу

$$\left| \frac{c}{a+b} (1 \pm \sigma_c z) \right| = \left| \frac{a}{a+b} (1 \pm \sigma_a x) \right| + \left| \frac{b}{a+b} (1 \pm \sigma_b y) \right|$$

$$\left| \frac{c}{a+b} (1 \pm \sigma_c z) \right| = \left| \frac{a}{a+b} (1 \pm \sigma_a x) \right| + \left| \frac{b}{a+b} (1 \pm \sigma_b y) \right|$$

Справа розкриємо дужки:

$$\frac{c}{a+b} (1 + \sigma_c z) = \frac{a}{a+b} + \frac{a}{a+b} \sigma_a x + \frac{b}{a+b} + \frac{b}{a+b} \sigma_b y$$

Зведемо подібні члени

$$\frac{c}{a+b} (1 + \sigma_c z) = \frac{a+b}{a+b} + \frac{a}{a+b} \sigma_a x + \frac{b}{a+b} \sigma_b y$$

$$\frac{a+b}{a+b} (1 + \sigma_c z) = \frac{a+b}{a+b} + \frac{a}{a+b} \sigma_a x + \frac{b}{a+b} \sigma_b y$$

Скоротимо дроби

$$1 + \sigma_c z = 1 + \frac{a}{a+b} \sigma_a x + \frac{b}{a+b} \sigma_b y$$

$$\sigma_c z = \frac{a}{a+b} \sigma_a x + \frac{b}{a+b} \sigma_b y \approx \frac{a}{a+b} \sigma_a x + \frac{b}{a+b} \sigma_b y$$

Визначення відносної похибки різниці. Відносну похибку різниці визначають виразом:

$$c = a - b \rightarrow z = x - y$$

$$x = a \pm \Delta_a x = a \left(1 + \frac{\pm \Delta_a x}{a} \right) = a(1 \pm \sigma_a x)$$

$$y = b \pm \Delta_b y = b \left(1 + \frac{\pm \Delta_b y}{b} \right) = b(1 \pm \sigma_b y)$$

$$c(1 \pm \sigma_c z) = a(1 \pm \sigma_a x) - b(1 \pm \sigma_b y)$$

$$\left| \frac{c}{a-b} (1 \pm \sigma_c z) \right| = \left| \frac{a}{a-b} (1 \pm \sigma_a x) \right| - \left| \frac{b}{a-b} (1 \pm \sigma_b y) \right|$$

$$\frac{a-b}{a-b} (1 \pm \sigma_c z) = \frac{a-b}{a-b} + \frac{a}{a-b} \sigma_a x - \frac{b}{a-b} \sigma_b y$$

$$\sigma_c z = \frac{a}{a-b} \sigma_a x - \frac{b}{a-b} \sigma_b y \approx \frac{a}{a-b} \sigma_a x - \frac{b}{a-b} \sigma_b y$$

1.4.8. Відносні похибки добутку наближених чисел

Очевидно, що наближене число

$$x = a \pm \Delta_a x = a \left(1 + \frac{\pm \Delta_a x}{a} \right) = a(1 \pm \sigma_a x).$$

Тоді для добутку $c = ab \rightarrow z = xy$ одержуємо

$$\begin{aligned} z &= c(1 \pm \sigma_c z) = xy = ab(1 \pm \sigma_a x)(1 \pm \sigma_b y) = \\ &= ab(1 \pm (\sigma_a x + \sigma_b y) + \sigma_a x \cdot \sigma_b y) \end{aligned}$$

У практичних розрахунках нехтують малим доданком $\sigma_a x \cdot \sigma_b y$.

Виходячи із цього спрощення, можна сформулювати правила для обчислення відносних похибок добутку й частки.

Відносна похибка добутку наближених чисел дорівнює сумі відносних похибок множників: $z = x \cdot y$

$$\sigma_c z = \sigma_a x + \sigma_b y.$$

Відносна похибка частки наближених чисел дорівнює різниці відносних похибок діленого й дільника: $z = x/y$

$$\sigma_c z = \sigma_a x - \sigma_b y .$$

1.4.9. Обчислення з строгим врахуванням похибок

Теорема 1.2. Якщо $a \approx x$, $b \approx y$ з межами абсолютних похибок h_a та h_b , то межу h_{a+b} і межу h_{a-b} наближень $a + b \approx x + y$, $a - b \approx x - y$ обчислюють за формулами:

$$\begin{aligned} h_{a+b} &= h_a + h_b, \\ h_{a-b} &= h_a - h_b \end{aligned}$$

Теорема 1.3. Якщо $a \approx x$, $b \approx y$ з межами абсолютних похибок h_a та h_b , то межу $h_{a \cdot b}$ і межу $h_{a/b}$ наближень $a \cdot b \approx x \cdot y$, $a/b \approx x/y$ обчислюють за формулами:

$$\begin{aligned} h_{a \cdot b} &= h_a y + h_b x, \\ h_{a/b} &= \frac{h_a y - h_b x}{y^2}. \end{aligned}$$

Теорема 1.4. Якщо $a \approx x$, $b \approx y$, $b \neq 0$, $y \neq 0$ з межами відносних похибок E_a та E_b , то межі $E_{a \cdot b}$ і $E_{\frac{a}{b}}$ відносних похибок наближень $a \cdot b \approx x \cdot y$, $a/b \approx x/y$ обчислюють за формулами:

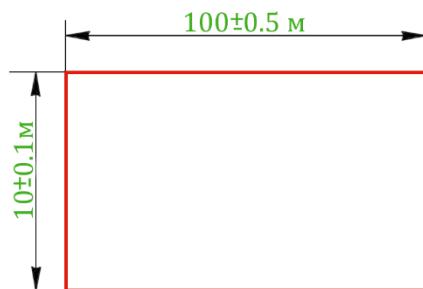
$$\begin{aligned} E_{a \cdot b} &= E_a + E_b \\ E_{a/b} &= E_a - E_b \end{aligned}$$

Теорема 1.5. Якщо $a \approx x$ з межею відносної похибки E_a , то $a^n \approx x^n$ з межею відносної похибки $n \cdot E_a$.

Теорема 1.6. Якщо $a \approx x$ з межею відносної похибки E_a , то $\sqrt[n]{a} \approx \sqrt[n]{x}$ з межею відносної похибки $1/n \cdot E_a$.

Приклад 1.14. Наближене обчислення периметра

Завдання. Для сторін прямокутника $a = 10$ і $b = 100$ знайдені



наближення $x = 10 \pm 0.1$ м та $b = 100 \pm 0.5$ м. Знайти наближене значення периметра і межі абсолютної і відносної похибок.

Розв'язок

$$P \approx (a + b) \cdot 2 = (10 + 100) \cdot 2 = 220 \text{ (м)},$$

$$h_p = (h_a + h_b) \cdot 2 = (0.1 + 0.5) \cdot 2 = 1.2 \text{ (м)},$$

$$P = 220 \pm 1.2 \text{ (м)},$$

$$E_p = \frac{h_p}{P} \cdot 100\% = \frac{1.2}{220} \cdot 100\% = 0.0055 \cdot 100\% \approx 0.6\%$$

Приклад 1.15. Визначення межі похибок при обчисленні площі

Завдання. При вимірюванні аудиторії було визначено $x = 12.2 \pm 0.03$ м та $y = 8.3 \pm 0.02$ м. Знайти площу аудиторії та визначити межі похибок.

Розв'язок

$$S = a \cdot b = 12.2 \cdot 8.3 = 101.26 \text{ (м}^2\text{)}$$

$$h_s = h_a b + h_b h_a = 0.03 \cdot 8.3 + 0.02 \cdot 12.2 = 0.493 \text{ (м}^2\text{)}$$

$$S = 101.26 \pm 0.493 \text{ (м}^2\text{)}$$

$$E_a = \frac{h_a}{a} \cdot 100\% = \frac{0.03}{12.2} \cdot 100\% \approx 0.25\%, \quad E_b = \frac{h_b}{b} \cdot 100\% = \frac{0.02}{8.3} \cdot 100\% \approx 0.25\%$$

$$E_S = E_a + E_b = 0.25\% + 0.25\% = 0.5\%$$

Приклад 1.16. Визначення межі похибок при обчисленні об'єму

Завдання. У заявці на виготовлення деталі сферичної форми вказаний радіус $r = 2.1 \pm 0.02$ см. Визначити об'єм деталі і вказати межі похибок наближення.

Розв'язок

$$V = \frac{4}{3}\pi r^3, \quad \frac{4}{3} = 1.33 \pm 0.004, \quad \pi = 3.14 \pm 0.002$$

$$V = 1.33 \cdot 3.14 \cdot (2.1)^3 = 38.6757882 \approx 38.68$$

$$E_4 = \frac{0.004}{1.33} \cdot 100\% \approx 0.31, \quad E_\pi = \frac{0.002}{3.14} \cdot 100\% \approx 0.07, \quad E_r = \frac{0.02}{2.1} \cdot 100\% \approx 0.96$$

$$E_V = 0.31 + 0.07 + 3 \cdot 0.96 = 3.25\%$$

$$E_V = 0.0031 + 0.0007 + 0.0096 = 0.0325$$

$$h_V = E_V \cdot V = 0.0325 \cdot 38.68 = 1.26$$

1.4.10. Значуща цифра

Визначення. Будь-яку цифру числа, записаного в позиційній системі числення, починаючи з першої зліва ненульової цифри, називають значущою.

$$123 = 1 \cdot 10^2 + 2 \cdot 10 + 3 \cdot 10^0 \quad 0456 = 0 \cdot 10^3 + 4 \cdot 10^2 + 5 \cdot 10^1 + 6 \cdot 10^0$$

Кількість значущих цифр у числі (S):

$$123 \Rightarrow S = 3, \quad 0456 \Rightarrow S = 3, \quad 0.3678 \Rightarrow S = 4, \quad 1.023 \Rightarrow S = 4$$

Представлення незначущих цифр у числі. У числі 0.012030 значущими є останні п'ять цифр. Перші два нулі не є значущими. Вони самі по собі не представляють кількісну величину, а служать для визначення розрядів інших цифр.

При зображенні цілих чисел, наприклад, числа 456000, нулі праворуч можуть служити як для позначення значущих цифр, так і для визначення розрядів інших цифр. Щоб уникнути цієї невизначеності, зазначене число

слід записати у вигляді $4.56e10^5$, якщо воно має три значущих цифри, або $4.5600e10^5$, якщо воно має п'ять значущих цифр.

1.4.11. Правила обчислення без строгого врахування похибок

Правило 1. При додаванні і відніманні наближених чисел найменший десятковий розряд результату повинен бути найбільшим серед десяткових розрядів, що виражають останні значущі цифри доданків.

Приклад 1.17. До правила 1.

Додавання чисел з різною кількістю значущих цифр:

$$2.366 + 17.25 + 11.563 + 78.345 \approx 109.52$$

Правило 2. При множенні та діленні наближених чисел у результаті потрібно зберігати стільки значущих цифр, скільки їх має те з наближених вхідних даних, у якому найменше число надійних значущих цифр.

Приклад 1.18. До правила 2.

Помножити наближені числа.

$$23.41 \cdot 0.0324 = 0.758484 \approx 0.758$$

Правила підрахунку цифр (за В. М. Брадїсом)

Ці правила даються з припущенням, що компоненти дій містять тільки вірні цифри й число дій невелике.

1) При додаванні й відніманні наближених чисел у результаті слід зберегти стільки десяткових знаків після коми, скільки їх у наближеному числі, даному з **найменшим числом десяткових знаків після коми**.

$$a = 234,341; \quad b = 12,23353; \quad a + b = 246,574353 \approx 246,574$$

$$a = 227,34345; \quad b = 10,2; \quad a - b = 217,14345 \approx 217,1$$

2) При множенні й діленні у результаті слід зберегти стільки значущих цифр, скільки їх у наближеному числі, даному з **найменшим числом вірних значущих цифр**.

Множення

$$a = 45,32;$$

$$b = 0,098;$$

$$a \cdot b = 45,32 \cdot 0,098 = 4,44136$$

$$a \cdot b = 4,4$$

Ділення

$$a = 134,234;$$

$$b = 432,1;$$

$$a/b = 134,234/432,1 = 0,3106549409858289$$

$$a/b = 0,3107$$

3) При піднесенні наближеного числа у квадрат у результаті слід зберегти стільки значущих цифр, скільки їх у основі степеня.

$$a = 1,2345; \quad a^2 = 1,52399025; \quad a^2 = 1,5240$$

4) При добуванні квадратного й кубічного кореня із наближеного числа у результаті слід зберегти стільки значущих цифр, скільки їх у підкореневому числі.

$$a = 675,6; \quad \sqrt{a} = \sqrt{675,6} = 25,9923065540555673$$

$$\sqrt{a} = 25,99$$

5) При обчисленні проміжних результатів слід зберегти на одну цифру більше, ніж рекомендують правила 1–4. В остаточному результаті ця «запасна цифра» відкидається.

$$a = 2,345; \quad b = 5,1238; \quad (ab)^2 = ? \quad ab = 2,345 \cdot 5,1238 = 12,015311$$

$$(12,015)^2 = 144,360225$$

$$\text{Відкидаємо запасну цифру: } (ab)^2 = 144,4$$

6) Якщо дані можна брати з довільною точністю, то для одержання результату з m вірними цифрами вхідні дані слід брати з таким числом

цифр, які згідно з попередніми правилами забезпечують $m+1$ цифру в результаті.

Приклад 1.19. *Визначення кількості значущих цифр за В. М. Брадїсом*

Завдання. Скільки потрібно взяти значущих цифр у числах a, b і c ,

щоб результат обчислення виразу $d = \frac{ab}{c}$ мав 4 вірних значущих цифри?

Розв'язок. Згідно з пунктом 6 правил Брадїса числа a, b й c повинні мати не менше ніж 5 значущих цифр.

1.4.12. Загальна формула для похибки

Основна задача теорії похибок полягає в наступному.

Якщо відомі похибки деякої системи величин, потрібно визначити похибку даної функції від цих величин.

Нехай дана диференційована функція

$$u = f(x_1, x_2, \dots, x_n)$$

і нехай $|\Delta x_i|$ ($i = 1, 2, \dots, n$) – абсолютні похибки аргументів функції.

Тоді абсолютна похибка функції

$$|\Delta u| = \left| f(x_1 + \Delta x_1, x_2 + \Delta x_2, \dots, x_n + \Delta x_n) - f(x_1, x_2, \dots, x_n) \right|.$$

$$|\Delta u| \approx |du| = \left| df(x_1, x_2, \dots, x_n) \right|$$

1. Використовуємо визначення диференціала $du = f'(x) \Delta x$.

2. На практиці $|\Delta x_i|$ – малі величини, добутками, квадратами

й вищими степенями яких можна знехтувати, тому $du = \sum_i df(x_i)$.

Беручи до уваги пункти 1 та 2, запишемо:

$$|\Delta u| = \sum_{i=1}^n \left| \frac{\partial f}{\partial x_i} \Delta x_i \right| \leq \sum_{i=1}^n \left| \frac{\partial f}{\partial x_i} \right| \cdot |\Delta x_i|. \quad (1.1)$$

Отже,
$$|\Delta u| \leq \sum_{i=1}^n \left| \frac{\partial f}{\partial x_i} \right| \cdot |\Delta x_i| \quad (1.2)$$

Тепер позначимо через h_{x_i} ($i = 1, 2, \dots, n$) межі абсолютних похибок аргументів x_i та через h_u – межу похибки функції u .

Тоді межа абсолютної похибки:

$$h_u = \sum_{i=1}^n \left| \frac{\partial u}{\partial x_i} \right| \cdot h_{x_i} \quad (1.3)$$

Для одержання виразу відносної похибки поділимо обидві частини виразу $|\Delta u| \leq \sum_{i=1}^n \left| \frac{\partial f}{\partial x_i} \right| \cdot |\Delta x_i|$ на u і, використовуючи табличну формулу

$(\ln f(\mathbf{x}))' = f'(\mathbf{x})/f(\mathbf{x})$, одержимо:

$$\sigma_u \leq \sum_{i=1}^n \left| \frac{\partial f}{\partial x_i} / u \right| |\Delta x_i| = \sum_{i=1}^n \left| \frac{\partial}{\partial x_i} \ln f(x_1, x_2, \dots, x_n) \right| |\Delta x_i| \quad (1.4)$$

Отже, межа відносної похибки функції u дорівнює:

$$E_u = \sum_{i=1}^n \left| \frac{\partial}{\partial x_i} \ln u \right| \cdot h_{x_i}$$

1.4.13. Приклади обчислення похибки

Приклад 1.20. Обчислення межі похибки для круга

Завдання. Дано круг радіусом R . Знайти площу круга і визначити межу похибки обчислення за умови, що вхідні дані задані в такий спосіб:

$$\pi = 3,14 \pm 0,0026, \quad R = 1,549 \pm 0,0005, \quad h_\pi = \pm 0.0026, \quad h_R = \pm 0.0005$$

Розв'язок

$$S = \pi \cdot R^2, \quad h_S = \sum_{i=1}^2 \left| \frac{\partial S}{\partial x_i} \right| \cdot h_{x_i} \text{ де } x_1 = \pi, x_2 = R$$

$$h_S = \left| \frac{\partial (\pi R^2)}{\partial \pi} \right| \cdot h_\pi + \left| \frac{\partial (\pi R^2)}{\partial R} \right| \cdot h_R$$

$$E_S = \frac{|h_S|}{S} \approx \frac{h_S}{S}$$

$$h_S = R^2 h_\pi + 2\pi R h_R = 2.3994 \cdot 0.0026 + 9.7277 \cdot 0.0005 = \\ = 0.00624 + 0.00486 = 0.0111$$

$$S = \pi \cdot R^2 = 3.14 \cdot 2.399 \approx 7.534$$

$$E_S = \frac{h_S}{S} = \frac{0.0111}{7.534} \cdot 100\% \approx 0,147\%$$

Приклад 1.21. *Обчислення межі похибки при обчисленні площі прямокутника*

Завдання. Дано прямокутник ABCD довжиною $L = 15 \pm 0.002$ см та висотою $H = 5 \pm 0.001$ см. Знайти площу прямокутника. Визначити межу абсолютної похибки h_S та межу відносної похибки E_S таких обчислень.

Розв'язок

$$S_{ABCD} = L \cdot H$$

$$h_S = \left| \frac{\partial (LH)}{\partial L} \right| \cdot h_L + \left| \frac{\partial (LH)}{\partial H} \right| \cdot h_H. \quad E_S = \frac{|h_S|}{S} \approx \frac{h_S}{S}.$$

$$h_S = H h_L + L h_H = 5 \cdot 0.002 + 15 \cdot 0.001 = 0.01 + 0.015 = 0.025.$$

$$S \approx L \cdot H = 5 \cdot 15 = 75. \quad E_S = \frac{h_S}{S} = \frac{0.025}{75} = 0,00033 = 0,033\%.$$

Приклад 1.22. Обчислення межі похибки при обчисленні периметра прямокутника

Завдання. Дано прямокутник ABCD довжиною $L = 15 \pm 0.002$ см та висотою $H = 5 \pm 0.001$ см. Знайти периметр прямокутника. Визначити межу абсолютної похибки h_P та межу відносної похибки таких обчислень.

Розв'язок

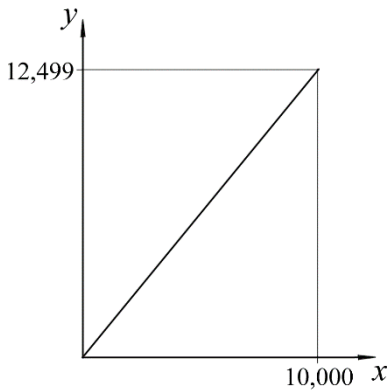
$$P_{ABCD} = 2L + 2H$$

$$h_P = 2 \left| \frac{\partial(L+H)}{\partial L} \right| h_L + 2 \left| \frac{\partial(L+H)}{\partial H} \right| h_H = 2h_L + 2h_H$$

$$h_P = 2h_L + 2h_H = 2 \cdot 0.002 + 2 \cdot 0.001 = 0.004 + 0.002 = 0.006$$

$$P = 2L + 2H = 2 \cdot 15 + 2 \cdot 5 = 40, \quad E_P = \frac{h_P}{P} = \frac{0.006}{40} = 0.015\%$$

Приклад 1.23. Обчислення межі похибки при обчисленні модуля вектора



Завдання. Нехай задано вектор координатами.

$$V = (x, y) = (10.000 \pm 0.001, 12.499 \pm 0.0012)$$

Визначити величину його модуля.

Визначити межу абсолютної похибки і межу відносної похибки обчислень.

Розв'язок

$$m = \sqrt{x^2 + y^2}, \quad h_m = \left| \frac{\partial m}{\partial x} \right| h_x + \left| \frac{\partial m}{\partial y} \right| h_y$$

$$\frac{\partial m}{\partial x} = \frac{\partial (x^2 + y^2)^{\frac{1}{2}}}{\partial x} = \frac{1}{2\sqrt{x^2 + y^2}} \frac{\partial (x^2 + y^2)}{\partial x} = \frac{2x}{2\sqrt{x^2 + y^2}} = \frac{x}{\sqrt{x^2 + y^2}}$$

$$\frac{\partial m}{\partial y} = \frac{y}{\sqrt{x^2 + y^2}}$$

$$h_m = \frac{x \cdot h_x + y h_y}{\sqrt{x^2 + y^2}} = \frac{10 \cdot 0.001 + 12.499 \cdot 0.0012}{\sqrt{100 + 156}} =$$

$$= \frac{0.01 + 0.0145}{\sqrt{256}} = \frac{0.0245}{16} = 0.0015, \quad m = \sqrt{100 + 156} = 16$$

$$E_m = \frac{h_m}{m} = \frac{0.0015}{16} \approx 0.01\%$$

Контрольні запитання

1. Користуючись правилами наближених обчислень, виконати такі обчислення: $a = 125.6784$, $b = 115.371$, $a + b = ?$

2. При вимірюванні ділянки землі визначили, що її довжина $x = 122.2 \pm 0.12$ м, а ширина $y = 11.1 \pm 0.09$ м. Знайти площу ділянки та визначити межі похибок.

3. Для виготовлення сталевих кульок задано радіус $r = 2 \pm 0.01$ см. Визначити об'єм деталі та вказати межу абсолютної похибки та межу відносної похибки.

4. Даний радіус R . Знайти площу круга й визначити межу абсолютної похибки та межу відносної похибки обчислення за умови, що $R = 1.5 \pm 0.005$, $\pi = 3.14 \pm 0.003$.

5. Даний прямокутник ABCD з довжиною $L = 15 \pm 0.002$ й висотою $H = 5 \pm 0.001$. Знайти площу прямокутника. Визначити граничну абсолютну й відносну похибки обчислення.

2. ОСНОВИ ТЕОРІЇ АЛГОРИТМІВ

2.1. Загальні поняття теорії алгоритмів

2.1.1. Формалізація поняття алгоритму через універсальні алгоритмічні моделі

Інтуїтивне поняття алгоритму має цілий ряд недоліків. Очевидно, що таке поняття, як елементарність кроків, яке використовують при описі загальних властивостей алгоритмів, також потребує уточнення. Зрозуміло також, що словесні визначення алгоритмів міститимуть нові поняття, які знову потребуватимуть уточнення і т. д. Починаючи з 30-х років було запропоновано кілька уточнень поняття алгоритму. Вважається, що всі вони досить повно окреслюють основні риси інтуїтивного поняття алгоритму.

Алгоритм – деякий точно визначений інструмент, за допомогою якого можна конструювати будь-які інтуїтивно зрозумілі процедури, що породжують (обчислюють) та розпізнають (розв’язують).

Алгоритм – точний набір інструкцій, що описують порядок дій виконавця для отримання розв’язку задачі за скінченний час.

Алгоритм – зрозумілі й точні приписи (інструкції) виконавцеві зробити скінченне число кроків, спрямованих на розв’язування поставленої задачі.

Алгоритм – це послідовність дій, яка або приводить до розв’язку задачі, або пояснює, чому цей розв’язок одержати не можна.

«**Алгоритм** – це скінченний набір правил, який визначає послідовність операцій для розв’язування конкретної множини задач і має п’ять важливих рис: скінченність, визначеність, ввід, вивід, ефективність» (Д. Кнут).

«**Алгоритм** – це будь-яка система обчислень, які виконуються по строго визначених правилах, що після деякого числа кроків явно (завідомо) приводить до розв’язку поставленої задачі» (А. Колмогоров).

«**Алгоритм** – це точний припис (інструкція), що визначає обчислювальний процес, який йде від варійованих (змінних) вхідних даних до шуканого результату» (А. Марков).

«**Алгоритм** – точний припис (інструкція) про виконання в певному порядку деякої системи операцій, що ведуть до розв’язку всіх задач даного типу» (Філософський словник / Під ред. Розенталя).

Дійсно, усі формальні визначення алгоритму в деякому сенсі еквівалентні один одному. Тому в теорії алгоритмів застосовується інший підхід: вибирається скінченний набір вхідних об’єктів, які оголошуються елементарними, і скінченний набір способів побудови з них нових об’єктів. Цей метод був уже використаний у теорії множин і отримав назву *конструктивного підходу*.

Алгоритмічні моделі, які претендують на право вважатися формалізацією поняття «алгоритм», повинні бути універсальними, тобто допускати опис будь-яких алгоритмів. Можна виділити три основних типи універсальних алгоритмічних моделей, що різняться вхідними евристичними міркуваннями щодо того, що таке алгоритм.

Перший тип зв’язує поняття алгоритму з найбільш традиційними поняттями математики – обчисленнями й числовими функціями. Найбільш розвинена й вивчена модель цього типу – *рекурсивні функції* – є історично першою формалізацією поняття алгоритму.

Другий тип пов’язаний з розвитком обчислювальної техніки й заснований на уявленні про алгоритм як про деякий детермінований пристрій, здатний виконувати в кожний окремий дискретний момент часу досить примітивні операції. Таке розуміння не залишає сумнівів у однозначності алгоритму й елементарності його кроків. Крім того,

евристика цієї моделі близька до ЕОМ а, отже, до інженерної інтуїції. Основною теоретичною моделлю цього типу є створена в 30-х роках концепція *машини Тьюринга*. Саме машина Тьюринга була моделлю сучасної ЕОМ і сприяла розвитку сучасної обчислювальної техніки.

Третій тип алгоритмічних моделей – це перетворення слів у довільних алфавітах, у яких елементарними операціями є підстановки, тобто заміни частини слова (підслова) іншим словом. Переваги цього типу моделей полягають у максимальній абстрактності й можливості застосувати поняття алгоритму до об'єктів довільної, не обов'язково числової природи. Прикладами моделей цього типу є *канонічні системи Поста* й *нормальні алгоритми Маркова*. Водночас спільність формалізації в конкретній моделі не втрачається і можна довести зведення одних моделей до інших, тобто показати, що будь-який алгоритм, описаний засобами однієї моделі, може бути описаний засобами іншої.

Завдяки взаємному зведенню моделей у загальній теорії алгоритмів удалося виробити інваріантну стосовно моделей систему понять, що дозволяє говорити про властивості алгоритмів незалежно від того, яка формалізація алгоритму обрана. Ця система понять заснована на понятті *обчислюваної функції*, тобто функції, для обчислення якої існує алгоритм.

2.1.2. Обчислювана функція

Ми будемо розглядати алгоритми, що мають справу тільки з натуральними числами. Можна довести, що це не є втратою спільності, тому що об'єкти іншої природи можна закодувати натуральними числами. Для користувачів комп'ютерів таке твердження повинно бути очевидним [12].

Нехай N позначає множину натуральних чисел $\{0, 1, 2, \dots\}$. Об'єкти, які ми будемо розглядати, це функції з областю визначення $D_f \subseteq N^n$ (n –

ціле додатне число) і з областю значень $R_f \subseteq N$. Такі функції будемо називати k -місними частковими. Слово «часткова» повинно нагадати про те, що функція визначена на підмножині N (звичайно, в окремому випадку може бути $D_f = N^k$, тоді функцію називають *всюди визначеною*).

Визначення. Назвемо n -місну функцію $f: N^n \rightarrow N$ ефективно обчислюваною (або просто обчислюваною), якщо існує алгоритм A , що її обчислює, тобто такий алгоритм A , що:

1. Якщо на вхід алгоритму A надійшов вектор $\mathbf{x} = (x_1, x_2, \dots, x_n)$ з D_f , то обчислення повинно закінчитися після скінченного числа кроків і видати $f(\mathbf{x})$.

2. Якщо на вхід алгоритму A надійшов вектор x , що не належить області визначення D_f , то алгоритм A ніколи не закінчується.

Кілька зауважень із приводу цього визначення:

1. Поняття обчислюваності визначається тут для часткових функцій (областю визначення яких є деяка підмножина натурального ряду). Наприклад, ніде не визначена функція є обчислюваною, а прикладом відповідного алгоритму A є програма, яка завжди зациклюється.

2. Можна було б змінити визначення, сказавши так: «якщо $f(x)$ не визначено, то або алгоритм A не зупиняється, або зупиняється, але нічого не друкує на виході». Насправді від цього нічого б не змінилося (замість того, щоб зупинитися, нічого не надрукувавши, алгоритм може зациклюватися).

3. Входами й виходами алгоритмів можуть бути не тільки натуральні числа, але й двійкові рядки (слова в алфавіті $\{0, 1\}$), скінченні послідовності слів і взагалі будь-які «конструктивні об'єкти».

4. Множину ефективно обчислюваних функцій ми не ототожнюємо із множиною «практично обчислюваних функцій», оскільки не накладаємо на першу множину жодних обмежень, пов'язаних із сучасними

обчислювальними машинами. Незважаючи на те, що кожне вхідне натуральне число повинно бути скінченним, не передбачається верхня границя розміру цього числа, так, наприклад, кількість цифр числа може бути більшою за число електронів у Всесвіті. Так само немає верхньої границі на число кроків, які може зробити алгоритм для конкретних x з області визначення.

Приклад 2.1. Задана множина натуральних чисел N і деяка функція $y = f(x, a, b) = ax + b$.

Область визначення $D_f \subseteq N^4 = U$ й область значень $R \subseteq N$.

Функція f обчислювана, оскільки існує алгоритм $f:(a, b, x) \rightarrow y$, який по будь-якій трійці (a, b, x) обчислює y . Обчислюваність, у цьому випадку, може бути легко поширена на множину дійсних чисел D .

2.1.3. Розв'язність (можливість розв'язання)

Множина A *розв'язна* на M , тобто для $A \subseteq M$ існує процедура (алгоритм), яка для кожного $m \in M$ визначає, чи належить m до A ($m \in A$) чи ні, тобто $m \notin A$.

Проблема *розв'язності* зводиться до *обчислюваності* відповідного предиката:

$$P(m) = \begin{cases} 1, & \text{true, } m \in A, \\ 0, & \text{false, } m \notin A. \end{cases}$$

Приклад 2.2. Нехай A – множина розв'язків рівняння $y = ax + b$.

Існує тривіальний алгоритм Π перевірки для кожної четвірки (a, b, x, y) у вигляді підстановки цих чисел у рівняння й виконання зазначених у рівнянні дій, який *обчислює* предикат і в такий спосіб вирішує проблему *розв'язності* для заданого рівняння.

Приклад 2.3. Теорема Ферма

Чи можна запропонувати для рівняння $x^n + y^n = z^n$, де $x, y, z, n \in \mathbb{N}$ алгоритм П, що породжує всі розв'язки рівняння при заданому n ?

Розв'язність. Розв'язність множини розв'язків рівняння вирішується просто, тому що є тривіальний алгоритм, який обчислює предикат. Наприклад,

$$P(x, y, z, n) \rightarrow P(1, 1, 2, 1) \rightarrow \text{true}, \text{ оскільки } 1^1 + 1^1 = 2^1$$

$$P(x, y, z, n) \rightarrow P(1, 1, 2, 2) \rightarrow \text{false}, \text{ оскільки } 1^2 + 1^2 \neq 2^2$$

Отже, множина розв'язків рівняння $x^n + y^n = z^n$ розв'язна.

Обчислюваність. Хоча обчислюваність для теореми доведена, але алгоритм настільки складний, що поки не допускає практичного застосування.

Тому рівняння, що задане теоремою Ферма, не обчислюване.

Приклад 2.4. Чи має рівняння $x^3 + y^3 + z^3 = 29$ цілий розв'язок?

Легко перевірити (*обчислити предикат*), що є розв'язок

$$(x, y, z) \rightarrow (3, 1, 1).$$

$$P(x, y, z, n) \rightarrow P(3, 1, 1, 3) \rightarrow \text{true}, \quad \text{оскільки} \quad 3^3 + 1^3 + 1^3 = 29;$$

$$27 + 1 + 1 = 29$$

Приклад 2.5. Рівняння $x^3 + y^3 + z^3 = 30$ також має розв'язок, знайдений у 1990 році.

$$P(x, y, z, n) \rightarrow P(-283059965, -2218888517, 2220422932, 3) \rightarrow \text{true}$$

$$(-283059965)^3 + (-2218888517)^3 + (2220422932)^3 = 30$$

Приклад 2.6. Для рівняння $x^3 + y^3 + z^3 = 33$ розв'язки не відомі.

Ці задачі пов'язані з десятою проблемою Гільберта й у 1970 році Ю. Матіясевич довів, що *не існує алгоритму* знаходження розв'язків таких задач!

1) якщо A обчислювана в M , то A розв'язна в M ;

2) якщо A розв'язна в M , то із цього *не випливає* її обчислюваність.

Проблеми розв'язності й обчислюваності множин вимагають більш точного визначення типу алгоритму.

Контрольні запитання

1. Дайте визначення алгоритму в залежності від сфери його застосування.
2. Вкажіть три основних типи універсальних алгоритмічних моделей.
3. Нехай задана функція $y = f(k, l) = 2k + 3l$ на множині натуральних чисел N . Чи є дана функція обчислюваною?
4. Яка функція є розв'язною функцією?
5. Чи всі розв'язні функції є обчислюваними?

2.2. Перша універсальна алгоритмічна модель.

Рекурсивні функції

Будь-який алгоритм однозначно ставить у відповідність вхідним даним (у випадку, якщо визначений на них) певний результат. Тому з кожним алгоритмом однозначно зв'язана функція, яку він обчислює. Дослідження цих питань привело до створення в 30-х роках минулого століття *теорії рекурсивних функцій*. У цій теорії, як і взагалі в теорії алгоритмів, прийнятий *конструктивний*, фінітний підхід, основною рисою якого є те, що вся множина досліджуваних об'єктів (у цьому випадку функцій) будується з скінченного числа вхідних об'єктів – базису – за допомогою простих операцій, ефективна обчислюваність яких достатньо очевидна. Операції над функціями будемо називати *операторами*.

Будемо розглядати тільки числові функції, тобто функції, аргументи й значення яких належать множині натуральних чисел N (у теорії рекурсивних функцій вважають $N=0, 1, 2, \dots$).

Рекурсивним визначенням функції прийнято називати таке визначення, при якому значення функції для даних аргументів визначаються

значеннями функції для більш простих аргументів (уже обчислених) або значеннями більш простих функцій [13].

Найпростішим прикладом рекурсивного визначення є числа Фібоначчі, що представляють собою послідовність чисел $f(n)$, які задовольняють умовам: $f(0)=1$, $f(1)=1$, $f(n+2)=f(n)+f(n+1)$, тобто числа 1, 1, 2, 3, 5, 8, 13 ... Кожне наступне число є сумою двох попередніх чисел.

Визначимо тепер конкретний набір засобів, за допомогою яких відбувається побудова обчислюваних функцій.

2.2.1. Примітивно рекурсивні функції

Базова система функцій Геделя

Очевидно, що до обчислюваних функцій слід віднести всі константи, тобто 0 і всі натуральні числа 1, 2, 3 Проте у прямому включенні нескінченної множини констант у базис немає необхідності, оскільки достатньо нуля функції $0(x)=0$ і функції слідування $S(x)=x+1$, щоб одержати весь натуральний ряд.

Крім того, у базис слід включити сімейство $\{I_m^n\}$ функцій проектування (або функцій введення фіктивних змінних)

$$I_m^n(x_1, x_2, \dots, x_n) = x_m \quad 1 \leq m \leq n.$$

Запропоновані Геделем усюди визначені числові функції назвемо найпростішими [14]:

- 1) $0(x)=0$ – нуль-функція.
- 2) $S(x)=x+1$ – функція слідування.
- 3) $I_m^n(x_1, x_2, \dots, x_n) = x_m$, де $m = 1, \dots, n$ – функція проектування.

Усі ці функції обчислювані в інтуїтивному сенсі.

Визначимо тепер оператори. Вони мають ту властивість, що, застосовуючи їх до функцій, обчислюваних в інтуїтивному сенсі, одержуємо функції, також обчислювані в інтуїтивному сенсі.

Оператор суперпозиції. Суперпозиція є потужним засобом одержання нових функцій із наявних. Нагадаємо, що суперпозицією називають будь-яку підстановку функцій у функції. Оператором суперпозиції P_m^n називають підстановку у функцію від m змінних m функцій від n тих самих змінних. Наприклад, дана m -місна функція: $h(x_1, x_2, \dots, x_m)$ і m функцій від n змінних:

$$g_1(x_1, x_2, \dots, x_n), \dots, g_m(x_1, x_2, \dots, x_n)$$

Застосувавши до цих функцій оператор суперпозиції, одержимо:

$$P_m^n(h, g_1, g_2, \dots, g_m) = h(g_1(x_1, x_2, \dots, x_n), \dots, g_m(x_1, x_2, \dots, x_n)) = f(x_1, x_2, \dots, x_n).$$

У цьому випадку говорять, що n -місна функція $f(x_1, x_2, \dots, x_n)$ отримана за допомогою оператора суперпозиції з m -місної функції $h(x_1, x_2, \dots, x_m)$ й n -місних функцій $g_1(x_1, x_2, \dots, x_n), \dots, g_m(x_1, x_2, \dots, x_n)$, якщо

$$f(x_1, x_2, \dots, x_n) = h(g_1(x_1, x_2, \dots, x_n), \dots, g_m(x_1, x_2, \dots, x_n)).$$

Оператор примітивної рекурсії. Оператор примітивної рекурсії R_n визначає $(n+1)$ -місну функцію f через n -місну функцію g і $(n+2)$ -місну функцію h у такий спосіб:

$$\left. \begin{aligned} f(x_1, \dots, x_n, 0) &= g(x_1, x_2, \dots, x_n); \\ f(x_1, \dots, x_n, y+1) &= h(x_1, x_2, \dots, x_n, y, f(x_1, x_2, \dots, x_n, y)) \end{aligned} \right\} (2.1)$$

Цю пару рівностей називають схемою примітивної рекурсії.

Той факт, що функція f визначена схемою (2.1), виражають рівністю:

$$f(x_1, \dots, x_n, y) = R_n(g, h).$$

У випадку, коли $n=0$, тобто обумовлена функція f є одномісною, схема (2.1) набуває більш простого вигляду:

$$f(0) = C; \quad f(y+1) = h(y, f(y)); \quad (2.2)$$

де C – константа.

Схеми (2.1) і (2.2) визначають f рекурсивно не тільки через інші функції g і h , але й через значення f у попередніх точках: значення f у точці $y+1$ залежить від значення f у точці y . Для обчислення $f(x_1, x_2, \dots, x_n, k)$ знадобиться $k+1$ обчислень за схемою (2.1) для $y = 0, 1, \dots, k$. Приклад – обчислення $n!$.

Істотним в операторі примітивної рекурсії є те, що незалежно від числа змінних у f рекурсія ведеться тільки по одній змінній y , інші n змінних x_1, x_2, \dots, x_n на момент застосування схем (2.1) і (2.2) зафіксовані й відіграють роль параметрів.

Функцію називають *примітивно рекурсивною*, якщо вона може бути отримана з нуль-функції $0(x)$, функції слідування $S(x)$ та функції проектування I_m^n за допомогою скінченного числа застосувань операторів суперпозиції й примітивної рекурсії.

Цьому визначенню можна надати більш формального індуктивного вигляду.

1. Функції $0(x)$, $S(x)$ і $I_m^n(x)$ для всіх натуральних n, m , де $m \leq n$, є примітивно рекурсивними.

2. Якщо $g_1(x_1, x_2, \dots, x_n), \dots, g_m(x_1, x_2, \dots, x_n), \quad h(x_1, x_2, \dots, x_m)$ примітивно рекурсивні, то $P_m^n(h, g_1, \dots, g_m)$ – примітивно рекурсивні функції для будь-яких натуральних n, m .

3. Якщо $g_1(x_1, \dots, x_n)$ й $h(x_1, \dots, x_n, y, z)$ – примітивно рекурсивні функції, то $R_n(g, h)$ – примітивно рекурсивна функція.

4. Інших примітивно рекурсивних функцій немає.

З такого індуктивного опису легко отримати процедуру, що породжує всі примітивно рекурсивні функції.

Приклад 2.7. Покажемо, що додавання, множення, піднесення до степеня примітивно рекурсивні.

1. Функція $f_+(x, y) = x + y$ примітивно рекурсивна.

$$\text{Дійсно, } f_+(x, 0) = x + 0 = x = I_1^1(x),$$

$$f_+(x, y + 1) = x + (y + 1) = (x + y) + 1 = S(x + y) = S(f_+(x, y))$$

Це означає, що функція $x + y$ будується з двох примітивно рекурсивних функцій I_1^1 , $h(x, y, z) = z + 1 = S(z)$ за допомогою примітивної рекурсії. Тому $x + y$ примітивно рекурсивна функція.

2. Множення $f_\times(x, y) = xy$ примітивно рекурсивне:

$$f_\times(x, 0) = x \cdot 0 = 0(x);$$

$$f_\times(x, y + 1) = x \cdot (y + 1) = xy + x = f_\times(x, y) + x = f_+(f_\times(x, y), x).$$

У термінах оператора примітивної рекурсії одержимо:

$$g(x) = 0(x), \quad h(x, y, z) = z + x$$

Оскільки із прикладу 2.7.1 випливає, що функція суми є примітивно рекурсивною, $h(x, y, z) = z + x$ – це примітивно рекурсивна функція.

Звідси випливає, що $f_\times(x, y) = xy$ є примітивно рекурсивною.

3. Піднесення до степеня $f_{\text{exp}}(x, y) = x^y$ примітивно рекурсивне:

$$f_{\text{exp}}(x, 0) = 1; \quad f_{\text{exp}}(x, y + 1) = x^y \cdot x = f_\times(x, f_{\text{exp}}(x, y)).$$

У термінах оператора примітивної рекурсії одержимо:

$$g(x) = 0(x) = 1 - \text{константи входять у базисний набір Геделя.}$$

$$h(x, y, z) = f_\times(x, z) - \text{примітивно рекурсивна, що доведено раніше.}$$

Отже, $f_{\text{exp}}(x, y) = x^y$ – примітивно рекурсивна.

2.2.2. Частково рекурсивні функції

Клас функцій, одержуваних з базових застосуванням скінченного числа операцій двох типів – *суперпозиції* й *примітивної рекурсії*, називають класом *примітивно рекурсивних функцій*. У цей клас, крім наведених вище, входять багато функцій. Загальною властивістю таких функцій є те, що вони всюди визначені.

Не всі функції, значення яких можуть бути обчислені, належать класу примітивно рекурсивних функцій. В операції R_n рекурсія проводиться по одній змінній. Якщо побудувати схему з рекурсією по двох змінних (подвійна рекурсія, рекурсія 2-го ступеня), то за її допомоги будемо отримувати функції, що не належать у загальному випадку до класу примітивно рекурсивних.

Такі функції належать до класу частково рекурсивних функцій. Для визначення їх обчислюваності необхідно до відомих нам операторів суперпозиції й примітивної рекурсії додати оператор мінімізації.

2.2.3. Оператор мінімізації

Оператор мінімізації μ має один операнд, $f = \mu(g)$. Значення функції f на заданому наборі аргументів x_1, x_2, \dots, x_n знаходять у такий спосіб. Спочатку за допомогою функції g формують рівняння $g(x_1, x_2, \dots, x_{n-1}, y) = x_n$, а потім відшуковують його розв'язок відносно змінної y . Якщо таких розв'язків декілька, то беруть мінімальний з них (звідси й назва оператора); і його вважають значенням функції на даному наборі аргументів, $f(x_1, x_2, \dots, x_n)$.

Іншими словами, операція мінімізації ставить у відповідність частковій функції $g : N^{n+1} \rightarrow N$ часткову функцію $f : N^n \rightarrow N$.

Операцію мінімізації зазвичай записують за допомогою оператора мінімізації

$$f(x_1, x_2, \dots, x_n) = \mu[g(x_1, x_2, \dots, x_{n-1}, y) = x_n],$$

$$\text{де } \mu[g] = \min_{y \in N} \{g(x_1, \dots, x_{n-1}, y) = x_n\}.$$

Може трапитися, що рівняння не має жодного розв'язку. У цьому випадку вважається, що функція f не визначена на заданому наборі аргументів.

Приклад 2.8.

Функція $g(x) = \begin{cases} x-1, x > 0, \\ \text{не визначено}, x = 0 \end{cases}$ – частково рекурсивна.

Дійсно, $g(x) = \mu[S(x)] = \min_{y \in N} \{S(y) \equiv y + 1 = x\}$

Як видно, функція g отримана з найпростішої функції S за допомогою операції мінімізації.

Наведений приклад показує, що оператор мінімізації може перетворити всюди визначену функцію в часткову, тобто вивести за межі множини примітивно рекурсивних функцій.

Кожна примітивно рекурсивна функція є частково рекурсивною. Зворотне невірно, оскільки до класу частково рекурсивних функцій відповідно до визначення входить часткова функція $\mu[S(x)]$ із прикладу 2.8 і, наприклад, ніде не визначена функція

$$f(x) = \min_{y \in N} \{x + 1 + y = 0\}.$$

Множину функцій, одержуваних застосуванням до базового набору скінченного числа операцій суперпозиції, примітивної рекурсії й мінімізації називають множиною *частково рекурсивних* функцій.

Звертаємо увагу на те, що мінімізація може бути організована як послідовний алгоритмічний процес. Дійсно, послідовно знаходимо $g(x_1, \dots, x_{n-1}, 0), g(x_1, \dots, x_{n-1}, 1), \dots, g(x_1, \dots, x_{n-1}, y), \dots$

Найменше a , для якого $g(x_1, \dots, x_{n-1}, a) = x_n$ – це значення для $f(x_1, x_2, \dots, x_n)$.

Приклади обчислення оператора мінімізації

Нехай існує деякий механізм обчислення частково рекурсивної функції, причому значення функції не визначене тоді й тільки тоді, коли цей механізм працює нескінченно, не видаючи жодного конкретного результату.

Фіксуємо будь-які значення x_1, \dots, x_{n-1} для перших $n-1$ аргументів функції f і розглянемо рівняння:

$$f(x_1, \dots, x_{n-1}, y) = x_n$$

Щоб знайти розв'язок y (натуральне) цього рівняння, будемо обчислювати за допомогою зазначеного вище «механізму» послідовно значення $f(x_1, \dots, x_{n-1}, y)$ для $y = 0, 1, 2, \dots$. Найменше значення a , для якого $f(x_1, \dots, x_{n-1}, a) = x_n$, позначимо через

$$\mu y (f(x_1, \dots, x_{n-1}, y) = x_n)$$

Описаний процес знаходження виразу $\mu y (f(x_1, \dots, x_{n-1}, y) = x_n)$ буде тривати нескінченно в наступних випадках:

- Значення $f(x_1, \dots, x_{n-1}, 0)$ не визначене;
- Значення $f(x_1, \dots, x_{n-1}, y)$ для $y = 0, 1, \dots, a-1$ визначені, але відмінні від x_n , а значення $f(x_1, \dots, x_{n-1}, a)$ – не визначене;

- Значення $f(x_1, \dots, x_{n-1}, y)$ визначені для всіх $y = 0, 1, 2, \dots$ і відмінні від x_n .

У всіх цих випадках значення виразу $\mu y(f(x_1, \dots, x_{n-1}, y) = x_n)$ вважають невизначеним. У інших випадках описаний процес обривається й дає найменший розв'язок $y = a$ рівняння $f(x_1, \dots, x_{n-1}, y) = x_n$.

Цей розв'язок і буде значенням виразу $\mu y(f(x_1, \dots, x_{n-1}, y) = x_n)$.

Наприклад, для функції різниці $f(x, y) = x - y$ відповідно до зазначеного змісту символу μ , для всіх x, y маємо:

$$f(x, y) = x - y = \mu z(y + z = x)$$

Підрахунок значень функції $f(x, y) = x - y$ у цьому випадку буде проходити по двох сценаріях.

Приклад 2.9. Нехай $x = 5$, $y = 3$. Послідовно, починаючи з $z = 0$, перебираємо можливі значення для знаходження $\mu z(y + z = x)$:

- при $z = 0$ одержимо вираз $3+0=5$, його значення «неправда»;
- при $z = 1$ одержимо вираз $3+1=5$, його значення «неправда»;
- при $z = 2$ одержимо вираз $3+2=5$, його значення «істина», процес обривається, і одержуємо результат: $\mu z(3 + z = 5) = 2$.

Отже, не вміючи робити операцію віднімання, ми змогли порахувати значення часткової функції $f(x, y) = x - y$ для двох заданих значень аргументів.

Приклад 2.10. Нехай $x = 3$, $y = 5$. Послідовно, починаючи з $z = 0$, перебираємо можливі значення для знаходження $\mu z(y + z = x)$:

- при $z = 0$ одержимо вираз $5+0=3$, його значення «неправда»;
 - при $z = 1$ одержимо вираз $5+1=3$, його значення «неправда»;
 - при $z = 2$ одержимо вираз $5+2=3$, його значення «неправда»;
- і т. д. – тобто при збільшенні Z ситуація, при якій вираз $5 + Z=3$ буде мати значення «істина», так і не настане, а отже, значення функції $\mu z(5 + z = 3)$ так і не буде знайдено.

Приклад 2.10 – це ілюстрація випадку, при якому оператор мінімізації не дає результату саме тоді, коли такий результат не може бути отриманий у принципі, з тієї причини, що в заданій точці функція дійсно не визначена.

Це третій випадок, у якому процес пошуку виразу $\mu y(f(x_1, \dots, x_{n-1}, y) = x_n)$ триває нескінченно.

Значення ж виразу $\mu y((y + 1)(y - (x + 1)) = 0)$ не визначене, оскільки вже при $y = 0$ значення терма $1 \cdot (0 - (x + 1))$ для будь-якого x не визначене. У той же час рівняння $(y + 1)(y - (x + 1)) = 0$ має розв'язок $y = x + 1$, але воно не збігається зі значенням виразу $\mu y((y + 1)(y - (x + 1)) = 0)$.

Цей приклад показує, що для часткових функцій $f(x_1, \dots, x_{n-1}, y)$ вираз $\mu y(f(x_1, \dots, x_{n-1}, y) = x_n)$, строго кажучи, не є найменшим розв'язком рівняння $f(x_1, \dots, x_{n-1}, y) = x_n$. Якщо ж функція $f(x_1, \dots, x_{n-1}, y)$ усюди визначена й рівняння $f(x_1, \dots, x_{n-1}, y) = x_n$ має розв'язок, то $\mu y(f(x_1, \dots, x_{n-1}, y) = x_n)$ є найменшим розв'язком для цього рівняння.

Звідси виникає закономірне бажання використовувати під оператором мінімізації тільки всюди визначені функції. Найкращий спосіб переконатися

в тому, що функція всюди визначена – використовувати завідомо примітивно рекурсивні функції (або функції, примітивна рекурсивність яких уже доведена раніше або впливає зі способу їх задавання).

У цьому випадку значення $f(x_1, \dots, x_n)$ буде невизначеним тільки в одному із трьох випадків, а саме, якщо значення $f(x_1, \dots, x_{n-1}, y)$ визначені для всіх $y = 0, 1, 2, \dots$ і відмінні від x_n .

Для цього наявну функцію потрібно перетворити в такий спосіб (шляхом переносу доданків, множення обох частин, і т. д.), щоб по обидві сторони рівності стояли примітивно рекурсивні (а отже, усюди визначені) вирази.

У загальному вигляді, під оператором мінімізації одержимо яесь рекурсивне рівняння, частини якого залежать від змінних $(x_1, \dots, x_{n-1}, x_n, y)$. Це рівняння набуває значення «істина» або «неправда» при послідовному вибиранні значень параметра y .

Приклад 2.11. Ніде не визначена функція $f(x) = -x - 1$.

Побудуємо рекурсивне рівняння:

$$f(x) = \mu_z(z + x + 1 = 0)$$

Результат операції мінімізації не визначений навіть для точки $x=0$.

Приклад 2.12. Функція, визначена в одній точці, $f(x) = -x$.

Побудуємо рекурсивне рівняння:

$$f(x) = \mu_z(z + x = 0).$$

Результат операції мінімізації визначений тільки для точки $x = 0$, при інших значеннях x обчислення (підбір потрібного значення z) ніколи не буде закінчено.

Значення операції мінімізації, застосоване до функції $f(x, y)$, можна записати як $\mu[f(x, y)]$.

2.2.4. Загально рекурсивні функції

Загально рекурсивна функція – частково рекурсивна функція, визначена для всіх значень аргументів. Задача визначення того, чи є частково рекурсивна функція з даним описом загально рекурсивною, алгоритмічно нерозв’язна.

Легко зрозуміти, що будь-яка примітивно рекурсивна функція є частково рекурсивною, тому що за визначенням оператори для побудови частково рекурсивних функцій містять оператори для побудови примітивно рекурсивних функцій.

Також зрозуміло, що примітивно рекурсивна функція визначена всюди і тому є загально рекурсивною функцією (у примітивно рекурсивної функції немає причини «зависати», тому що при її побудові використовуються оператори, що визначають всюди визначені функції).

Досить складно довести існування й привести приклад загально рекурсивної функції, що не є примітивно рекурсивною. Одним з популярних прикладів є функція Акермана.

$$A(m, n) = \begin{cases} n + 1, & m = 0; \\ A(m - 1, 1), & m > 0, n = 0; \\ A(m - 1, A(m, n - 1)), & m > 0, n > 0. \end{cases}$$

```
function akker(m,n:extended):extended;
```

```
begin
```

```
if m=0 then
```

```
  akker:=n+1;
```



```

if (m>0) and (n=0) then
akker:=akker(m-1,1)else
if (m>0) and (n>0) then
  akker:=akker(m-1,akker(m,n-1));
end;
int Akk(int m, int n)
{
  if (m=0)return (n+1);
  else if ((m>0) and (n=0))
  return Akk(m-1,1);
  return Akk(m-1,Akk(m,n-1));
}

```

Таблиця 2.1

Таблиця значень функції Акермана

M	0	1	2	3	4
N					
0	1	2	3	5	13
1	2	3	5	13	65533
2	3	4	7	29	$2^{65536} - 3$
3	4	5	9	61	$2^{2^{65536}} - 3$
4	5	6	11	125	$2^{2^{2^{65536}}} - 3$

Як було показано Геделем, частково рекурсивні функції збігаються з множиною обчислюваних функцій.

Примітивно рекурсивні функції в програмі

У термінах імперативного програмування – примітивно рекурсивні функції відповідають програмним блокам, у яких використовуються:

1) Арифметичні операції.

2) Умовний оператор.

3) Оператор детермінованого циклу.

Імперативне програмування – це парадигма програмування, яка описує процес обчислення у вигляді інструкцій, які змінюють стан програми.

Приклади імперативних мов: FORTRAN, BASIC, Pascal, Java, PHP.

Функціональна мова: Haskell. Логічна мова: Prolog.

Частково рекурсивні функції в програмі

Програмні блоки, що містять оператор ітераційного циклу, переходять до класу частково рекурсивних функцій.

Число ітерацій в ітераційному циклі заздалегідь не відомо і, у принципі, може бути нескінченним.

Частково рекурсивні функції для деяких значень аргументу можуть бути не визначені через те, що оператор мінімізації аргументу не завжди коректно визначений, тому що функція f може не дорівнювати нулю при жодних значеннях аргументів.

З позицій імперативного програмування результатом частково рекурсивної функції може бути не тільки число, але й Exception або нескінченний цикл, що відповідає невизначеному значенню.

Контрольні запитання

1. Яку функцію одержують з $g = 0$ і $h(x, y) = 2x$ за допомогою схеми примітивної рекурсії?

2. Показати, що $S(x, y) = x + y$ - примітивно рекурсивна функція.

3. Показати, що функція $f_{\text{exp}}(x, y) = x^y$ є примітивно рекурсивною функцією.

4. Яку функцію можна побудувати із g і h за допомогою схеми примітивної рекурсії, за умови, що $g(x) = 1$ й $h(x, y) = kx$, де $k = \text{const}$?

5. За допомогою операції мінімізації обчислити $f(5, 2)$, якщо $f(x, y) = x - y$.

2.3. Друга універсальна алгоритмічна модель.

Машина Тьюринга

2.3.1. Машина Тьюринга

Розглянемо другий тип алгоритмів. Актуальність їх пов'язана з розвитком обчислювальної техніки. Розглянуті алгоритми засновані на уявленні про алгоритм як про деяке детерміноване обладнання, здатне виконувати в кожний окремих дискретний момент часу примітивні операції. Таке представлення служить гарантією однозначності алгоритму й елементарності його кроків. Основною теоретичною моделлю цього типу є створена в 30-х роках концепція *машини Тьюринга*. Саме машина Тьюринга стала моделлю сучасної ЕОМ і сприяла розвитку сучасної обчислювальної техніки [15].

Область використання машини Тьюринга

Поняття *машини Тьюринга* виникає в результаті прямої спроби розкласти інтуїтивно відомі нам обчислювальні процедури на елементарні операції: Тьюринг привів ряд доводів на користь того, що повторення його елементарних операцій було б достатньо для проведення будь-якого можливого обчислення. Тому машина Тьюринга (МТ) використовується:

- 1) якщо потрібно довести *можливість* алгоритмічної реалізації обчислюваної функції;
- 2) якщо потрібно *оцінити обчислювальну складність* або *трудомісткість* розв'язування задачі за даним алгоритмом.

Для цього ми моделюємо роботу довільного алгоритму в термінах розглянутої задачі. Потім *визначається* клас машин-обчислювачів, які можуть розв'язати дану задачу – формально описуються правила роботи машини, вхідні дані, обмеження і т. ін. (оскільки у визначенні задачі нічого не говориться про програми у звичному для нас розумінні, то алгоритмічна

можливість розв'язання або нерозв'язність зводиться до проблеми зупинки довільного алгоритму розв'язування задачі). У якості машини-обчислювача виберемо машину Тьюринга, оскільки раніше було показано, що будь-яка обчислювана функція може бути реалізована на МТ, і зведемо розв'язування даної задачі до існуючих груп задач, для яких відомо, що вони розв'язуються на МТ.

Математична модель машини Тьюринга

Для опису алгоритму МТ зручно представляти деяке обладнання, що складається із чотирьох частин: стрічки, зчитувальної голівки, пристрою керування й внутрішньої пам'яті [16].

1. Стрічка передбачається потенційно нескінченною, розбитою на клітинки. За необхідності до першої або останньої клітинки, у якій перебувають символи, додається порожня клітинка. Машина працює в часі, який вважається дискретним, і його моменти занумеровано $1, 2, 3, \dots$. У кожний момент стрічка містить скінченне число клітинок. У клітинку в дискретний момент часу може бути записаний тільки один символ (буква) із зовнішнього алфавіту $A = \{a_0, a_1, \dots, a_m\}$. Символ a_0 є порожнім символом (також позначуваним λ). Інші символи a_1, \dots, a_m називають непустими. У цьому алфавіті A у вигляді слова (кінцевого впорядкованого набору символів) кодується та інформація, яка подається в МТ. Машина «переробляє» (перетворює) інформацію, подану у вигляді слова, у нове слово.

2. Зчитувальна голівка (деякий зчитувальний елемент) переміщається уздовж стрічки так, що в кожний момент часу вона оглядає рівно одну клітинку стрічки. Голівка може зчитувати вміст клітинки й записувати в неї новий символ з алфавіту A . В одному такті роботи вона може переміщуватися тільки на одну клітинку вправо (R), вліво (L) або залишатися на місці (E). Позначимо множину переміщень голівки

$D = \{R, L, E\}$. Якщо в момент часу t голівка перебуває в крайній клітинці й пересувається у відсутню клітинку, то додається нова порожня клітинка, над якою з'явиться голівка в момент $t + 1$.

3. Внутрішня пам'ять машини – це деяка скінченна множина внутрішніх станів $Q = \{q_0, q_1, \dots, q_n\}$. Два стани машини мають особливе значення: q_1 – початковий внутрішній стан (початкових внутрішніх станів може бути декілька), q_0 – заключний стан або стоп-стан (заклучний стан завжди один). У кожний момент часу МТ характеризується положенням голівки й внутрішнім станом. Наприклад, під клітинкою, над якою перебуває голівка, вказується внутрішній стан машини.

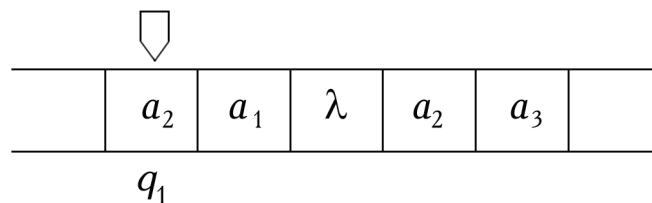


Рис. 2.1. Розташування елементів машини Тьюрінга

4. Пристрій керування в кожний момент часу t залежно від зчитуваного в цей момент символу на стрічці й внутрішнього стану машини виконує наступні дії:

- 1) змінює зчитуваний у момент t символ a_i на новий символ a_j (зокрема, залишає його без змін, тобто $a_i = a_j$);
- 2) пересуває голівку в одному з наступних напрямків: R, L, E ;
- 3) змінює наявний у момент t внутрішній стан машини q_i на новий q_j , у якому буде машина в момент часу $t + 1$ (може бути, що $q_i = q_j$).

Такі дії пристрою керування називають командою, яку можна записати у вигляді:

$$q_i a_i \rightarrow q_j a_j D$$

де q_i – внутрішній стан машини в цей момент; a_i – зчитуваний в цей момент символ; a_j – символ, на який заміняється символ a_i (може бути $a_i = a_j$); символ D – це один із символів: R , або L , або E , що вказує напрямок руху голівки; q_j – внутрішній стан машини в наступний момент (може бути $q_i = q_j$). Вирази $q_i a_i$ й $q_j a_j D$ називають лівою й правою частинами цієї команди відповідно. Число команд, у яких ліві частини попарно різні, є скінченним числом, оскільки множини $Q \setminus \{q_0\}$ й A скінченні.

Не існує команд з однаковими лівими частинами, тобто якщо програма МТ містить вираз

$$q_i a_i \rightarrow q_j a_j D \text{ і } q_t a_t \rightarrow q_k a_k D, \text{ то } q_i \neq q_t \text{ й } a_i \neq a_t, \text{ а також } D \in \{R, L, E\}.$$

Сукупність усіх команд називають програмою машини Тьюринга. Максимальне число команд у програмі дорівнює $(m+1) \times n$, де

$$\{q_i \mid 1 \leq i \leq n\} \text{ і } \{a_j \mid 0 \leq j \leq m\}$$

Вважають, що заключний стан команди q_0 може стояти тільки в правій частині команди, початковий стан q_1 може стояти як у лівій, так і в правій частині команди.

Виконання однієї команди називають кроком. Обчислення (або робота) машини Тьюринга є послідовністю кроків один за одним без пропусків, починаючи з першого.

Отже, МТ задана, якщо відомі чотири скінченних множини: зовнішній алфавіт A , внутрішній алфавіт Q , множина D переміщень голівки й програма машини, що є скінченною множиною команд.

Робота машини Тьюринга

Робота машини повністю визначається задаванням у перший (початковий) момент:

1) слова на стрічці, тобто послідовності символів, записаних у клітинках стрічки (слово будується з цих символів пересуванням по клітинках стрічки зліва направо);

2) положення голівки;

3) внутрішнього стану машини. Сукупність цих трьох умов (у цей момент) називають конфігурацією (у цей момент). Зазвичай у початковий момент внутрішнім станом машини є q_1 , а голівка перебуває або над першою ліворуч, або над першою праворуч клітинкою стрічки.

Задане слово на стрічці за умови, що початковий стан машини q_1 й голівка перебуває над першим словом, називають початковою конфігурацією. В іншому випадку говорять, що машина Тьюринга незастосовна до слова початкової конфігурації. Іншими словами, у початковий момент конфігурація може бути подана в наступному вигляді:

- на стрічці, що складається з деякого числа клітинок, у кожній клітинці записаний один із символів зовнішнього алфавіту A ,

- голівка перебуває над першою ліворуч або першою праворуч клітинкою стрічки,

- внутрішнім станом машини є q_1 .

Слово на стрічці, що вийшло в результаті реалізації цієї команди й положення голівки, називають *заключною конфігурацією*.

Наприклад, якщо в початковий момент на стрічці записано слово $a_1 a_2 \lambda a_1 a_1$, то початкова конфігурація буде мати вигляд (під клітинкою, над якою перебуває голівка, вказують внутрішній стан машини):

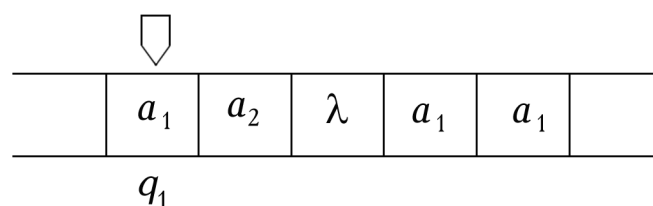


Рис. 2.2. Початкова конфігурація машини Тьюринга

Робота машини Тьюринга полягає в послідовному застосуванні команд, причому, застосування тієї або іншої команди визначається поточною конфігурацією. Так, у наведеному вище прикладі повинна застосуватися команда з лівою частиною q_1a_1 .

Отже, знаючи програму й задавши початкову конфігурацію, повністю визначаємо роботу машини над словом у початковій конфігурації. Якщо в роботі машини Тьюринга в деякий момент t виконується команда, права частина якої містить q_0 , то в такий момент робота машини вважається закінченою, і говорять, що машина застосовна до слова на стрічці в початковій конфігурації. Насправді, q_0 не зустрічається в лівій частині жодної команди – цим і пояснюється назва q_0 «заключний стан». Результатом роботи машини в такому випадку вважається слово, яке буде записано на стрічці в заклучній конфігурації, тобто в конфігурації, у якій внутрішній стан машини є q_0 . Якщо ж у роботі машини в жоден з моментів не реалізується команда із заклучним станом, то процес обчислення буде нескінченним. У цьому випадку говорять, що машина незастосовна до слова на стрічці в початковій конфігурації.

Приклади машин Тьюринга, що працюють в алфавіті $\{a, b\}$

Проілюструємо роботу машини Тьюринга на наступному прикладі.

Приклад 2.13. Побудувати машину Тьюринга T_1 , яка застосовна до всіх слів із зовнішнім алфавітом $\{a, b\}$ і робить наступне: будь-яке слово x_1, x_2, \dots, x_n , де $x_i = a$ або $x_i = b$, ($i = 1, 2, \dots, n$) перетворить у слово x_2, \dots, x_n, x_1 , тобто, починаючи працювати при слові x_1, x_2, \dots, x_n на стрічці у початковій конфігурації, машина зупиниться, і в заклучній конфігурації на деякій ділянці стрічки буде записане слово x_2, \dots, x_n, x_1 , а всі інші клітинки стрічки (якщо такі будуть) виявляться порожніми.

Розв'язок. За зовнішній алфавіт машини T_1 візьмемо множину $A = \{\lambda, a, b\}$, а за внутрішній – $Q = \{q_0, q_1, q_2, q_3\}$. Команди визначимо у такий спосіб:

$$q_1a \rightarrow \lambda R q_2, q_1b \rightarrow \lambda R q_3, q_i y \rightarrow y R q_i, \text{ де } y \in \{a, b\}, i = 2, 3; q_2\lambda \rightarrow a E q_0, \\ q_3\lambda \rightarrow b E q_0.$$

Розглянемо роботу машини T_1 над словом ba . У роботі машини над словом ba початкова конфігурація має такий вигляд:

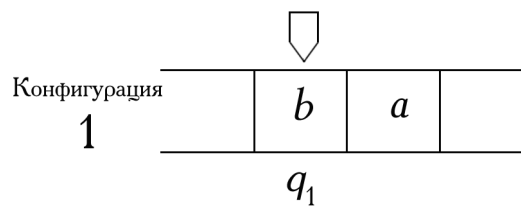


Рис. 2.3. Конфігурація 1

На першому кроці діє команда: $q_1b \rightarrow \lambda R q_3$.

У результаті створюється наступна конфігурація:

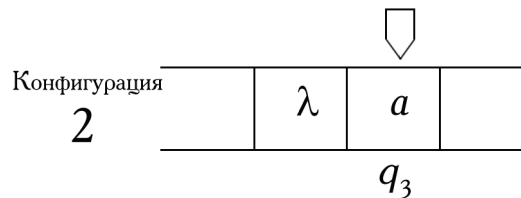


Рис. 2.4. Конфігурація 2

На другому кроці діє команда $q_3a \rightarrow a R q_3$

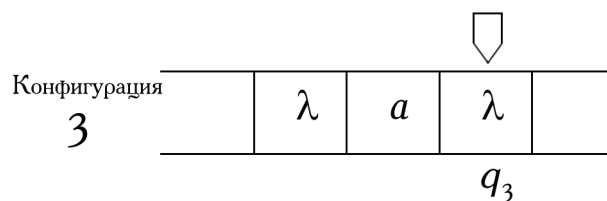


Рис. 2.5. Конфігурація 3

Нарешті, третій крок обумовлений командою $q_3\lambda \rightarrow bEq_0$.

У результаті чого створюється конфігурація:

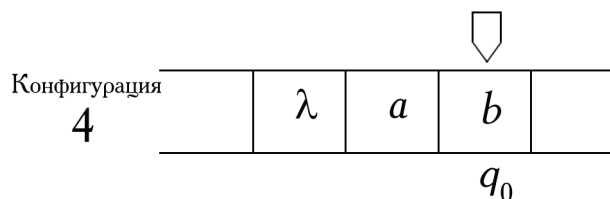


Рис. 2.6. Конфігурація 4

Ця конфігурація є заключною, тому що машина опинилася в стані зупинки q_0 . Таким чином, слово ba перетворено в слово ab .

Отриману послідовність конфігурацій можна записати у більш короткий спосіб.

Конфігурація 1 записується у вигляді наступного слова в алфавіті $a \cup Q$: q_1ba (вміст клітинки, що оглядається, записаний праворуч від стану, у якому перебуває в даний момент машина).

Конфігурація 2 записується так: q_3a .

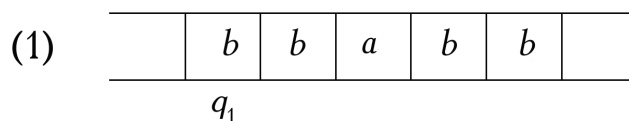
Конфігурація 3 – $aq_3\lambda$.

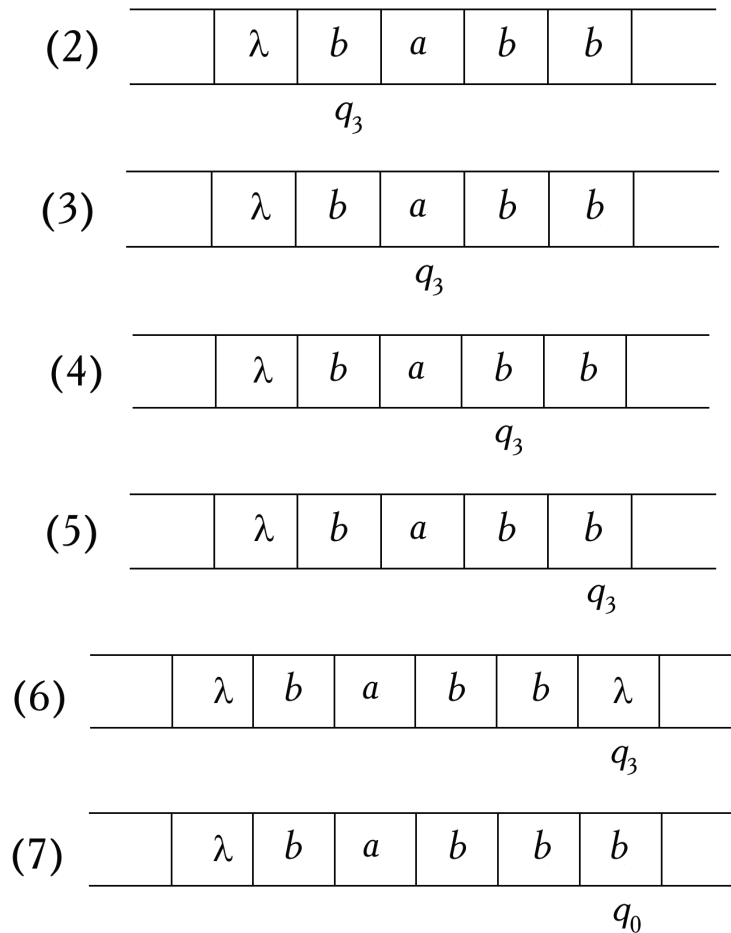
Конфігурація 4 – abq_0 .

Уся послідовність записується так: $q_1ba \Rightarrow \lambda q_3a \Rightarrow aq_3\lambda \Rightarrow abq_0$.

Приклад 2.14. Застосувати машину Тьюринга T_1 із прикладу 1 до слова $bbabb$, виходячи з початкового стану, при якому в стані q_1 оглядається крайня ліва клітинка, у якій міститься символ цього слова.

Розв'язок





Більш короткий запис цієї послідовності конфігурацій, тобто процесу роботи машини, має вигляд

$$\begin{aligned}
 q_1 b b a b b &\Rightarrow \lambda q_3 b a b b \Rightarrow \lambda b q_3 a b b \Rightarrow \lambda b a q_3 b b \Rightarrow \\
 &\Rightarrow \lambda b a b q_3 b \Rightarrow \lambda b a b b q_3 \lambda \Rightarrow b a b b b
 \end{aligned}$$

Таким чином, слово $bbabb$ перетворено машиною в слово $babbb$.

Приклад машини Тьюрінга, що працює в алфавіті $\{a, b, c\}$

Приклад 2.15

Задамо машину Тьюрінга

- зовнішнім алфавітом $A = \{a, b, c\}$,

- алфавітом внутрішніх станів $Q = \{q_0, q_1, q_2, q_3\}$,

- системою команд

$$q_1 a \rightarrow q_1 L a, q_2 a \rightarrow q_3 R b, q_3 a \rightarrow q_1 L a, q_1 b \rightarrow q_2 L a, q_2 b \rightarrow q_2 L b,$$

$$q_3b \rightarrow q_3Rb, q_1c \rightarrow q_0a, q_2c \rightarrow q_2Lc, q_2\lambda \rightarrow q_2Ea, q_3c \rightarrow q_3Rc.$$

Представлена система команд може бути записана у вигляді таблиці:

Таблиця 2.2

Система команд машини Тьюринга

	q_1	q_2	q_3
a	q_1La	q_3Rb	q_1La
b	q_2La	q_2Lb	q_3Rb
c	q_0a	q_2Lc	q_3Rc

Для того, щоб визначити по таблиці, що буде робити машина, перебуваючи, наприклад, у стані q_2 й оглядаючи в клітинці символ b , потрібно знайти в таблиці клітинку, що перебуває на перетині стовпця q_2 й рядка b . У цій клітинці записане q_2Lb . Це означає, що на наступному кроці машина залишиться в колишньому (попередньому) стані q_2 , збереже вміст b клітинки, що оглядається, і перейде до огляду наступної лівої клітинки на стрічці.

Припустимо, що в початковій конфігурації голівка перебуває над крайньою правою клітинкою. Застосуємо цю машину до слова $bbcbb$. Послідовність конфігурацій, що виникають у процесі роботи машини (початкова конфігурація – стандартна початкова):

Таблиця 2.3

Таблиця роботи машини Тьюринга

	1	2	3	4	5	6
Комбінація	$bbcq_1b$	$bbcq_2ba$	bbq_2cba	bq_2bcba	λq_2bbcba	$\lambda q_2\lambda bbcba$
Команда	q_2La	q_2Lb	q_2Lc	q_2Lb	q_2Lb	q_2Ea

	7	8	9	10	11	12
Комбінація	$\lambda q_2 abbcba$	$bq_3 bbcb a$	$bbq_3 bcba$	$bbbq_3 cba$	$bbbcq_3 ba$	$bbbcq_3 a$
Команда	$q_3 Rb$	$q_3 Rb$	$q_3 Rb$	$q_3 Rc$	$q_3 Rb$	$q_1 La$
	13	14	15	16	17	18
Комбінація	$bbbcq_1 ba$	$bbbq_2 caa$	$bbq_2 bcaa$	$bq_2 bbcaa$	$\lambda q_2 bbbcaa$	$\lambda q_2 \lambda bbbcaa$
Команда	$q_2 La$	$q_2 Lc$	$q_2 Lb$	$q_2 Lb$	$q_2 Lb$	$q_2 Ea$
	19	20	21	22	23	24
Комбінація	$\lambda q_2 abbbcaa$	$bq_3 bbbcaa$	$bbq_3 bbcaa$	$bbbq_3 bcaa$	$bbbbq_3 caa$	$bbbbcq_3 aa$
Команда	$q_3 Rb$	$q_3 Rb$	$q_3 Rb$	$q_3 Rb$	$q_3 Rc$	$q_1 La$
	25	26				
Комбінація	$bbbbq_1 caa$	$bbbbq_0 aaa$				
Команда	$q_0 a$					

Неважко помітити, що дана МТ реалізує операцію додавання: у результаті її роботи на стрічці записано підряд стільки букв b , скільки їх було всього записано по обидві сторони від букви c перед початком роботи машини.

З наведених прикладів випливає, що МТ – це деяке правило (алгоритм) для перетворення слів алфавіту $A \cup Q$, тобто конфігурацій. Таким чином, для визначення (побудови) машини Тьюринга потрібно задати її зовнішній і внутрішній алфавіти, програму й указати, які із символів позначають порожню клітинку й заключний стан.

2.3.2. Функції, обчислювані за Тьюрингом

Опис класу функцій

З розвитком науки й практики з'явилися задачі, для яких не були знайдені методи їх розв'язування. Виникло запитання: відсутність другого типу алгоритму для розв'язування задач є результатом недостатнього знання про задачі цього класу чи алгоритму розв'язування для даного класу не існує?

Для розв'язання цієї проблеми, за аналогією із уже розглянутими нами задачами, що описуються алгоритмами першого типу, введено поняття обчислюваної функції й для даного класу задач.

Будемо розглядати функції f від одного або декількох аргументів, які задані на множині $N_0 = \{0, 1, 2, 3, \dots, n, \dots\}$ всіх невід'ємних цілих чисел або на деяких його підмножинах (часткові функції), і набувають значень у множині N . Область визначення Df функції f – це підмножина множини $N_0^n = N_0 \times N_0 \times \dots \times N_0$:

$$D_f = \left\{ (x_1, \dots, x_n) \in N_0^n \mid f(x_1, \dots, x_n) \in N - \text{визначене} \right\}$$

Значення функції $f(x_1, \dots, x_n)$ на наборі $(m_1, \dots, m_n) \in N_0^n$ визначене, якщо $f(m_1, \dots, m_n) = k$, де $k \in N_0$, а якщо ні, то функція f вважається не визначеною на заданому наборі.

Під *числовими функціями* будемо розуміти функції, значеннями яких і значеннями їх аргументів є невід'ємні цілі числа.

При обчисленні числових функцій на машинах Тьюринга часто користуються спеціальним кодуванням чисел. Наприклад, натуральне число m будемо задавати набором з $m + 1$ одиниць, який будемо позначати через 1^{m+1} .

У такий спосіб:

Число 0 будемо позначати як 1 (однією одиницею);

Число 1 будемо позначати як 11 (двома одиницями);

Число 2 будемо позначати як 111 (трьома одиницями) і т. д.

Числову функцію $f(x_1, \dots, x_n)$ називають обчислюваною за

Тьюрингом, якщо існує машина Тьюринга T_f , що задовольняє наступним двом умовам:

1) для будь-якого набору $(m_1, \dots, m_n) \in D_f \subset N^n$ такого, що $f(m_1, \dots, m_n) = k, k \in N$, машина Тьюринга застосовна до слова

$$1^{m_1+1} \lambda 1^{m_2+1} \lambda \dots \lambda 1^{m_n+1} \quad (2.3)$$

і в заключній конфігурації на деякій ділянці стрічки буде записане слово 1^{k+1} , а інші ділянки стрічки, якщо такі будуть, виявляться порожніми.

2) якщо $(m_1, \dots, m_n) \notin D_f$, машина Тьюринга T_f не застосовна до слова (2.3).

Якщо функція f обчислювана за Тьюрингом за допомогою машини T_f , то будемо говорити, що машина T_f обчислює функцію f .

Приклади функцій, обчислюваних за Тьюрингом

Приклад 2.16. Побудувати машину Тьюринга T_3 із зовнішнім алфавітом $\{\lambda, 1\}$, яка обчислює функцію $f(x) = x + 1$.

Розв'язок. Будемо припускати, що перед початком роботи на стрічці машини записані початкові (вхідні) значення аргументу й зчитувальна голівка оглядає перший зліва значущий символ (рис. 2.7). Крім того, після виконання обчислень зчитувальна голівка зупиняється в заключній конфігурації.

Перш ніж почати написання програми роботи машини Тьюринга, слід визначити порядок її роботи для одержання результату. У нашому випадку

після закінчення роботи машини на стрічці повинно бути зайнято на одну клітинку більше, ніж на ній зайнято клітинок перед початком роботи.

Команди цієї машини можуть бути визначені в такий спосіб:

$$q_1 1 \rightarrow 1Rq_1, q_1 \lambda \rightarrow 1Eq_0$$

Робота машини T_3 при обчисленні $f(1)$ складається з конфігурацій:

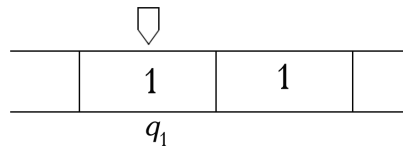


Рис. 2.7

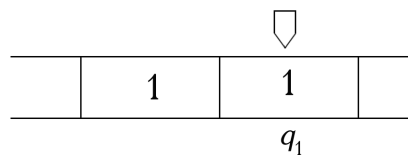


Рис. 2.8

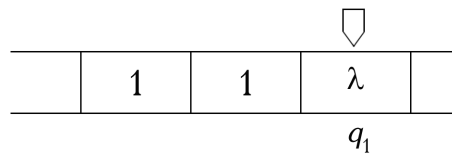


Рис. 2.9

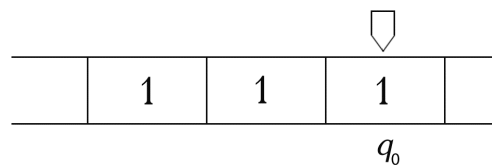


Рис. 2.10

Очевидно, що найкращий спосіб виконати вимогу збільшення кількості клітинок полягає в наступному. Після початку роботи зчитувальна голівка машини повинна переміститися на одну клітинку вліво, записати в порожню клітинку 1 і зупинитися.

Перша команда має вигляд: $q_1 1 \rightarrow 1Rq_1$ (рис. 2.8).

Згідно із цією командою машина, перебуваючи в стані q_1 , читає символ 1, записаний у клітинці, що оглядається, залишає цей символ

у клітинці, залишається в попередньому стані q_1 й пересувається на одну клітинку стрічки вправо. Друга команда має вигляд: $q_1\lambda \rightarrow 1Eq_0$ (рис. 2.9).

Таким чином, згідно із цією командою, машина, перебуваючи в стані q_1 , читає порожній символ, записаний у клітинці, що оглядається, записує на його місце символ 1, переходить у стан q_0 і зупиняється (рис. 2.10).

Програма роботи машини Тьюринга також може бути записана у вигляді таблиці. Стовпцям таблиці відповідають символи зовнішнього алфавіту, а рядкам – стан машини.

Таблиця 2.4

Система команд машини T_3

	λ	1
q_1	$1Eq_0$	$1Rq_1$

Приклад 2.17. Побудувати машину T_4 , що обчислює числову функцію $f(x, y) = x + y$.

Розв'язок. Нехай зовнішнім алфавітом даної машини є алфавіт $A = \{\lambda, 1\}$. Внутрішній алфавіт $Q = \{q_0, q_1, q_3, q_4\}$. Роботу машини забезпечує набір команд:

Таблиця 2.5

Система команд машини T_4

	λ	1
q_1	$1Eq_0$	$1Rq_1$
q_2	λLq_3	$1Rq_2$
q_3	λEq_3	λLq_4
q_4	λEq_4	λLq_0

Слід зазначити, що для даної машини T_4 виписані всі команди, що здійснюють обчислення функції $f(x, y) = x + y$.

Нехай задане початкове слово $11\lambda 11$, що еквівалентно $f(x, y) = 1 + 1$

$$\begin{aligned} q_1 11\lambda 11 &\Rightarrow 1q_1 1\lambda 11 \Rightarrow 11q_1 \lambda 11 \Rightarrow 111q_2 11 \Rightarrow \\ &\Rightarrow 1111q_2 1 \Rightarrow 11111q_2 \lambda \Rightarrow 1111q_3 1\lambda \Rightarrow 111q_4 1\lambda\lambda \Rightarrow \\ &\Rightarrow 111q_0 \lambda\lambda\lambda \end{aligned}$$

Зауваження. Усі арифметичні операції є обчислюваними за Тьюрингом функціями.

Контрольні запитання

1. Побудувати машину Тьюринга, що обчислює нуль-функцію $\mathbf{0}(x) = 0$ у двійковій системі числення та записати програму у вигляді послідовності конфігурацій для слова 1010.
2. Побудувати машину Тьюринга, що обчислює функцію проектування $I_2^2(x_1, x_2)$. Записати програму у вигляді послідовності конфігурацій при $x_1 = 110$ і $x_2 = 101$.
3. Побудувати машину Тьюринга, що обчислює числову функцію $f(x, y) = x + y$. Початкове слово на стрічці $11\lambda 111$.
4. Нехай задана машина Тьюринга зовнішнім алфавітом $A = \{0, 1, 2, 3\}$ і алфавітом внутрішніх станів $Q = \{q_0, q_1, q_2\}$. Система команд включає такі команди:
 $q_1\lambda \rightarrow \lambda E q_0$, $q_1 2 \rightarrow 1 R q_1$, $q_1 6 \rightarrow 8 R q_1$, $q_1 1 \rightarrow 9 R q_1$, $q_1 0 \rightarrow 9 R q_1$
 Записати програму машини Тьюринга, якщо її початкова конфігурація має вигляд: $q_1 2016$.
5. Побудувати машину Тьюринга, що обчислює числову функцію $f(x, y) = x - y$. Початкове слово на стрічці $111\lambda 011$.

2.4. Третя універсальна алгоритмічна модель.

Нормальні алгоритми Маркова

2.4.1. Виникнення теорії нормальних алгоритмів

Теорія нормальних алгоритмів (або алгорифмів, як називав їх творець теорії) була розроблена математиком А.А. Марковим (1903–1979) наприкінці 1940-х – початку 1950-х рр. ХХ ст. Ці алгоритми є набором деяких правил з перетворення слів у певному алфавіті, тобто вхідні (початкові) дані й шукані результати для алгоритмів є словами в певному алфавіті [17].

2.4.2. Підстановки Маркова

Алфавітом називають будь-яку непусту множину. Елементи алфавіту називають *буквами*, а будь-які послідовності букв – *словами* в даному алфавіті. Для зручності міркувань допускаються порожні слова (вони не мають у своєму складі жодної букви). *Порожнє слово* будемо позначати Λ . Якщо A й B – два алфавіти, причому $A \subseteq B$, то алфавіт B називають *розширенням* алфавіту A .

Слова будемо позначати латинськими буквами: P, Q, R (або цими ж буквами з індексами). Одне слово може бути складовою частиною іншого слова. Тоді перше називають *підсловом* другого або *входженням* у друге. Наприклад, якщо A – український алфавіт, то можемо розглянути такі слова: P_1 = параграф, P_2 = граф, P_3 = ра. Слово P_2 є підсловом слова P_1 , а P_3 – підсловом P_1 і P_2 , причому у P_1 воно входить двічі. Перше входження є найбільш вагомим (викликає найбільший інтерес).

Визначення. Підстановкою Маркова називають операцію над словами, яку задають за допомогою впорядкованої пари слів (P, Q) , що полягає в наступному. У заданому слові R знаходять перше входження слова

P (якщо таке є) і, не змінюючи інших частин слова R , заміняють у ньому це входження словом Q . Отримане слово називають *результатом* застосування підстановки Маркова (P, Q) до слова R . Якщо ж входження P в слово R немає, то вважають, що підстановка Маркова (P, Q) незастосовна до слова R .

Окремими випадками підстановок Маркова є підстановки з порожніми словами: (Λ, Q) , (P, Λ) , (Λ, Λ) .

Для позначення підстановки Маркова (P, Q) використовують запис $P \rightarrow Q$. Цей запис називають *формулою підстановки* (P, Q) . Деякі підстановки (P, Q) будемо називати *заклучними* (зміст назви стане зрозумілим дещо пізніше). Для позначення таких підстановок будемо використовувати запис $P \rightarrow .Q$, називаючи його *формулою заклучної підстановки*. Слово P називають *лівою частиною*, а Q – *правою частиною* у формулі підстановки.

2.4.3. Нормальні алгоритми та їх застосування до слів

Упорядкований скінченний список формул підстановок

$$\left\{ \begin{array}{l} P_1 \rightarrow Q_1 \\ P_2 \rightarrow Q_2 \\ \dots\dots\dots \\ P_r \rightarrow Q_r \end{array} \right.$$

в алфавіті A називають *схемою* (або *записом*) нормального алгоритму в A . Дана схема визначає (детермінує) алгоритм перетворення слів, названий нормальним алгоритмом Маркова. Дамо його точне визначення.

Визначення. Нормальним алгоритмом (Маркова) в алфавіті A називають наступне правило побудови послідовності V_i слів у алфавіті A , виходячи з початкового слова V у цьому алфавіті.

Нехай $V = V_0$, побудовано слово V_i для певного $i \geq 0$ і процес побудови розглянутої послідовності ще не завершився. Якщо при цьому в схемі нормального алгоритму немає формул, ліві частини яких входили б у V_i , то V_{i+1} вважають рівним V_i , і процес побудови послідовності вважають таким, що завершився.

Якщо ж у схемі є формули з лівими частинами, що входять у V_i , то як V_{i+1} беруть результат підстановки Маркова правої частини першої з таких формул замість першого входження її лівої частини в слово V_i . Процес побудови послідовності вважають таким, що *завершився*, якщо на даному кроці була застосована формула заключної підстановки, а якщо ні, то таким, що *триває*.

Якщо процес побудови згаданої послідовності обривається, то говорять, що розглянутий *нормальний алгоритм застосовний до слова V_0* . Останній член W послідовності називають *результатом застосування нормального алгоритму до слова V_0* . Говорять, що *нормальний алгоритм перетворює V_0 у W* .

Послідовність V_i будемо записувати в такий спосіб:

$$\begin{aligned} V_0 &\Rightarrow V_1 \\ V_1 &\Rightarrow V_2 \\ &\dots \\ V_{m-1} &\Rightarrow V_m \end{aligned}$$

де $V_0 = V$ й $V_m = W$

Ми визначили поняття нормального алгоритму в алфавіті A . Якщо ж алгоритм заданий у деякому розширенні алфавіту A , то говорять, що він є *нормальним алгоритмом над A* .

Розглянемо приклади нормальних алгоритмів.

Приклад 2.18. Нехай $A = \{a, b\}$ – алфавіт. Розглянемо наступну схему нормального алгоритму в A :

$$\begin{cases} a \rightarrow \Lambda \\ bb \rightarrow \Lambda \end{cases}$$

Неважко зрозуміти, як працює визначений цією схемою нормальний алгоритм. Кожне слово V у алфавіті A , що містить хоча б одне входження букви a , алгоритм перетворює в слово, отримане із V викреслюванням у ньому найлівішого (першого) входження букви a . Порожнє слово він перетворює в порожнє.

Виберемо початкове слово $aabab$ й застосуємо до нього задану схему:

$$aabab \Rightarrow abab$$

$$abab \Rightarrow bab$$

$$bab \Rightarrow bb$$

$$bb \Rightarrow \Lambda$$

Приклад 2.19. Нормальний алгоритм у алфавіті $A = \{a, b, 1\}$ заданий схемою:

$$\begin{cases} a \rightarrow 1 \\ b \rightarrow 1 \end{cases}$$

Застосуємо його до слова $abaabbb$.

$$abaabbb \Rightarrow 1baabbb$$

$$1baabbb \Rightarrow 1b1abbb$$

$$1b1abbb \Rightarrow 1b11bbb$$

$$1b11bbb \Rightarrow 1111bbb$$

$$1111bbb \Rightarrow 11111bb$$

$$11111bb \Rightarrow 111111b$$

$$111111b \Rightarrow 1111111$$

До отриманого слова 1111111 жодна з підстановок даної схеми вже не застосовна. Отже, робота алгоритму завершена.

Приклад 2.20. Нормальний алгоритм у алфавіті $A = \{a, b\}$ заданий схемою:

$$\begin{cases} ab \rightarrow a, \\ b \rightarrow \Lambda, \\ a \rightarrow b \end{cases}$$

Проаналізуємо роботу алгоритму на слові $abbbaaab$.

$$abbbaaab \Rightarrow abbaaab$$

$$abbaaab \Rightarrow abaaab$$

$$abaaab \Rightarrow aaaaab$$

$$aaaaab \Rightarrow aaaa$$

$$aaaa \Rightarrow baaa$$

$$baaa \Rightarrow aaa$$

$$aaa \Rightarrow baa$$

$$baa \Rightarrow aa$$

$$aa \Rightarrow ba$$

$$ba \Rightarrow a$$

$$a \Rightarrow b$$

$$b \Rightarrow \Lambda.$$

Використовуючи як V довільні слова, складені із символів алфавіту A , легко дійти висновку, що даний алгоритм довільне слово переводить у порожній рядок.

2.4.4. Нормально обчислювані функції

Як і машини Тьюринга, нормальні алгоритми не виконують обчислень: вони лише здійснюють перетворення слів, замінюючи в них одні букви іншими за запропонованими правилами. У свою чергу, ми пропонуємо їм такі правила, результати застосування яких ми можемо інтерпретувати як обчислення. Розглянемо два приклади, у яких подібно до

машини Тьюринга числа представлені словами з відповідною кількістю одиниць.

Приклад 2.21. У алфавіті $A = \{1\}$ схема $\Lambda \rightarrow .1$ визначає нормальний алгоритм, який до кожного слова в алфавіті $A = \{1\}$ (усі такі слова є наступними: $\Lambda, 1, 11, 111, 1111, 11111$ і т. д.) приписує ліворуч 1. Отже, алгоритм реалізує (обчислює) функцію $f(x) = x + 1$.

Зауваження. У схемі $\Lambda \rightarrow .1$ крапка перед одиницею означає, що дана заміна виконується завершально.

Приклад 2.22. Дана функція

$$\varphi(111\dots 1) = \begin{cases} 1, & \text{якщо } n \text{ ділиться на } 3, \\ \Lambda, & \text{якщо } n \text{ не ділиться на } 3, \end{cases}$$

де n – число одиниць у слові $111\dots 1$.

Розглянемо нормальний алгоритм в алфавіті $A = \{1\}$ з наступною схемою:

$$\left\{ \begin{array}{l} 111 \rightarrow \Lambda, \\ 11 \rightarrow .\Lambda, \\ 1 \rightarrow .\Lambda, \\ \Lambda \rightarrow .11. \end{array} \right.$$

Цей алгоритм працює за таким принципом: поки число одиниць у слові не менше 3, алгоритм послідовно стирає по три одиниці. Якщо число одиниць менше 3, але більше 0, то одиниці, що залишилися, 1 або 11 стираються завершально; якщо слово порожнє, тобто число одиниць у початковому слові було кратне трьом, то воно завершально переводиться в слово 1.

Наприклад: $1111111 \Rightarrow 1111$

$1111 \Rightarrow 1$

$1 \Rightarrow \Lambda$

Початкове число 7 не ділиться на 3. Отже, результатом роботи алгоритму є Λ .

Наступний приклад виконує перевірку ділення на 3 числа 9.

$$111111111 \Rightarrow 111111$$

$$111111 \Rightarrow 111$$

$$111 \Rightarrow \Lambda$$

$$\Lambda \Rightarrow 1$$

Число 9 ділиться на 3 без остачі. Отже, слово 1 є результатом роботи алгоритму.

Визначення. Функцію f , задану на деякій множині слів алфавіту A , називають *нормально обчислюваною*, якщо існує таке розширення B даного алфавіту ($A \subseteq B$) й такий нормальний алгоритм у B , що кожне слово V (у алфавіті A) з області визначення функції f цей алгоритм перетворює у слово $f(V)$.

Таким чином, нормальні алгоритми прикладів 2.21 та 2.22 показують, що функції $f(x) = x + 1$ й $\varphi(x)$ нормально обчислювані. Причому відповідні нормальні алгоритми вдалося побудувати в тому ж самому алфавіті A , на словах якого були задані функції, що розглядалися, тобто розширювати алфавіт було не потрібно ($B = A$). Наступний приклад демонструє нормальний алгоритм, який обчислює дану функцію у розширеному алфавіті.

Приклад 2.23. Побудуємо нормальний алгоритм для обчислення функції $f(x) = x + 1$. Представимо алфавіт цифрами $A = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, а нормальний алгоритм побудуємо з використанням розширення $B = A \cup \{a, b\}$.

Схема даного алгоритму має вигляд (читати по стовпцях):

$0b \rightarrow .1$	$a0 \rightarrow 0a$	$1a \rightarrow 1b$
$1b \rightarrow .2$	$a1 \rightarrow 1a$	$2a \rightarrow 2b$
$2b \rightarrow .3$	$a2 \rightarrow 2a$	$3a \rightarrow 3b$
$3b \rightarrow .4$	$a3 \rightarrow 3a$	$4a \rightarrow 4b$
$4b \rightarrow .5$	$a4 \rightarrow 4a$	$5a \rightarrow 5b$
$5b \rightarrow .6$	$a5 \rightarrow 5a$	$6a \rightarrow 6b$
$6b \rightarrow .7$	$a6 \rightarrow 6a$	$7a \rightarrow 7b$
$7b \rightarrow .8$	$a7 \rightarrow 7a$	$8a \rightarrow 8b$
$8b \rightarrow .9$	$a8 \rightarrow 8a$	$9a \rightarrow 9b$
$9b \rightarrow b0$	$a9 \rightarrow 9a$	$\Lambda \rightarrow a.$
$b \rightarrow .1$	$0a \rightarrow 0b$	

Спробуємо застосувати алгоритм до порожнього слова Λ . Неважко зрозуміти, що на кожному кроці має застосовуватися найостанніша формула даної схеми. Отже, отримуємо нескінченний процес:

$$\begin{aligned} \Lambda &\Rightarrow a \\ a &\Rightarrow aa \\ aa &\Rightarrow aaa \\ &\dots \end{aligned}$$

Це означає, що до порожнього слова даний алгоритм незастосовний.

Якщо застосувати тепер алгоритм до слова 499, одержимо наступну послідовність слів:

$$\begin{aligned} \Lambda 499 &\Rightarrow a499 \\ \mathbf{a}499 &\Rightarrow 4a99 \\ \mathbf{4a}99 &\Rightarrow 49a9 \\ \mathbf{49a}9 &\Rightarrow 499a \\ \mathbf{499a} &\Rightarrow 499b \\ \mathbf{499b} &\Rightarrow 49b0 \\ \mathbf{49b0} &\Rightarrow 4b00 \\ \mathbf{4b00} &\Rightarrow 500 \end{aligned}$$

У розглянутому прикладі нормальний алгоритм побудований у алфавіті B , що є істотним розширенням алфавіту A (тобто $A \subseteq B$

й $A \neq B$), але даний алгоритм слова в алфавіті A перетворює знову в слова в алфавіті A . У такому випадку говорять, що алгоритм заданий *над* алфавітом A .

Приклад 2.24. У алфавіті $B = A \cup \{a, b\}$, що є розширенням алфавіту A , розглянемо нормальний алгоритм, заданий схемою (читати по стовпцях):

$$\begin{array}{lll}
 0b \rightarrow b9 & a0 \rightarrow 0a & 0a \rightarrow 0b \\
 1b \rightarrow .0 & a1 \rightarrow 1a & 1a \rightarrow 1b \\
 2b \rightarrow .1 & a2 \rightarrow 2a & 2a \rightarrow 2b \\
 3b \rightarrow .2 & a3 \rightarrow 3a & 3a \rightarrow 3b \\
 4b \rightarrow .3 & a4 \rightarrow 4a & 4a \rightarrow 4b \\
 5b \rightarrow .4 & a5 \rightarrow 5a & 5a \rightarrow 5b \\
 6b \rightarrow .5 & a6 \rightarrow 6a & 6a \rightarrow 6b \\
 7b \rightarrow .6 & a7 \rightarrow 7a & 7a \rightarrow 7b \\
 8b \rightarrow .7 & a8 \rightarrow 8a & 8a \rightarrow 8b \\
 9b \rightarrow .8 & a9 \rightarrow 0a & 9a \rightarrow 9b \\
 & & \Lambda \rightarrow a.
 \end{array}$$

Дана схема дозволяє реалізувати нормальний алгоритм обчислення функції $f(x) = x - 1$.

Перевіримо роботу алгоритму, застосувавши його до слова 3000.

Розв'язок. Алгоритм працює в такий спосіб. На першому кроці застосовується остання підстановка:

$$\Lambda 3000 \Rightarrow a3000.$$

Далі починають працювати підстановки із другого стовпця, міняючи місцями букву a із сусідніми праворуч цифрами даного числа доти, поки a не займе крайнє праве положення:

$$a3000 \Rightarrow 30a00$$

$$30a00 \Rightarrow 300a0$$

$$300a0 \Rightarrow 3000a$$

Тепер спрацьовує одна з підстановок третього стовпця (у цьому випадку перша), фактично міняючи букву a на букву b :

$$3000a \Rightarrow 3000b.$$

Нарешті, спрацьовує одна з підстановок першого стовпця. Якби слово, що перетворюється, завершилося цифрою, відмінною від 0, то спрацювала б одна із заключних підстановок, остання цифра була б замінена цифрою, на одиницю меншою, а алгоритм закінчив би свою роботу. У цьому випадку слово завершується цифрою 0. Тому спрацьовує перша підстановка першого стовпця:

$$3000b \Rightarrow 300b9.$$

Оскільки й передостання цифра в слові, що перетворюється, є 0, то алгоритм знову застосовує першу підстановку, міняючи місцями b і 0 й заміняючи 0 на 9. Так буде відбуватися доти, поки ліворуч від букви b не з'явиться цифра, відмінна від 0:

$$300b9 \Rightarrow 30b99$$

$$30b99 \Rightarrow 3b999.$$

Тепер спрацьовує одна із заключних підстановок:

$$3b999 \Rightarrow 2999$$

2.4.5. Принцип нормалізації Маркова

Творець теорії нормальних алгоритмів А.А. Марков висунув гіпотезу, що отримала назву «Принцип нормалізації Маркова». Згідно з цим принципом, для знаходження значень функції, заданої в деякому алфавіті, тоді й тільки тоді існує який-небудь алгоритм, коли функція нормально обчислювана.

2.4.6. Співпадіння класу всіх нормально обчислюваних функцій з класом усіх функцій, обчислюваних за Тьюрингом

Поняття нормально обчислюваної функції рівносильне поняттю функції, обчислюваної за Тьюрингом, а разом з ним і поняттю частково рекурсивної функції.

Теорема 2.4. *Будь-яка функція, обчислювана за Тьюрингом, буде також і нормально обчислюваною. Доведена також зворотна теорема.*

Теорема 2.5. *Будь-яка нормально обчислювана функція обчислювана за Тьюрингом.*

2.4.7. Еквівалентність різних теорій алгоритмів

Отже, ми ознайомилися із трьома теоріями, кожна з яких уточнює поняття алгоритму, і з'ясували, що всі ці теорії рівносильні між собою. Інакше кажучи, вони описують один і той самий клас функцій, тобто справедливою є наступна теорема.

Теорема 2.6. *Наступні класи функцій (що задані на натуральних числах та набувають натуральних значень) співпадають:*

- а) клас усіх функцій, обчислюваних за Тьюрингом;
- б) клас усіх частково рекурсивних функцій;
- в) клас усіх нормально обчислюваних функцій.

Відзначимо, що існують ще й інші, менш відомі теорії алгоритмів, і для всіх цих теорій також доведена їх еквівалентність із розглянутими теоріями.

Контрольні запитання

1. Побудувати нормальний алгоритм Маркова в алфавіті $A = \{1, 0\}$, який реалізує нуль-функцію: $0(x) = 0$ у двійковій системі числення.

Розглянути роботу алгоритму для числа 101101

2. Побудувати нормальний алгоритм Маркова в алфавіті $A = \{1\}$, який реалізує функцію слідування $f(x) = x + 2$.

3. Побудувати нормальний алгоритм Маркова в алфавіті $A = \{1\}$, який реалізує функцію проектування $I_3^3(x_1, x_2, x_3) = x_3$ на прикладі $I_3^3(2, 3, 1) = 1$

4. Нормальний алгоритм Маркова в алфавіті $A = \{a, b, 0, 1\}$ заданий схемою:

$$\left\{ \begin{array}{l} a \rightarrow 1, \\ b \rightarrow 0, \\ 10 \rightarrow b \end{array} \right. \quad \text{Застосувати дану схему до початкового слова: } ababa$$

5. Нормальний алгоритм Маркова в алфавіті $A = \{a, b, c\}$ заданий схемою:

$$\left\{ \begin{array}{l} a \rightarrow c, \\ bc \rightarrow b, \\ cb \rightarrow c \end{array} \right. \quad \text{Застосувати дану схему до початкового слова: } ababa$$

3. МЕТОДИ ОБЧИСЛЕНЬ

3.1. Інтерполяція функцій

3.1.1. Інтерполяція й задача інтерполяції

У загальному випадку під інтерполяцією розуміють наближене або точне знаходження якої-небудь величини по відомих окремих значеннях цієї ж або інших величин, пов'язаних з нею.

У практичних обчисленнях часто зустрічаються функції, значення яких задані лише в декількох точках відрізка, а саме, нехай функція $f(x)$ задана своїми значеннями або, говорять, таблицею своїх значень для деякої скінченної множини значень x_i аргументу x :

$$y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n).$$

У загальному вигляді записують $y_i = f(x_i)$, $i = 1, 2, 3, \dots, n$.

Нехай потрібно використовувати значення цієї функції $f(x)$ для значень x , відмінних від заданих x_j . Із цією метою будують функцію $g(x)$, що співпадає з $f(x)$ у заданих точках x_i , і застосовують її замість $f(x)$ для значень x , відмінних від заданих x_i . Такий спосіб визначення значень функції називають *інтерполяцією* [18].

Задачею інтерполяції називають спосіб побудови або знаходження такої функції $g(x)$, за допомогою якої можна з тим або іншим ступенем точності проводити обчислення замість заданої функції $f(x)$, або, говорять, заповнювати значення функції $f(x)$ [19,20]. Схематично задача інтерполяції може бути представлена у вигляді:

$$f(x) \rightarrow \{(x_i, y_i)\}_{i=1}^n \rightarrow g(x)$$

Інтерполяцію функцій використовують у наступних випадках:

- заміна функції, яку складно обчислити, іншою, яку обчислити легко;
- наближене відновлення функції на всій області задавання за значеннями її в окремих точках або по інших відомих величинах;
- одержання згладжуючих функцій;
- наближене знаходження граничних значень функцій;
- у задачах прискорення збіжності послідовностей і рядів та ін.

3.1.2. Постановка задачі інтерполяції

Нехай на деякому відрізку $[a, b]$ задані $n + 1$ різних точок $x_0, x_1, x_2, \dots, x_n$, $x_k \neq x_j$ при $0 \leq k \leq n$, $0 \leq j \leq n$ і значення деякої функції $f(x)$ у цих точках

$$f(x_0) = y_0, f(x_1) = y_1, f(x_2) = y_2, \dots, f(x_n) = y_n, \text{ або} \\ f(x_i) = y_i, \quad i = 0, 1, \dots, n \quad (3.1)$$

Завдання полягає в побудові функції $g(x)$ такої, щоб у точках $x_0, x_1, x_2, \dots, x_n$ вона отримувала ті ж значення, що й початкова $f(x)$, тобто $g(x_0) = y_0, g(x_1) = y_1, g(x_2) = y_2, \dots, g(x_n) = y_n$ або

$$g(x_i) = y_i, \quad i = 1, 2, 3, \dots, n \quad (3.2)$$

Умову (3.2) називають *умовою інтерполяції*.

Точки $x_0, x_1, x_2, \dots, x_n$ називають *вузлами інтерполяції*, а функцію $g(x)$ – *інтерполяційною* або *інтерполянтом*. Вузли можуть бути рівновіддаленими, тобто відстань між вузлами однакова

$$x_1 - x_0, \dots, x_n - x_{n-1} = h = \text{const},$$

тоді $x_{i+1} = x_i + h = x_0 + (i + 1)h$, $i = 0, 1, \dots, n - 1$.

Вузли також можуть бути довільно розташованими:

$$x_1 - x_0 \neq x_2 - x_1 \neq \dots \neq x_n - x_{n-1}$$

Геометрично розв'язування задачі означає, що потрібно знайти криву $y = g(x)$ деякого певного типу, що проходить через задану систему точок (x_i, y_i) , $i = 0, 1, \dots, n$. Криву називають *інтерполяційною кривою* (Рис. 3.1).

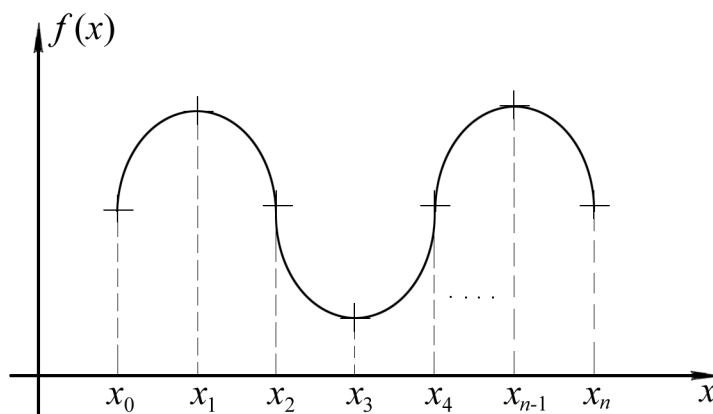


Рис. 3.1. Інтерполяційна крива

У такій загальній постановці задача може не мати розв'язків або мати нескінченну множину розв'язків.

3.1.3. Узагальнені многочлени

Для однозначної можливості розв'язання задачі інтерполяції (3.1)–(3.2) замість довільної функції $g(x)$ використовують поліноміальні функції, або узагальнені многочлени [21]:

$$F(x) = \sum_{i=0}^n c_i \omega_i(x) = c_1 \omega_1(x) + c_2 \omega_2(x) + \dots + c_n \omega_n(x),$$

де $\{\omega_i(x)\}$ – скінченна лінійно незалежна на $[a, b]$ система функцій.

При практичних обчисленнях найчастіше як $\{\omega_i(x)\}$ використовується послідовність:

$$1, x, x^2, \dots, x^n.$$

або послідовність тригонометричних функцій:

1, $\sin x$, $\cos x$, $\sin 2x$, $\cos 2x, \dots$,

або послідовність показникових функцій a^x .

Для коефіцієнтів узагальненого многочлена, використовуючи умови (3.2), отримаємо систему лінійних рівнянь:

$$\begin{cases} c_0 + c_1\omega(x_0) + c_2\omega(x_0) + \dots + c_m\omega(x_0) = f_0, \\ c_0 + c_1\omega(x_1) + c_2\omega(x_1) + \dots + c_m\omega(x_1) = f_1, \\ \dots \\ c_0 + c_1\omega(x_n) + c_2\omega(x_n) + \dots + c_m\omega(x_n) = f_n. \end{cases} \quad (3.3)$$

або в матричній формі $\Omega c = f$.

3.1.4. Інтерполяція алгебраїчними многочленами

Алгебраїчна інтерполяція полягає в тому, що *інтерполяційна функція є алгебраїчним многочленом* зі степенем, на одиницю меншим за кількість вузлів.

Нехай задані вузли $x_i, i = 0, 1, \dots, n$, серед яких немає таких, що співпадають, $x_k \neq x_j$ при $0 \leq k \leq n, 0 \leq j \leq n, j \neq k$ і задані значення функції $f(x)$ у цих вузлах:

$$y_0 = f(x_0), y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n) \quad (3.4)$$

Інтерполяційна функція є алгебраїчним многочленом

$$P_n(x) = \sum_{i=0}^n c_i x^i = c_0 + c_1 x + c_2 x^2 + \dots + c_n x^n, \quad (3.5)$$

названим *інтерполяційним многочленом*.

Для знаходження коефіцієнтів c_i многочлена використовують умову (3.2), тобто

$$P_n(x_i) = f(x_i) = y_i. \quad (3.6)$$

Існування та єдиність многочлена (3.5) визначається наступною теоремою.

Теорема про існування та єдиність алгебраїчного многочлена

Теорема 3.1. Інтерполяційний многочлен (3.5), що задовольняє умовам (3.6) по заданій функції (3.4), має степінь не нижче n і є єдиним.

Доведення. Використовуючи умову (3.6), одержимо систему лінійних алгебраїчних рівнянь відносно коефіцієнтів c_i :

$$\begin{cases} a_0 + a_1x_0 + a_2x_0^2 + \dots + a_nx_0^n = y_0, \\ a_0 + a_1x_1 + a_2x_1^2 + \dots + a_nx_1^n = y_1, \\ \dots\dots\dots \\ a_0 + a_1x_n + a_2x_n^2 + \dots + a_nx_n^n = y_n. \end{cases} \quad (3.7)$$

або

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \dots & x_{n-1}^n \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \\ y_n \end{pmatrix}$$

Отримана система рівнянь однозначно розв'язна (тобто розв'язок існує і він єдиний), оскільки за умовою $x_i, i = 0, 1, \dots, n$ різні.

Таким чином, коефіцієнти c_0, c_1, \dots, c_n системи, отримані у результаті розв'язання (3.7), визначають єдиний *інтерполяційний многочлен*, що має степінь не нижчу, ніж n , та побудований по $(n + 1)$ різних точках.

Розглянемо один зі способів побудови інтерполяційного многочлена.

3.1.5. Побудова інтерполяційного многочлена

Нехай відомі значення функції $f(x)$ у вузлах x_0, x_1 , тобто

$$y_0 = f(x_0), y_1 = f(x_1).$$

Побудувати інтерполяційний многочлен $P(x) = a_0 + a_1x$, який співпадає зі значеннями $f(x)$ у вузлах x_0, x_1 .

Розв'язок. Запишемо систему відносно a_0 та a_1

$$\begin{aligned} a_0 + a_1x_0 &= y_0, \\ a_0 + a_1x_1 &= y_1. \end{aligned} \quad \text{або} \quad \begin{pmatrix} 1 & x_0 \\ 1 & x_1 \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \end{pmatrix}.$$

Розв'яжемо дану систему методом виключення:

1. $a_0 = y_0 - a_1x_0$, визначаємо a_0 з рівняння 1
2. $y_0 - a_1x_0 + a_1x_1 = y_1$, підставляємо a_0 у рівняння 2
3. $a_1(x_1 - x_0) = y_1 - y_0$, зводимо подібні члени
4. $a_1 = \frac{y_1 - y_0}{x_1 - x_0}$, визначаємо значення a_1
5. $a_0 = y_0 - \frac{y_1 - y_0}{x_1 - x_0}x_0$, підставляємо значення a_1 у рівняння 1
6. $a_0 = \frac{y_0(x_1 - x_0) - x_0(y_1 - y_0)}{x_1 - x_0}$, приводимо до загального знаменника
7. $a_0 = \frac{y_0x_1 - y_0x_0 - x_0y_1 + x_0y_0}{x_1 - x_0}$ розкриваємо дужки
8. $a_0 = \frac{y_0x_1 - y_1x_0}{x_1 - x_0}$ визначаємо значення a_0

Будуємо інтерполяційний многочлен, підставивши у вираз

$$P(x) = a_0 + a_1x,$$

значення коефіцієнтів a_0 і a_1

$$P(x) = a_0 + a_1x = \frac{y_0x_1 - y_1x_0}{x_1 - x_0} + \frac{y_1 - y_0}{x_1 - x_0}x.$$

Висновок. Для довільної функції, заданої в точках

$$y_0 = f(x_0), y_1 = f(x_1)$$

існує інтерполяційний поліном

$$P(x) = \frac{y_0 x_1 - y_1 x_0}{x_1 - x_0} + \frac{y_1 - y_0}{x_1 - x_0} x, \text{ який співпадає зі значеннями функції}$$

$f(x)$ у точках y_0 і y_1 .

Перетворимо отриманий поліном

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

у такий спосіб

$$\begin{aligned} P(x) &= \frac{y_0 x_1}{x_1 - x_0} - \frac{y_1 x_0}{x_1 - x_0} + \frac{y_1 x}{x_1 - x_0} - \frac{y_0 x}{x_1 - x_0} = \\ &= \frac{y_0(x_1 - x)}{x_1 - x_0} + \frac{y_1(x - x_0)}{x_1 - x_0} = \\ &= \frac{x - x_1}{x_0 - x_1} y_0 + \frac{x - x_0}{x_1 - x_0} y_1 - \text{інтерполяційний поліном Лагранжа для двох} \end{aligned}$$

вузлів.

Контрольні запитання

1. Сформулюйте задачу інтерполяції.
2. Вкажіть сфери застосування інтерполяції.
3. Запишіть загальний вигляд алгебраїчного інтерполяційного многочлена.
4. Сформулюйте теорему про існування та єдиність алгебраїчного многочлена.
5. Побудуйте інтерполяційний многочлен першого степеня.

3.2. Інтерполяційний многочлен Лагранжа

Многочлен Лагранжа використовують для інтерполяції як з довільно заданими, так і рівновіддаленими вузлами.

3.2.1. Постановка задачі для многочлена Лагранжа

Нехай для функції $y = f(x)$ задані значення $y_i = f(x_i)$ у нерівновіддалених $(n+1)$ вузлах інтерполяції

$$y_0 = f(x_0), y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n).$$

Потрібно побудувати многочлен $L_n(x)$ зі степенем не вище n , який набуває в заданих вузлах x_i , $i = 0, 1, \dots, n$ значень, що співпадають зі значеннями функції $f(x)$,

$$L_n(x_i) = y_i, i = 0, 1, 2, \dots, n.$$

Існування та єдиність многочлена Лагранжа визначається наступною теоремою.

3.2.2. Теорема про існування та єдиність многочлена Лагранжа

Теорема 3.2. Нехай задані вузли $x_i, i = 0, 1, \dots, n$, серед яких немає таких, які співпадають, $x_k \neq x_j$ при $0 \leq k \leq n, 0 \leq j \leq n, j \neq k$, і задані значення функції $f(x)$ у цих вузлах

$$y_0 = f(x_0), y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n).$$

Тоді існує, і притому єдиний, многочлен [22]

$$L_n(x) = \sum_{i=0}^n \frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)} y_i \quad (3.8)$$

зі степенем не вище n , який набуває в заданих вузлах $x_i, i = 0, 1, \dots, n$, заданих значень $y_i, L_n(x_i) = y_i, i = 0, 1, \dots, n$.

$$L_n(x) = \sum_{i=0}^n \left[f(x_i) \prod_{\substack{j=0 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} \right]$$

Многочлен (3.8) називають *інтерполяційним многочленом Лагранжа* для нерівновіддалених вузлів, а коефіцієнти

$$\frac{(x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)} \quad (3.9)$$

при y_i називають лагранжевими коефіцієнтами.

3.2.3. Скорочена форма запису многочлена

Введемо допоміжний многочлен $w_{n+1}(x)$ зі степенем $n + 1$

$$w_{n+1}(x) = (x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_i)(x - x_{i+1}) \dots (x - x_n).$$

Обчислимо похідну, застосовуючи послідовно вираз для обчислення похідної добутку функцій: $z' = (uv)' = u'v + uv'$.

Приклад 3.1. Визначення $w'_3(x)$

$$w_3(x) = (x - x_0)(x - x_1)(x - x_2)$$

$$w'_3(x) = (x - x_1)(x - x_2) + (x - x_0)((x - x_1) + (x - x_2))$$

$$w'_3(x) = (x - x_1)(x - x_2) + (x - x_0)(x - x_2) + (x - x_0)(x - x_1)$$

$$w'_3(x) = \sum_{i=0}^2 \prod_{\substack{j=0 \\ j \neq i}}^2 (x - x_j)$$

У результаті одержимо

$$w'_{n+1}(x) = \sum_{i=0}^n (x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n).$$

Узагальнивши отримані вирази, одержимо:

$$w_{n+1}(x) = \prod_{i=0}^n (x - x_i), \quad w'_{n+1}(x) = \sum_{i=0}^n \prod_{\substack{j=0 \\ i \neq j}}^n (x - x_j)$$

Похідна цього многочлена в точці $x = x_i$ дорівнює:

$$w'_3(x_0) = (x_0 - x_1)(x_0 - x_2); \quad w'_3(x_1) = (x_1 - x_0)(x_1 - x_2)$$

$$w'_3(x_2) = (x_2 - x_0)(x_2 - x_1)$$

$$w'_{n+1}(x_i) = (x_i - x_0)(x_i - x_1) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n).$$

Тоді поліном Лагранжа можна записати у вигляді:

$$L_n(x) = \sum_{i=0}^n \frac{w_{n+1}(x)}{(x - x_i)w'_{n+1}(x_i)} y_i = w_{n+1}(x) \sum_{i=0}^n \frac{y_i}{(x - x_i)w'_{n+1}(x_i)}. \quad (3.10)$$

Вираз (3.10) називають скороченою формою запису многочлена Лагранжа. Розглянемо приклад використання многочлена Лагранжа.

Приклад 3.2. Визначення значення функції в точці

Для функції, заданої таблично, обчислити за допомогою многочлена Лагранжа значення функції в заданій точці x^* , відмінній від вузлової.

Таблиця 3.1

Значення функції у вузлових точках

i	0	1	2	3
x_i	2,10	2,67	3,01	3,82
y_i	122,23	123,45	120,02	119,65

Розв'язок

1. Побудуємо многочлен Лагранжа з урахуванням заданого числа вузлів, $n = 3$, маємо

$$L_3(x) = \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} y_0 + \frac{(x - x_0)(x - x_2)(x - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} y_1 +$$

$$+ \frac{(x - x_0)(x - x_1)(x - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} y_2 + \frac{(x - x_0)(x - x_1)(x - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} y_3.$$

2. Обчислимо значення функції в заданій точці $x^* = 2,2$:

$$\begin{aligned} f(2,2) &\approx L_3(2,2) = \\ &= \frac{(2,2-2,67)(2,2-3,01)(2,2-3,82)}{(2,10-2,67)(2,10-3,01)(2,10-3,82)} 122,23 + \\ &+ \frac{(2,2-2,10)(2,2-3,01)(2,2-3,82)}{(2,67-2,10)(2,67-3,01)(2,67-3,82)} 123,45 + \\ &+ \frac{(2,2-2,10)(2,2-2,67)(2,2-3,82)}{(3,01-2,10)(3,01-2,67)(3,01-3,82)} 120,02 + \\ &+ \frac{(2,2-2,10)(2,2-2,67)(2,2-3,01)}{(3,82-2,10)(3,82-2,67)(3,82-3,01)} 119,65 \approx 122,56. \end{aligned}$$

3.2.4. Похибка та оцінка похибки многочлена Лагранжа

При заміні функції $f(x)$ многочленом $L_n(x)$ виникає похибка $R_n(x) = f(x) - L_n(x)$, названа також залишковим членом інтерполяційної формули, $f(x) = L_n(x) + R_n(x)$.

Теорема 3.3. Похибку інтерполяційного многочлена Лагранжа для довільно заданих вузлів визначають за формулою

$$R_n(x) = \frac{w_{n+1}(x)}{(n+1)!} f^{(n+1)}(\xi). \quad (3.11)$$

де $w_{n+1}(x) = (x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})(x-x_{i+2})\dots(x-x_n)$.

Через невизначеність точки ξ визначити точно $R_n(x)$ не можна, тому при проведенні обчислень використовують тільки наближені оцінки похибок інтерполяції.

Оцінку похибки (залишкового члена) інтерполяції многочленом Лагранжа в деякій довільній фіксованій точці x^* з відрізка $[a, b]$, $x^* \in [a, b]$ визначають за формулою

$$|R_n| = \left| f(x^*) - L_n(x^*) \right| \leq \frac{C_{n+1}}{(n+1)!} |w_{n+1}(x^*)|, \quad (3.12)$$

$$C_{n+1} = \max \left| f^{(n+1)}(x) \right| \text{ на } [a, b].$$

Оцінка максимальної похибки

Оцінка максимальної похибки інтерполяції на всьому відрізку $[a, b]$, тобто в будь-якій точці $x \in [a, b]$, має вигляд

$$|R_n| = |f(x) - L_n(x)| \leq \frac{C_{n+1}}{(n+1)!} |w_{n+1}(x)| \quad (3.13)$$

Розглянемо приклад використання оцінок при інтерполяції многочленом Лагранжа.

Приклад 3.3. Визначення точності обчислення значення функції

Нехай потрібно визначити, з якою точністю можна обчислити значення функції $y = \sqrt{x}$ у точці $x^* = 112$ за допомогою інтерполяційної формули Лагранжа, якщо задані вузли $x_0 = 100$, $x_1 = 118$, $x_2 = 138$.

Розв'язок. Оскільки потрібно обчислити похибку в одній точці $x^* = 112$, то застосовуємо формулу (3.12). Визначимо значення C_3 :

$$y' = \frac{1}{2}x^{-\frac{1}{2}}; y'' = -\frac{1}{4}x^{-\frac{3}{2}}; y''' = \frac{3}{8}x^{-\frac{5}{2}}, \text{ тоді}$$

$$C_3 = \max |y''| = \frac{3}{8} \frac{1}{\sqrt{100^5}} = \frac{3}{8} \cdot 10^{-5} \text{ при } 100 \leq x \leq 138.$$

Далі,

$$w_3(x^*) = |(112 - 100)(112 - 118)(112 - 138)| = |12 \cdot (-6) \cdot (-26)| = 1872,$$

$$|R_2| \leq \frac{M_3}{3!} |w_3(x^*)|, \quad |R_2| \leq \frac{3}{8} \cdot 10^{-5} \cdot \frac{1}{3!} \cdot 1872 \approx 1,17 \cdot 10^{-3}.$$

Контрольні запитання

1. Запишіть поліном Лагранжа степеня 3 для довільно віддалених вузлів.
2. Запишіть поліном Лагранжа степеня 3 в скороченій формі для довільно віддалених вузлів.
3. Для функції $f(x)$ зі значеннями $f(1) = 3$, $f(2) = 12$, $f(4) = 6$, побудувати поліном Лагранжа. Відповідь представити у вигляді:

$$f(x) = ax^2 + bx + c.$$

4. Для функції $f(x)$ зі значеннями $f(4) = 80$, $f(8) = 48$, $f(14) = 60$ побудувати поліном Лагранжа. Відповідь представити у вигляді:
 $f(x) = ax^2 + bx + c.$

5. Для функції $f(x)$ зі значеннями:
 $f(2.10) = 122.23$, $f(2.67) = 123.4$, $f(3.01) = 120.02$, $f(3.82) = 119.65$,
 обчислити за допомогою многочлена Лагранжа значення функції в заданій точці $x^* = 3.2$.

3.3. Інтерполяційний многочлен Лагранжа для рівновіддалених вузлів

3.3.1. Створення многочлена Лагранжа для рівновіддалених вузлів

Теорема 3.4. Нехай задані рівновіддалені вузли інтерполяції

$$x_{i+1} - x_i = h = \text{const}, \quad i = 0, 1, \dots, n-1, \quad \text{і задані значення}$$

$y_0 = f(x_0), y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n)$ функції $f(x)$ у цих вузлах.

Тоді існує

$$L_n(x) = L_n(x_0 + mh) = \sum_{i=0}^n (-1)^{n-i} \frac{\prod_{j=0, j \neq i}^n (m-j)}{i!(n-i)!} y_i, \quad (3.14)$$

або

$$L_n(x) = L_n(x_0 + mh) = \frac{1}{n!} v_{n+1}(m) \sum_{i=0}^n (-1)^{n-i} \frac{C_n^i}{m-i} y_i \quad (3.15)$$

зі степенем не вище n , що має в заданих вузлах x_i , $i = 0, 1, \dots, n$ задані значення
 $y_i, x_1 - x_0 = x_2 - x_1 = \dots = x_{i+1} - x_i = \dots = x_n - x_{n-1} = h$ $L_n(x_i) = y_i$,
 $i = 0, 1, 2, \dots, n$.

Доведення. Оскільки за умовою вузли рівновіддалені,

$$x_1 - x_0 = x_2 - x_1 = \dots = x_{i+1} - x_i = \dots = x_n - x_{n-1} = h, \text{ то}$$

$$x_{i+1} = x_i + h = x_0 + (i+1)h, \quad i = 0, 1, \dots, n-1.$$

Далі, використовуємо позначення $x - x_0 = mh$:

$$x - x_1 = x - x_0 - h = mh - h = h(m-1),$$

$$x - x_2 = x - x_0 - 2h = mh - 2h = h(m-2),$$

.....

$$x - x_n = x - x_0 - nh = mh - nh = h(m-n),$$

Для фіксованих точок: $x_i = x_{i-1} + h = x_0 + ih$,

$$x_i - x_0 = x_0 + ih - x_0 = ih,$$

$$x_i - x_1 = x_0 + ih - x_0 - h = h(i-1),$$

.....

$$x_i - x_{i-1} = x_0 + ih - x_0 - (i-1)h = h,$$

$$x_i - x_{i+1} = x_0 + ih - x_0 - (i+1)h = -h$$

$$x_i - x_{i+2} = x_0 + ih - x_0 - (i+2)h = -2h$$

.....

$$x_i - x_n = x_0 + ih - x_0 - nh = -h(n-i).$$

Тоді i -й лагранжевий коефіцієнт (9) матиме вигляд

$$\frac{m(m-1)(m-2)\dots(m-(i-1))(m-(i+1))\dots(m-n)}{i(i-1)\dots 1 \cdot (-1)(-2)\dots(-(n-i))} = \frac{\prod_{j \neq i, j=0}^n (m-j)}{(-1)^{n-i} i!(n-i)!},$$

а многочлен Лагранжа набуває вигляду (3.14):

$$L_n(x) = L_n(x_0 + mh) = \sum_{i=0}^n (-1)^{n-i} \frac{\prod_{j=0, j \neq i}^n (m-j)}{i!(n-i)!} y_i.$$

Для одержання формули (3.15) представимо многочлени $w_{n+1}(x)$ і його похідну $w'_{n+1}(x_i)$ з формули (3.10), позначивши $x - x_0 = mh$, таким чином:

$$\begin{aligned} w_{n+1}(x) &= (x - x_0)(x - x_1) \dots \\ &\dots (x - x_{i-1})(x - x_i)(x - x_{i+1}) \dots (x - x_n) = \\ &= h^{n+1} m(m-1)(m-2) \dots (m-n) = w_{n+1}(m) = h^{n+1} v_{n+1}(m) \\ w'_{n+1}(x_i) &= (x_i - x_0)(x_i - x_1) \dots \\ &\dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n) = \\ &= h^{n+1} i(i-1)(i-2) \dots 1(-1)(-2) \dots (-(n-i)) = \\ &= h^{n+1} (-1)^{n-i} i!(n-i)! = w'_{n+1}(i) = h^{n+1} v'_{n+1}(i) \end{aligned}$$

Підставимо значення поліномів $w_{n+1}(x)$ і $w'_{n+1}(x)$ у формулу спрощеного лагранжевого коефіцієнта для нерівновіддалених вузлів:

$$\frac{w_{n+1}(x)}{(x - x_i) w'_{n+1}(x_i)}$$

Оскільки $\frac{w_{n+1}(x)}{x - x_i} = \frac{w_{n+1}(m)}{m - i} = \frac{h^{n+1} v_{n+1}(m)}{m - i}$

та $w'_{n+1}(x) = w'_{n+1}(m) = h^{n+1} (-1)^{n-i} i!(n-i)! = h^{n+1} v'_{n+1}(m)$

Тоді лагранжевий коефіцієнт набуде вигляду

$$\begin{aligned} \frac{w_{n+1}(m)}{(m - i) w'_{n+1}(m)} &= \frac{h^{n+1} v_{n+1}(m)}{h^{n+1} (m - i) v'_{n+1}(m)} = \\ &= \frac{(-1)^{n-i} v_{n+1}(m)}{(m - i) i!(n - i)!} = \frac{1}{n!} v_{n+1}(m) \frac{(-1)^{n-i} C_n^i}{m - i}, \quad i = 0, 1, 2, \dots, n, \end{aligned}$$

де $C_n^i = \frac{n!}{i!(n-i)!}$ – комбінація з n по i .

Визначення (довідково). Комбінаціями з n різних елементів по i елементів називають комбінації, які складені з даних n елементів по i елементів і відрізняються хоча б одним елементом.

Усі комбінації із множини $\{a,b,c,d,e\}$ по два – $ab, ac, ad, ae, bc, bd, be, cd, ce, de$.

Звідси формула полінома Лагранжа

$$L_n(x) = L_n(x_0 + mh) = \frac{1}{n!} v_{n+1}(m) \sum_{i=0}^n (-1)^{n-i} \frac{C_n^i}{m-i} y_i$$

Розглянемо приклад застосування многочлена Лагранжа для рівновіддалених значень аргументу x .

Приклад 3.4. *Визначення значення функції в точці*

Для функції $f(x) = e^x$, заданої таблицею, обчислити за допомогою многочлена Лагранжа для рівновіддалених вузлів значення функції в заданій точці $x^* = 0,022$.

Таблиця 3.2

Таблиця значень функції

i	0	1	2	3	4
x_i	0,0	0,01	0,02	0,03	0,04
$f(x_i)$	1,0000	1,0101	1,0202	1,0305	1,0408

Розв'язок

Етап 1. Знайдемо значення m , що відповідає $x = 0,022$. Вузли рівновіддалені з кроком $h = 0,01$. Зробимо лінійну заміну $x - x_0 = mh$, тоді

$$x \text{ буде відповідати значення } m = \frac{x^* - x_0}{h} = \frac{0,022 - 0}{0,01} = 2,2.$$

Етап 2. Використовуємо многочлен Лагранжа:

$$L_n(x) = L_n(x_0 + mh) = \frac{1}{n!} v_{n+1}(m) \sum_{i=0}^n (-1)^{n-i} \frac{C_n^i}{m-i} y_i$$

при знайденому значенні m :

$$L_4(2,2) = \frac{1}{4!} v_5(2,2) \sum_{i=0}^4 (-1)^{4-i} \frac{C_4^i}{(2,2-i)} y_i,$$

Підставимо у формулу вираз для кількості комбінацій C_4^i

$$L_4(2,2) = v_5(2,2) \sum_{i=0}^4 (-1)^{4-i} \frac{y_i}{(2,2-i)i!(n-i)!}$$

Етап 3. Обчислюємо значення $w_5(2,2)$:

$$\begin{aligned} v_5(2,2) &= \\ &= (2,2-0)(2,2-1)(2,2-2)(2,2-3)(2,2-4) \approx 0,76032 \end{aligned}$$

Етап 4. Обчислюємо $\frac{1}{(2,2-i)i!(n-i)!}$ для $i = 0,1,2,3,4$

$$i=0 \rightarrow \frac{1}{(2,2-0)4!} \approx 0,01894;$$

$$i \rightarrow 1 \rightarrow \frac{1}{(2,2-1) \cdot 1 \cdot 3!} \approx 0,13889;$$

$$i=3 \rightarrow \frac{1}{(2,2-3) \cdot 3! \cdot 1} \approx -0,20833;$$

$$i=2 \rightarrow \frac{1}{(2,2-2) \cdot 2! \cdot 2!} \approx 1,25;$$

$$i=4 \rightarrow \frac{1}{(2,2-4)4!} \approx -0,02315.$$

Остаточно маємо

$$\begin{aligned} L_4(2,2) &= 0,76032 \cdot \left((-1)^4 \cdot 0,01894 \cdot y_0 + (-1)^3 \cdot 0,13889 \cdot y_1 + \right. \\ &\left. + (-1)^2 \cdot 1,25 \cdot y_2 - (-1)^1 \cdot 0,20833 y_3 - (-1)^0 \cdot 0,02315 y_4 \right) \end{aligned}$$

$$\begin{aligned} f(2,2) = L_4(2,2) &= 0,76032 \cdot (0,01894 \cdot 1,0000 - 0,13889 \cdot 1,0101 + \\ &+ 1,25 \cdot 1,0202 + 0,20833 \cdot 1,0305 - 0,02315 \cdot 1,0408) \approx 1,0222. \end{aligned}$$

3.3.2. Похибка інтерполяційного многочлена Лагранжа для рівновіддалених вузлів

Похибку інтерполяційного многочлена Лагранжа для рівновіддалених вузлів визначають за формулою

$$R_n(x) = h^{n+1} \frac{v_{n+1}(m)}{(n+1)!} f^{(n+1)}(\xi) \quad (3.16)$$

де $v_{n+1}(m) = m(m-1)(m-2)\dots(m-n)$, $\xi \in [a, b]$.

Оцінка похибки. Оцінку похибки (залишкового члена) інтерполяції многочленом Лагранжа для рівновіддалених вузлів у деякій довільній фіксованій точці x^* з відрізка $[a, b]$, $x \in [a, b]$ визначають за формулою

$$|R_n| = \left| f(x^*) - L_n(x^*) \right| \leq h^{n+1} \frac{C_{n+1}}{(n+1)!} |v_{n+1}(m)|, \quad (3.17)$$

де $C_{n+1} = \max |f^{(n+1)}|$ на $[a, b]$.

Оцінка максимальної похибки. Оцінка максимальної похибки інтерполяції на всьому відрізку $[a, b]$, тобто у будь-якій точці $x \in [a, b]$, має вигляд

$$|R_n| = |f(x) - L_n(x)| \leq h^{n+1} \frac{C_{n+1}}{(n+1)!} M_n, \quad (3.18)$$

де $M_n = \max |v_{n+1}(m)| = \max |m(m-1)(m-2)\dots(m-n)|$ на відрізку $[a, b]$.

Контрольні запитання

1. Запишіть поліном Лагранжа степеня 3 для рівновіддалених вузлів.
2. Запишіть поліном Лагранжа степеня 3 у скороченій формі для рівновіддалених вузлів.
3. Для функції $f(x)$ зі значеннями $f(1) = 4$, $f(2) = 3$, $f(3) = 6$ побудувати поліном Лагранжа для рівновіддалених вузлів. Відповідь представити у вигляді: $f(x) = ax^2 + bx + c$.
4. Для функції $f(x)$ зі значеннями $f(0) = 4$, $f(1) = 2$, $f(2) = 8$ побудувати поліном Лагранжа для рівновіддалених вузлів. Відповідь представити у вигляді: $f(x) = ax^2 + bx + c$.
5. Для функції $f(x)$ зі значеннями $f(0) = 4$, $f(1) = 3$, $f(2) = 6$ побудувати поліном Лагранжа для рівновіддалених вузлів. Відповідь представити у вигляді: $f(x) = ax^2 + bx + c$.

3.4. Обернена інтерполяція многочленом Лагранжа

Поряд із задачею інтерполяції в технічних застосуваннях існує задача оберненої інтерполяції. Нехай відома залежність $y = f(x)$, у точках $x_i, i=0,1,\dots,n$, тобто відомі $y_i = f(x_i)$. Ця інформація еквівалентна тому, що відомі значення $x_i = g(y_i)$ – оберненої функції. За умови допустимості інтерполяції по змінній y можна замінити обернену функцію $g(y)$ інтерполяційним многочленом $L_n(y_i) = x_i, i = 0,1,\dots,n$.

Приклад 3.5. Відновлення вхідного сигналу

Потрібно відновити форму вхідного сигналу $x(t)$ (передбачається, що це якийсь низькочастотний сигнал). Зв'язок миттєвих значень вхідного сигналу $x(t)$ й вихідного сигналу $y(t)$ визначається нелінійною динамічною характеристикою $y = \varphi(x)$. Задачу розв'язати методом оберненої інтерполяції.

Таблиця 3.3

Таблиця значень оберненої функції

x	-0.9	-0.3	0.3	0.9
$y = \varphi(x)$	0.31623	0.83666	1.14017	1.37840

Розв'язок

Будуємо многочлен Лагранжа третього порядку

$$L_3(y) = \frac{(y - y_1)(y - y_2)(y - y_3)}{(y_0 - y_1)(y_0 - y_2)(y_0 - y_3)} x_0 + \frac{(y - y_0)(y - y_2)(y - y_3)}{(y_1 - y_0)(y_1 - y_2)(y_1 - y_3)} x_1 + \\ + \frac{(y - y_0)(y - y_1)(y - y_3)}{(y_2 - y_0)(y_2 - y_1)(y_2 - y_3)} x_2 + \frac{(y - y_0)(y - y_1)(y - y_2)}{(y_3 - y_0)(y_3 - y_1)(y_3 - y_2)} x_3.$$

Підставляючи табличні значення, одержуємо

$$L_3(y) = -0.00638y^3 + 1.01572y^2 + 0.01232y - 0.9976.$$

Таким чином, форма вхідного сигналу

$$x(y) = -0.9976 - 0.1232y + 1.01572y^2 - 0.00638y^3$$

для $y \in [-0.5; 0.25]$.

Основні формули з теми стосовно полінома Лагранжа

1. Поліном Лагранжа для нерівновіддалених вузлів

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

$$L_n(x) = w_{n+1}(x) \sum_{i=0}^n \frac{y_i}{(x - x_i) w'_{n+1}(x_i)}$$

$$w_{n+1}(x) = \prod_{i=0}^n (x - x_i) \quad w'_{n+1}(x) = \sum_{i=0}^n \prod_{\substack{j=0 \\ i \neq j}}^n (x_i - x_j)$$

2. Поліноми Лагранжа для рівновіддалених вузлів:

$$L_n(x) = L_n(x_0 + mh) = \sum_{i=0}^n (-1)^{n-i} \frac{\prod_{i \neq j, j=0}^n (m - j)}{i!(n-i)!} y_i$$

$$L_n(x) = L_n(x_0 + mh) = \frac{1}{n!} v_{n+1}(m) \sum_{i=0}^n (-1)^{n-i} \frac{C_n^i}{m-i} y_i$$

$$v_{n+1}(m) = \frac{1}{h^{n+1}} \prod_{i=0}^n (m - i), \quad m = \frac{x - x_0}{h}$$

$$C_n^i = \frac{n!}{i!(n-i)!}$$

Похибки

3. Для нерівновіддалених вузлів:

$$R_n(x) = \frac{w_{n+1}(x)}{(n+1)!} f^{(n+1)}(\xi), \quad w_{n+1}(x) = \prod_{i=0}^n (x - x_i)$$

4. Для рівновіддалених вузлів:

$$R_n(x) = \frac{w_{n+1}(m)}{(n+1)!} f^{(n+1)}(\xi) = h^{n+1} \frac{v_{n+1}(m)}{(n+1)!} f^{(n+1)}(\xi)$$

$$w_{n+1}(m) = h_{n+1} v_{n+1}(m), \quad v_{n+1}(m) = \prod_{i=0}^n (m - i), \quad m = \frac{x - x_0}{h}$$

Контрольні завдання

1. Дайте визначення оберненого полінома Лагранжа.

2. Дано функцію зі значеннями

$$y_0 = \varphi(-0.9) = 0.3, \quad y_1 = \varphi(-0.3) = 0.8, \quad y_2 = \varphi(0.3) = 1.$$

Побудувати обернений поліном Лагранжа у вигляді:

$$f(y) = ay^2 + by + c.$$

3. Для функції

$$y_0 = \varphi(-0.9) = 0.2, \quad y_1 = \varphi(-0.3) = 0.7, \quad y_2 = \varphi(0.3) = 1.2 \text{ знайти}$$

значення аргумента, при якому функція набуває значення $y^* = 0.5$.

4. Для функції

$$y_0 = \varphi(1) = 0, \quad y_1 = \varphi(1.1) = 0.2, \quad y_2 = \varphi(1.3) = 0.3 \text{ знайти значення}$$

аргумента, при якому функція набуває значення $y^* = 0.25$.

5. Запишіть відомі вам види запису полінома Лагранжа.

3.5. Інтерполяційний многочлен Ньютона

3.5.1. Розділені різниці

1. *Розділені різниці першого порядку.* Нехай дано набір x_0, x_1, \dots, x_n , $x_i \neq x_j$ при $i \neq j$, $x_i \in [a, b]$, на якому деяка функція $f(x) \in R$ набуває значень $f(x_0), f(x_1), \dots, f(x_n)$.

Відношення виду

$$\frac{f(x_1) - f(x_0)}{x_1 - x_0} = f(x_0; x_1) = y_{10}; \quad \frac{f(x_2) - f(x_1)}{x_2 - x_1} = f(x_1; x_2) = y_{21};$$
$$\dots; \quad \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} = f(x_{n-1}; x_n) = y_{n, n-1},$$

$$\text{або} \quad \frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} = f(x_i; x_{i+1}) = y_{i+1, i}, \quad i = 0, 1, \dots, n-1$$

називають розділеними різницями першого порядку, які визначаються для двох точок x_i, x_{i+1} .

2. *Розділені різниці другого порядку*

Відношення

$$\frac{f(x_1; x_2) - f(x_0; x_1)}{x_2 - x_0} = f(x_0; x_1; x_2) = y_{210};$$
$$\frac{f(x_2; x_3) - f(x_1; x_2)}{x_3 - x_1} = f(x_1; x_2; x_3) = y_{321}; \dots$$
$$\dots; \quad \frac{f(x_{n-1}; x_n) - f(x_{n-2}; x_{n-1})}{x_n - x_{n-2}} = f(x_{n-2}; x_{n-1}; x_n) = y_{n, n-1, n-2},$$

або

$$\frac{f(x_{i+1}; x_{i+2}) - f(x_i; x_{i+1})}{x_{i+2} - x_i} = f(x_i; x_{i+1}; x_{i+2}) = y_{i+2, i+1, i},$$
$$i = 0, 1, 2, \dots, n-2$$

називають розділеними різницями другого порядку й визначають для трьох точок x_i, x_{i+1}, x_{i+2} .

3. Розділені різниці k -го порядку

Розділеними різницями k -го порядку називають відношення

$$\frac{f(x_1; \dots; x_k) - f(x_0; \dots; x_{k-1})}{x_k - x_0} = f(x_0; \dots; x_k) = y_{k \dots 0};$$

$$\frac{f(x_2; \dots; x_{k+1}) - f(x_1; \dots; x_k)}{x_{k+1} - x_1} = f(x_1; \dots; x_{k+1}) = y_{k+1 \dots 1}; \dots$$

$$\dots \frac{f(x_{n-k+1}; \dots; x_n) - f(x_{n-k}; \dots; x_{n-1})}{x_n - x_{n-k}} = f(x_{n-k}; \dots; x_n) = y_{n \dots n-k};$$

$$\frac{f(x_{i+1}; \dots; x_{i+k}) - f(x_{n-k}; \dots; x_{n-1})}{x_{i+k} - x_i} = f(x_i; \dots; x_{i+k}) = y_{i+k \dots i},$$

$$i = 0, 1, \dots, n - k$$

і визначають для $x_i, x_{i+1}, \dots, x_{i+k}$.

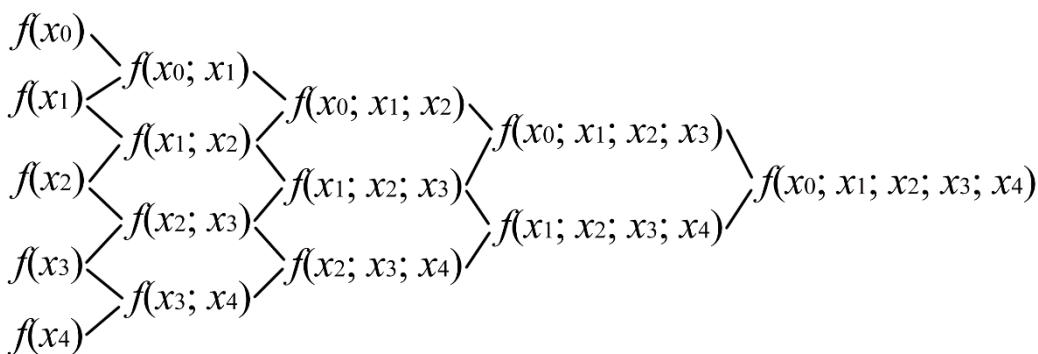
4. Розділені різниці n -го порядку.

Нарешті, розділеною різницею n -го порядку називають відношення

$$\frac{f(x_1; \dots; x_n) - f(x_0; \dots; x_{n-1})}{x_n - x_0} = f(x_0; \dots; x_n) = y_{n \dots 0}$$

і визначають для x_0, x_1, \dots, x_n .

Приклад 3.6. Взаємне розташування розділених різниць



3.5.2. Вираження розділених різниць через значення функцій

Перетворимо розділену різницю $f(x_0; x_1; x_2)$ до виду:

$$f(x_0; x_1; x_2) = \frac{f(x_1; x_2) - f(x_0; x_1)}{x_2 - x_0} = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

$$= \frac{f(x_2)(x_1 - x_0) - f(x_1)(x_1 - x_0) - f(x_1)(x_2 - x_1) + f(x_0)(x_2 - x_1)}{(x_2 - x_1)(x_1 - x_0)(x_2 - x_0)}$$

$$= \frac{f(x_0)}{(x_0 - x_1)(x_0 - x_2)} - \frac{f(x_1)(x_1 - x_0 + x_2 - x_1)}{(x_2 - x_1)(x_1 - x_0)(x_2 - x_0)} + \frac{f(x_2)}{(x_2 - x_1)(x_2 - x_0)}$$

$$= \frac{f(x_0)}{(x_0 - x_1)(x_0 - x_2)} + \frac{f(x_1)}{(x_1 - x_2)(x_1 - x_0)} + \frac{f(x_2)}{(x_2 - x_1)(x_2 - x_0)}$$

Отже,

$$f(x_0; x_1; x_2) = \frac{f(x_0)}{(x_0 - x_1)(x_0 - x_2)} + \frac{f(x_1)}{(x_1 - x_2)(x_1 - x_0)} + \frac{f(x_2)}{(x_2 - x_1)(x_2 - x_0)}$$

Узагальнивши отриманий вираз для k -ї розділеної різниці, одержимо:

$$f(x_0; x_1; \dots; x_k) = \sum_{i=0}^k \frac{f(x_i)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_k)}$$

Приклад 3.7. Побудова розділених різниць

Нехай задані x_0, x_1, x_2, x_3 й відомі значення функції $f(x_0) = y_0, f(x_1) = y_1, f(x_2) = y_2, f(x_3) = y_3$.

Побудувати послідовність одержання розділених різниць.

Розв'язок. У цьому випадку $n = 3$. Послідовно будемо застосовувати формулу.

У результаті одержимо:

- розділені різниці першого порядку

$$\frac{y_1 - y_0}{x_1 - x_0} = y_{10}, \quad \frac{y_2 - y_1}{x_2 - x_1} = y_{21}, \quad \frac{y_3 - y_2}{x_3 - x_2} = y_{32}.$$

- розділені різниці другого порядку

$$\frac{y_{21} - y_{10}}{x_2 - x_0} = y_{210}, \frac{y_{32} - y_{21}}{x_3 - x_1} = y_{321} ;$$

- розділена різниця третього порядку

$$\frac{y_{321} - y_{210}}{x_3 - x_0} = y_{3210}.$$

Приклад 3.8. Обчислення розділених різниць за таблицею

Скласти розділені різниці для функції $y = f(x)$, заданої таблицею.

Таблиця 3.4

Таблиця значень функції

$x_0=1,00$	$x_1=1,02$	$x_2=1,03$	$x_3=1,06$	$x_4=1,08$
$y_0=3,162$	$y_1=3,194$	$y_2=3,209$	$y_3=3,256$	$y_4=3,286$

Розв'язок. Використовуючи схему побудови й позначень попереднього прикладу, будемо мати:

- розділені різниці першого порядку

$$y_{10} = \frac{y_1 - y_0}{x_1 - x_0} = \frac{3,194 - 3,162}{1,02 - 1,00} = 1,6, \quad y_{21} = \frac{y_2 - y_1}{x_2 - x_1} = \frac{3,209 - 3,194}{1,03 - 1,02} = 1,5$$

$$y_{32} = \frac{y_3 - y_2}{x_3 - x_2} = \frac{3,256 - 3,209}{1,06 - 1,03} \approx 1,566,$$

$$y_{43} = \frac{y_4 - y_3}{x_4 - x_3} = \frac{3,286 - 3,256}{1,08 - 1,06} = 1,5$$

- розділені різниці другого порядку

$$y_{210} = \frac{y_{21} - y_{10}}{x_2 - x_0} = \frac{1,5 - 1,6}{1,03 - 1,00} \approx -3,333,$$

$$y_{321} = \frac{y_{32} - y_{21}}{x_3 - x_1} = \frac{1,566 - 1,5}{1,06 - 1,02} \approx 1,65,$$

$$y_{432} = \frac{y_{43} - y_{32}}{x_4 - x_2} = \frac{1,5 - 1,566}{1,08 - 1,03} \approx -1,32,$$

- розділені різниці третього порядку

$$y_{3210} = \frac{y_{321} - y_{210}}{x_3 - x_0} = \frac{1,65 + 3,333}{1,06 - 1,00} \approx 83,05,$$

$$y_{4321} = \frac{y_{432} - y_{321}}{x_4 - x_1} = \frac{-1,32 - 1,65}{1,08 - 1,02} \approx -49,5.$$

- розділену різницю четвертого порядку:

$$y_{43210} = \frac{y_{4321} - y_{3210}}{x_4 - x_0} = \frac{-49,5 - 83,05}{1,08 - 1,00} \approx -1656,875$$

3.5.3. Інтерполяційний многочлен Ньютона для довільно заданих вузлів

Постановка задачі. Нехай для функції $y = f(x)$ задані значення $y_i = f(x_i)$ у нерівновіддалених $(n + 1)$ вузлах інтерполяції,

$$y_0 = f(x_0), y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n).$$

Потрібно побудувати многочлен $N_n(x)$ зі степенем, який не вище n та набуває у заданих вузлах $x_i, i = 0, 1, 2, \dots, n$ значень, що збігаються зі значеннями y_i ,

$$N_n(x_i) = y_i, \quad i = 0, 1, 2, \dots, n.$$

Теорема 3. 5. Нехай задані вузли $x_i, i = 0, 1, 2, \dots, n$, серед яких немає таких, що співпадають, $x_i \neq x_j, i \neq j$, і задані значення

$$y_0 = f(x_0), y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n)$$

функції $f(x)$ у цих вузлах. Тоді існує многочлен

$$\begin{aligned} N_n(x) = & f(x_0) + f(x_0; x_1)(x - x_0) + \\ & + f(x_0; x_1; x_2)(x - x_0)(x - x_1) + \dots \\ & \dots + f(x_0; x_1; \dots; x_n)(x - x_0)(x - x_1) \dots (x - x_{n-1}). \end{aligned}$$

зі степенем не вище n , який набуває в заданих вузлах $x_i, i = 0, 1, 2, \dots, n$ заданих значень $y_i, N_n(x_i) = y_i, i = 0, 1, \dots, n$.

Многочлен називають *інтерполяційним многочленом Ньютона* для нерівновіддалених вузлів [23].

Виведення формули многочлена Ньютона

Нехай $L_n(x)$ – інтерполяційний поліном Лагранжа, побудований для функції $f(x)$ по вузлах x_0, x_1, \dots, x_n . Тоді

$$L_n(x) = L_0(x) + [L_1(x) - L_0(x)] + [L_2(x) - L_1(x)] + \dots + [L_n(x) - L_{n-1}(x)]$$

Знайдемо різницю $L_2(x) - L_1(x)$:

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

$$\text{У загальному випадку } L_k(x) - L_{k-1}(x) = \frac{(x-x_0) \cdot \dots \cdot (x-x_{k-1})}{(x_k-x_0) \cdot \dots \cdot (x_k-x_{k-1})}y_k.$$

Дана різниця набуває значення 0 у точках x_0, x_1, \dots, x_{k-1} . Тому

$$L_k(x) - L_{k-1}(x) = A(x-x_0) \cdot \dots \cdot (x-x_{k-1})$$

Для визначення величини A приймемо $x = x_k$

$$f(x_k) - L_{k-1}(x_k) = A(x_k - x_0) \cdot \dots \cdot (x_k - x_{k-1}).$$

$$A = \frac{f(x_k)}{\prod_{i=0}^{k-1} (x_k - x_i)} - \frac{\sum_{j=0}^{k-1} f(x_j) \prod_{\substack{i=0 \\ i \neq j}}^{k-1} \frac{x_k - x_i}{x_j - x_i}}{\prod_{i=0}^{k-1} (x_k - x_i)} = \sum_{j=0}^{k-1} \frac{f(x_j)}{\prod_{i=0}^{k-1} (x_j - x_i)}$$

Розглянемо дане перетворення на прикладі $x = x_2$:

$$f(x_2) - L_1(x_2) = A(x_2 - x_0) \cdot (x_2 - x_1)$$

$$f(x_2) - f(x_0) \frac{(x_2 - x_1)}{(x_0 - x_1)} - f(x_1) \frac{(x_2 - x_0)}{(x_1 - x_0)} = A(x_2 - x_0) \cdot (x_2 - x_1)$$

$$A = \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)} - \frac{f(x_0) \frac{(x_2 - x_1)}{(x_0 - x_1)}}{(x_2 - x_0)(x_2 - x_1)} - \frac{f(x_1) \frac{(x_2 - x_0)}{(x_1 - x_0)}}{(x_2 - x_0)(x_2 - x_1)} =$$

$$= \frac{f(x_0)}{(x_0 - x_1)(x_0 - x_2)} + \frac{f(x_1)}{(x_1 - x_0)(x_1 - x_2)} + \frac{f(x_2)}{(x_2 - x_0)(x_2 - x_1)}$$

Згадаємо раніше виведений вираз для розділеної різниці другого порядку:

$$f(x_0; x_1; x_2) = \frac{f(x_0)}{(x_0 - x_1)(x_0 - x_2)} + \frac{f(x_1)}{(x_1 - x_2)(x_1 - x_0)} + \frac{f(x_2)}{(x_2 - x_1)(x_2 - x_0)}$$

Ми бачимо, що даний вираз збігається з виведеним виразом для A .

Тому $A = f(x_0; x_1; x_2)$.

У загальному випадку $A = f(x_0; x_1; \dots; x_k)$.

Враховуючи, що

$$L_n(x) = L_0(x) + [L_1(x) - L_0(x)] + [L_2(x) - L_1(x)] + \dots + [L_n(x) - L_{n-1}(x)]$$

$$f(x_k) - L_{k-1}(x_k) = A(x_k - x_0) \cdot \dots \cdot (x_k - x_{k-1}).$$

Підставивши цей вираз і вираз для A у вираз для $L_n(x)$, одержимо:

$$L_n(x) = f_0(x) + f(x_0; x_1)(x - x_0) + f(x_0; x_1; x_2)(x - x_0)(x - x_1) + \dots$$

$$f(x_0; x_1; \dots; x_n)(x - x_0)(x - x_1) \dots (x - x_{n-1})$$

Підсумок виведення інтерполяційного многочлена Ньютона

Отриманий вираз для полінома Лагранжа одержав назву *інтерполяційного полінома Ньютона для нерівновіддалених вузлів*:

$$N_n(x) = f_0(x) + f(x_0; x_1)(x - x_0) + f(x_0; x_1; x_2)(x - x_0)(x - x_1) + \dots$$

$$f(x_0; x_1; \dots; x_n)(x - x_0)(x - x_1) \dots (x - x_{n-1})$$

Перевага многочлена Ньютона

Додавання чергового вузла не приводить до переобчислення всіх попередніх вузлів [24].

3.5.4. Похибка інтерполяційного многочлена Ньютона

Залишковий член такий же, як і в полінома Лагранжа, але його записують у іншій формі:

$$R_n(x) = f(x) - N_n(x) = (x - x_0)(x - x_1)\dots(x - x_n)f(x; x_0; x_1; \dots; x_n)$$

Якщо $f(x)$ має похідну $(n+1)$ порядку, то

$$f(x; x_0; x_1; \dots; x_n) = \frac{f^{(n+1)}(\xi)}{(n+1)!}.$$

$$\text{Звідси } R_n(x) = (x - x_0)(x - x_1)\dots(x - x_n) \frac{f^{(n+1)}(\xi)}{(n+1)!}$$

Розглянемо приклад побудови й застосування многочлена Ньютона.

Приклад 3.9. Нехай функція $y = f(x)$ задана таблицею.

Таблиця 3.5

Таблиця значень функції

i	0	1	2	3
x_i	1,00	1,03	1,05	1,09
y_i	1,00	1,015	1,034	1,044

Потрібно побудувати многочлен Ньютона та за його допомоги обчислити $f(1,08)$.

Розв'язок

Етап 1. Формуємо розділені різниці:

$$\frac{y_1 - y_0}{x_1 - x_0} = \frac{1,015 - 1,00}{1,03 - 1,00} = 0,5 = y_{10},$$

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{1,034 - 1,015}{1,05 - 1,03} = 0,95 = y_{21},$$

$$\frac{y_3 - y_2}{x_3 - x_2} = \frac{1,044 - 1,034}{1,09 - 1,05} = 0,25 = y_{32}.$$

$$\frac{y_{21} - y_{10}}{x_2 - x_0} = \frac{0,95 - 0,5}{1,05 - 1,0} = 9 = y_{210},$$

$$\frac{y_{32} - y_{21}}{x_3 - x_1} = \frac{0,25 - 0,95}{1,09 - 1,03} \approx -11,666 = y_{321},$$

$$\frac{y_{321} - y_{210}}{x_3 - x_0} = \frac{-11,666 - 9}{1,09 - 1,00} = -229,63 = y_{3210}.$$

Етап 2. Використовуючи формулу

$$\begin{aligned} N_n(x) = & f(x_0) + f(x_0; x_1)(x - x_0) + \\ & + f(x_0; x_1; x_2)(x - x_0)(x - x_1) + \dots \\ & \dots + f(x_0; x_1; \dots; x_n)(x - x_0)(x - x_1) \dots (x - x_{n-1}). \end{aligned}$$

знаходимо многочлен Ньютона:

$$\begin{aligned} N_3(x) = & y_0 + y_{10}(x - x_0) + y_{210}(x - x_0)(x - x_1) + \\ & + y_{3210}(x - x_0)(x - x_1)(x - x_2) \\ y = & 1 + 0,5(x - 1,00) + 9(x - 1,00)(x - 1,03) - \\ & - 229,63(x - 1,00)(x - 1,03)(x - 1,05) \end{aligned}$$

Етап 3. Підставимо в даний многочлен задане значення аргументу й одержимо:

$$\begin{aligned} y = f(1,08) = & 1,00 + 0,5(1,08 - 1,00) + 9(1,08 - 1,00)(1,08 - 1,03) - \\ & - 229,63(1,08 - 1,00)(1,08 - 1,03)(1,08 - 1,05) \approx 1,041 \end{aligned}$$

3.5.5. Поняття про скінченні різниці

Нехай є набір рівновіддалених точок x_0, x_1, \dots, x_n , $x_i \neq x_j$ при $i \neq j$, $x_i \in [a, b]$, $x_{i+1} - x_i = h = const$, на якому деяка функція $f(x) \in R$ приймає значення $f(x_0), f(x_1), \dots, f(x_n)$.

Вирази виду

$$\Delta f(x_0) = f(x_1) - f(x_0),$$

$$\Delta f(x_1) = f(x_2) - f(x_1),$$

...

$$\Delta f(x_{n-1}) = f(x_n) - f(x_{n-1})$$

або

$$\Delta f(x_i) = f(x_{i+1}) - f(x_i), \quad i = 0, 1, 2, \dots, n-1$$

називають *скінченною різницею першого порядку*.

Вирази

$$\Delta^2 f(x_0) = \Delta(\Delta f_0) = \Delta f(x_1) - \Delta f(x_0) = f(x_2) - 2f(x_1) + f(x_0),$$

$$\Delta^2 f(x_1) = \Delta(\Delta f_1) = \Delta f(x_2) - \Delta f(x_1) = f(x_3) - 2f(x_2) + f(x_1),$$

...

$$\Delta^2 f(x_{n-2}) = \Delta(\Delta f_{n-2}) = \Delta f(x_{n-1}) - \Delta f(x_{n-2}) = f(x_n) - 2f(x_{n-1}) + f(x_{n-2}),$$

або

$$\Delta^2 f(x_i) = \Delta(\Delta f_i) = \Delta f(x_{i+1}) - \Delta f(x_i) = f(x_{i+2}) - 2f(x_{i+1}) + f(x_i),$$

де $i = 0, 1, \dots, n-2$.

називають *скінченною різницею другого порядку*.

Різницями k -го порядку називають відношення

$$\Delta^k f(x_0) = \Delta(\Delta^{k-1} f(x_0)) = \sum_{j=0}^k (-1)^j C_k^j f(x_j),$$

$$\Delta^k f(x_1) = \Delta(\Delta^{k-1} f(x_1)) = \sum_{j=0}^k (-1)^j C_k^j f(x_{j+1}),$$

...

$$\Delta^k f(x_{n-k}) = \Delta(\Delta^{k-1} f(x_{n-k})) = \sum_{j=0}^k (-1)^j C_k^j f(x_{n-k+j}),$$

$$\text{або } \Delta^k f(x_i) = \Delta(\Delta^{k-1} f(x_i)) = \sum_{j=0}^k (-1)^j C_k^j f(x_{i+j})$$

$$i = 0, 1, 2, \dots, n - k, C_k^j = \frac{k!}{j!(k-j)!}$$

Скінченну різницю n -го порядку визначають виразом

$$\Delta^n f(x_0) = \Delta(\Delta^{n-1} f(x_0)) = \sum_{j=0}^n (-1)^j C_n^j f(x_0)$$

Розділені й скінченні різниці

При постійному кроці h розділені й скінченні різниці зв'язані співвідношенням

$$f(x_i; \dots; x_{i+k}) = \frac{\Delta^k f_i}{k! h^k}, \quad i = 0, 1, \dots, n - k$$

3.5.6. Інтерполяційний многочлен Ньютона для рівновіддалених вузлів

Постановка задачі. Нехай для функції $y = f(x)$ задані значення $y_i = f(x_i)$ у рівновіддалених $(n + 1)$ вузлах інтерполяції,

$$y_0 = f(x_0), y_1 = f(x_1), y_2 = f(x_2), \dots, y_n = f(x_n).$$

Потрібно побудувати многочлен $N_n(x)$ зі степенем не вище n , який приймає в заданих вузлах $x_i, i = 0, 1, 2, \dots, n$ значення, що збігаються зі значеннями $y_i, N_n(x_i) = y_i, i = 0, 1, 2, \dots, n$.

Перший інтерполяційний многочлен Ньютона

Перший інтерполяційний многочлен Ньютона можна отримати за допомогою підстановки в многочлен

$$\begin{aligned} N_n(x) = & f(x_0) + f(x_0; x_1)(x - x_0) + \\ & + f(x_0; x_1; x_2)(x - x_0)(x - x_1) + \dots \\ & \dots + f(x_0; x_1; \dots; x_n)(x - x_0)(x - x_1) \dots (x - x_{n-1}) \end{aligned}$$

замість розділених різниць їх вирази через скінченні різниці:

$$f(x_i; \dots; x_{i+k}) = \frac{\Delta^k f_i}{k! h^k}, \quad i = 0, 1, \dots, n-k$$

Уведемо заміну $d = \frac{x - x_0}{h}$ – число кроків, необхідних для досягнення точки x , виходячи з x_0 , названої фазою інтерполяції, визначеною відносно точки x_n .

Визначимо значення доданків першого многочлена Ньютона:

$$y_{10}(x - x_0) = \frac{\Delta y_0 (x - x_0)}{1! h} = d \Delta y_0,$$

$$y_{210}(x - x_0)(x - x_1) = \frac{\Delta^2 y_0}{2! h^2} (x - x_0)(x - x_0 - h) = \frac{d(d-1)}{2!} \Delta^2 y_0$$

$$\begin{aligned} y_{3210}(x - x_0)(x - x_1)(x - x_2) &= \\ &= \frac{\Delta^3 y_0}{3! h^3} (x - x_0)(x - x_0 - h)(x - x_0 - 2h) = \frac{d(d-1)(d-2)}{3!} \Delta^3 y_0 \end{aligned}$$

$$\begin{aligned} y_{n\dots 0}(x - x_0) \cdot \dots \cdot (x - x_{n-1}) &= \\ &= \frac{\Delta^n y_0}{n! h^n} (x - x_0)(x - x_0 - h) \dots (x - x_0 - (n-1)h) = \frac{d(d-1)(d-n+1)}{n!} \Delta^n y_0 \end{aligned}$$

$$\begin{aligned} N_n(x) &= y_0 + d \Delta y_0 + \frac{d(d-1)}{2!} \Delta^2 y_0 + \frac{d(d-1)(d-2)}{3!} \Delta^3 y_0 + \dots \\ &\quad \dots + \frac{d(d-1) \dots (d-n+1)}{n!} \Delta^n y_0, \end{aligned}$$

Другий інтерполяційний многочлен Ньютона

Введемо заміну $d = \frac{x - x_n}{h}$. Тоді $\frac{x - x_{n-1}}{h} = \frac{x - x_n + h}{h} = d + 1$,

$$\frac{x - x_{n-2}}{h} = \frac{x - x_n + 2h}{h} = d + 2$$

Підставимо в многочлен:

$$N_n(x) = y_0 + \frac{\Delta y_0}{1!h}(x-x_0) + \frac{\Delta^2 y_0}{2!h^2}(x-x_0)(x-x_1) + \dots + \frac{\Delta^3 y_0}{3!h^3}(x-x_0)(x-x_1)(x-x_2) + \dots + \frac{\Delta^n y_0}{n!h^n}(x-x_0)(x-x_1)\dots(x-x_{n-1}),$$

У результаті отримаємо аналогічно:

$$N_n(x) = y_n + d\Delta y_{n-1} + \frac{d(d+1)}{2!}\Delta^2 y_{n-2} + \dots + \frac{d(d+1)(d+2)}{3!}\Delta^3 y_{n-3} + \dots + \frac{d(d+1)\dots(d+n-1)}{n!}\Delta^n y_0,$$

3.5.7. Похибка інтерполяційного полінома Ньютона з рівновіддаленими вузлами

Залишковий член може бути отриманий з попередньої формули для залишкового члена полінома Ньютона з нерівновіддаленими вузлами.

$$R_n(x) = (x-x_0)(x-x_0-h)\dots(x-x_0-nh) \frac{f^{(n+1)}(\xi)}{(n+1)!} = \frac{h^{n+1} f^{(n+1)}(\xi)}{(n+1)!} m(m-1)(m-2)\dots(m-n).$$

де $x-x_0 = mh$.

Або для другого многочлена аналогічно

$$R_n(x) = \frac{h^{n+1} f^{(n+1)}(\xi)}{(n+1)!} m(m+1)\dots(m+n) \text{ при } x-x_n = mh$$

Контрольні завдання

1. Скласти розділені різниці першого порядку для функції

$y = f(x)$, яка задана таблицею, та обчислити їх значення.

$x_0=1$	$x_1=3$	$x_2=7$	$x_3=12$	$x_4=22$
$y_0=1$	$y_1=5$	$y_2=9$	$y_3=29$	$y_4=69$

2. Скласти розділені різниці першого порядку для функції $Y=f(X)$ при $Y=(y_i)_{i=0}^4 = \{1,5,9,15,21\}$, $X=\{x_i\}_{i=0}^4 = \{1,2,6,8,10\}$.

3. Нехай відомі значення функції $Y=f(X)$ у точках: $Y=(y_i)_{i=0}^2 = \{2,6,16\}$, $X=\{x_i\}_{i=0}^2 = \{1,3,5\}$. За допомогою полінома Ньютона знайти значення функції в точці $x^* = 4$.

4. Функція представлена рівновіддаленими вузлами x_i , де $i = 0,1,2,3,4 \dots$. Відомі її початкове значення $f(x_0) = 0$, де $x_0 = 0$ й скінченні різниці: $\Delta f(x_0) = 4$, $\Delta^2 f(x_0) = 8$. Знайти значення функції в точці $x = 3$ за умови, що $x_{i+1} - x_i = 2$.

5. Скласти скінченні різниці другого порядку для функції $Y=f(X)$ при $Y=(y_i)_{i=0}^4 = \{1,4,8,9,10\}$, $X=\{x_i\}_{i=0}^4 = \{1,2,6,8,16\}$.

3.6. Сплайн-інтерполяція

Кубічні сплайн-функції є деякими математичними моделями гнучкого тонкого стрижня із пружного матеріалу, закріпленого в точках (сусідніх вузлах інтерполяції) (рис. 3.2). Між точками закріплення цей стрижень прийме деяку форму, яка мінімізує його потенційну енергію. У загальному випадку сплайн задають глобальним способом, тобто з використанням усіх вузлів при будь-якому їхньому розташуванні [25].

Нехай на відрізку $[a, b]$ задана неперервна функція $f(x)$.

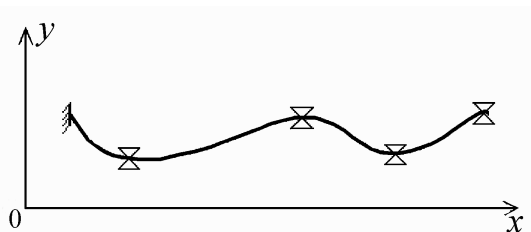


Рис. 3.2. Локальний кубічний сплайн

Введемо розбивку відрізка:

$$a = x_0 < x_1 < x_2 < \dots < x_{i-1} < x_i < \dots < x_{n-1} < x_n = b$$

Нехай для функції $y = f(x)$ задані значення $y_i = f(x_i)$ у $(n+1)$ вузлах інтерполяції.

Сплайном, який відповідає даній функції $f(x)$ й вузлам інтерполяції $x_i, i=0, \dots, n$ будемо називати деяку функцію $s_i(x)$, що задовольняє умови:

1) на кожному відрізку $[x_{i-1}, x_i]$ $i=1, 2, \dots, n$ функція $s_i(x)$ є кубічним многочленом,

2) $s_i(x)$, а також $s'_i(x)$ і $s''_i(x)$ – неперервні на відрізку $[a, b]$:

$$\begin{aligned} s_i(x_i - 0) &= s_{i+1}(x_i + 0), \quad s'_i(x_i - 0) = s'_{i+1}(x_i + 0), \\ s''_i(x_i - 0) &= s''_{i+1}(x_i + 0), \quad i=1, \dots, n-1 \end{aligned}$$

3) $s_{i+1}(x_i) = f(x_i) = y_i, i=0, 1, \dots, n$ – умова інтерполяції,

4) $s''_1(x_0) = 0, s''_n(x_n) = 0$.

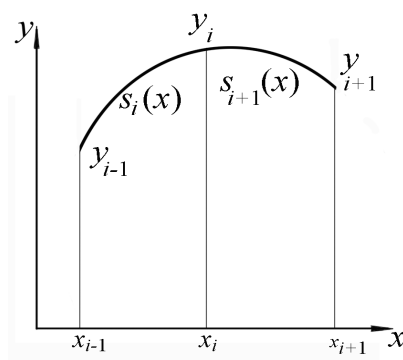


Рис. 3.3. Сплайн-інтерполяція

На кожному відрізку $[x_{i-1}, x_i]$, $i=1, 2, \dots, n$ будемо шукати сплайн-функцію $s(x) = s_i(x)$ у вигляді полінома третього степеня:

$$s_i(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3, \quad x_{i-1} \leq x \leq x_i, \quad i=1, 2, \dots, n, \quad (1)$$

Крім того, у точці x_{i-1} :

$s_i(x_{i-1}) = a_i = y_{i-1}$ – умова зшивки сплайнів у вузлах. Введемо

позначення: $h_i = x_i - x_{i-1}$.

Тоді

$$s_i(x_i) = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = y_i, \quad i = 1, \dots, n-1, \quad (3.19)$$

де a_i, b_i, c_i, d_i – шукані коефіцієнти.

Продиференціюємо даний вираз (3.19) по x :

$$s'_i(x) = b_i + 2c_i(x - x_{i-1}) + 3d_i(x - x_{i-1})^2, \quad x \in [x_{i-1}, x_i];$$

$$s'_{i+1}(x) = b_{i+1} + 2c_{i+1}(x - x_i) + 3d_{i+1}(x - x_i)^2, \quad x \in [x_i, x_{i+1}];$$

Використовуючи вимогу 2) неперервності перших похідних у вузлах інтерполяції, одержимо:

$$b_i + 2c_i h_i + 3d_i h_i^2 = b_{i+1}; \quad i = 1, \dots, n-1. \quad (3.20)$$

Обчислимо другі похідні на суміжних відрізках:

$$s''_i(x) = 2c_i + 6d_i(x - x_{i-1}), \quad x \in [x_{i-1}, x_i];$$

$$s''_{i+1}(x) = 2c_{i+1} + 6d_{i+1}(x - x_i), \quad x \in [x_i, x_{i+1}];$$

Використовуючи вимогу 2) неперервності перших похідних у вузлах інтерполяції одержимо:

$$2c_i + 6d_i h_i = 2c_{i+1}; \quad i = 1, \dots, n-1. \quad (3.21)$$

Або після скорочення: $c_i + 3d_i h_i = c_{i+1}; \quad i = 1, \dots, n-1.$

Із граничних умов 4) $s''(x_0) = 0, s''(x_n) = 0$ одержимо, що

$$c_1 = 0; \quad c_n + 3d_n h_n = 0. \quad (3.22)$$

Співвідношення (3.19–3.22) утворюють систему $4n$ лінійних алгебраїчних рівнянь відносно $4n$ невідомих:

$$\begin{cases} a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = y_i, \\ b_i + 2c_i h_i + 3d_i h_i^2 = b_{i+1}, \\ 2c_i + 6d_i h_i = 2c_{i+1}, \\ c_1 = 0; \quad c_n + 3d_n h_n = 0, \end{cases} \quad i = 1, \dots, n-1$$

Із третього рівняння знайдемо d_i й підставимо в перше:

$$d_i = \frac{c_{i+1} - c_i}{3h_i}; \quad y_i = a_i + b_i h_i + c_i h_i^2 + \frac{c_{i+1} - c_i}{3} h_i^2, \quad i = 1, \dots, n-1 \quad (3.23)$$

Знову ж з умов неперервності $y_{i-1} = a_i$, $i = 1, \dots, n$.

$$\text{Тоді} \quad y_i = y_{i-1} + b_i h_i + c_i h_i^2 + \frac{c_{i+1} - c_i}{3} h_i^2, \quad i = 1, \dots, n-1.$$

$$y_i = y_{i-1} + b_i h_i + c_i h_i^2 + \frac{c_{i+1} - c_i}{3} h_i^2, \quad i = 1, \dots, n-1$$

Виразимо із цього співвідношення b_i :

$$b_i = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{3} (c_{i+1} + 2c_i); \quad i = 1, \dots, n-1 \quad (3.24)$$

Підставимо тепер значення

$$b_i = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{3} (c_{i+1} + 2c_i), \quad b_i + 2c_i h_i + 3d_i h_i^2 = b_{i+1};$$

$$b_{i+1} = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{h_{i+1}}{3} (c_{i+2} + 2c_{i+1}) \quad \text{і} \quad d_i = \frac{c_{i+1} - c_i}{3h_i}$$

у друге рівняння системи:

$$\begin{aligned} & \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{3} (c_{i+1} + 2c_i) + 2h_i c_i + 3h_i^2 \frac{c_{i+1} - c_i}{3h_i} = \\ & = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{h_{i+1}}{3} (c_{i+2} + 2c_{i+1}), \quad i = 1, \dots, n-1 \end{aligned}$$

Домножимо обидві частини співвідношення на 3, згрупуємо доданки й зведемо подібні члени:

$$\begin{aligned} 3 \left[\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i} \right] &= -h_i c_{i+1} - 2h_i c_i + 6h_i c_i + 3h_i c_{i+1} - 3h_i c_i + h_{i+1} c_{i+2} + 2h_{i+1} c_{i+1}, \\ 3 \left[\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i} \right] &= h_i c_i + 2(h_i + h_{i+1}) c_{i+1} + h_{i+1} c_{i+2}, \quad (3.25) \\ & i = 1, \dots, n-1 \end{aligned}$$

Одержали систему $(n-1)$ лінійних алгебраїчних рівнянь відносно $(n+1)$ невідомих c_1, c_2, \dots, c_{n+1} .

Для однозначного визначення всіх невідомих скористаємося граничними умовами:

$$c_1 = 0, c_{n+1} = c_n + 3d_n h_n = 0,$$

Отриману систему (3.25) представимо у вигляді

$$3 \left[\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i} \right] = h_i c_i + 2(h_i + h_{i+1})c_{i+1} + h_{i+1}c_{i+2}, \quad i = 1, \dots, n-1$$

Обчисливши коефіцієнти c_2, \dots, c_n

$$1) \text{ знаходимо з (3.23) } d_i = \frac{c_{i+1} - c_i}{3h_i},$$

$$2) \text{ знаходимо з (3.22) } d_n = -c_n / 3h_n,$$

$$3) \text{ знаходимо з (3.24) } b_i = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{3}(c_{i+1} + 2c_i)$$

$$b_n = \frac{y_n - y_{n-1}}{h_n} - \frac{2h_n c_n}{3}, \text{ оскільки } c_{n+1} = 0.$$

Коефіцієнти $a_i, i = 1, \dots, n$ відомі з умови зшивки $a_i = y_{i-1}$.

Таким чином, існує й знайдений єдиний кубічний сплайн, оскільки однозначно визначені a_i, b_i, c_i, d_i для інтервалу $[x_{i-1}, x_i]$.

3.6.1. Послідовність обчислень функції $f(x)$ методом сплайн-інтерполяції

1. За описаною методикою обчислюють коефіцієнти $a_i, b_i, c_i, d_i; i = 1, \dots, n$ кубічних сплайнів.

2. Шукають інтервал $[x_{i-1}, x_i]$, якому належить дане x . Значення функції $f(x)$ на цьому інтервалі обчислюють з кубічного сплайна:

$$f(x) = a_i + b_i(x - x_{i-1}) + c_i(x - x_{i-1})^2 + d_i(x - x_{i-1})^3$$

з параметрами a_i, b_i, c_i, d_i для інтервалу $[x_{i-1}, x_i]$.

Контрольні завдання

1. Яким умовам повинна задовольняти функція сплайна?
2. Визначте послідовність обчислень значень функції методом сплайн-інтерполяції.
3. Побудувати сплайн-інтерполяційний многочлен для функції, заданої таблично:

x	1	2	3	4	5
$f(x)$	1	1.1	0.6	0.5	0.2

4. Знайти значення функції в точці $x = 1.2$ методом сплайн-інтерполяції, якщо відомі її значення в точках:

$$f(0) = 1, f(1) = 1.1, f(2) = 1.3.$$

5. Для функції $y = x^3$, представленої відомими значеннями $f(1) = 1, f(2) = 8, f(3) = 8$, побудувати інтерполяційний поліном методом сплайн-інтерполяції і визначити $f(x)$ при $x = 2.4$.

3.7. Тригонометрична інтерполяція

Тригонометрична інтерполяція полягає в тому, що *тригонометричний многочлен* виду [26]

$$p(x) = a_0 + \sum_{m=1}^n a_m \cos(mx) + \sum_{m=1}^n b_m \sin(mx) \quad (3.26)$$

де n – натуральні числа, *використовують як інтерполяційну функцію*.

Рівняння містить $2n + 1$ невідомих коефіцієнтів:

$$a_0, a_1, \dots, a_n; b_1, b_2, \dots, b_n.$$

Завдання полягає в тому, щоб обчислити ці коефіцієнти за умови, що значення функції $f(x)$ відомі в N точках: $p(x_k) = y_k, k = 1, \dots, N$.

Оскільки тригонометричний поліном періодичний з періодом 2π , доцільно розглядати функцію на періоді:

$$0 \leq x_0 < x_1 < x_2 < \dots < x_N \leq 2\pi$$

3.7.1. Розв'язування задачі тригонометричної інтерполяції

1. При $N \leq 2n + 1$ задача має розв'язок для будь-якої множини впорядкованих пар $(x_k, p(x_k))$

2. При $N > 2n + 1$ розв'язку іноді може не існувати.

3. При $N = 2n + 1$ задача завжди має єдиний розв'язок.

Розв'язок даної задачі може бути представлено у формі полінома Лагранжа:

$$p(x) = \sum_{k=1}^{2n+1} y_k \prod_{m=1, m \neq k}^{2n+1} \frac{\sin \frac{x - x_m}{2}}{\sin \frac{x_k - x_m}{2}}$$

Існує теорема, що доводить еквівалентність даного виразу й виразу (3.26), що підтверджує дотримання даним поліномом умови інтерполяції $p(x_i) = f(x_i) = f_i, i = 1, 2, \dots$

Пошук коефіцієнтів тригонометричного полінома

$$p(x) = a_0 + \sum_{m=1}^n a_m \cos(mx) + \sum_{m=1}^n b_m \sin(mx)$$

виконують методом невизначених коефіцієнтів, що приводить до системи:

$$\begin{pmatrix} 1 & \cos x_1 & \sin x_1 \dots & \cos nx_1 & \sin nx_1 \\ 1 & \cos x_2 & \sin x_2 \dots & \cos nx_2 & \sin nx_2 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & \cos x_{2n+1} & \sin x_{2n+1} \dots & \cos nx_{2n+1} & \sin nx_{2n+1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \dots \\ b_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \dots \\ y_{2n+1} \end{pmatrix}$$

Ця система однозначно розв'язна відносно $2n + 1$ коефіцієнта:

$$a_0, a_1, \dots, a_n; b_1, b_2, \dots, b_n$$

3.7.2. Формули коефіцієнтів для рівновіддалених вузлів

Якщо взяти рівновіддалені вузли інтерполяції:

$$x_j = \frac{2(j-1)\pi}{2n+1} \quad \text{при } j \in \{1, 2, \dots, 2n+1\}$$

то коефіцієнти знайдемо по формулах:

$$a_0 = \frac{1}{2n+1} \sum_{j=1}^{2n+1} y_j, \quad a_k = \frac{2}{2n+1} \sum_{j=1}^{2n+1} y_j \cos(kx_j), \quad k = 1, \dots, n,$$

$$b_k = \frac{2}{2n+1} \sum_{j=1}^{2n+1} y_j \sin(kx_j), \quad k = 1, 2, \dots, n$$

Розглянемо нумерацію вузлів інтерполяції:

$$x_j = \frac{2\pi j}{2n+1} \quad \text{при } j \in \{-n, (-n+1), \dots, -1, 0, 1, 2, \dots, n\}.$$

Коефіцієнти на інтервалі $[-\pi, \pi]$ знайдемо по формулах:

$$a_0 = \frac{1}{2n+1} \sum_{j=-n}^n y_j, \quad a_k = \frac{2}{2n+1} \sum_{j=-n}^n y_j \cos(kx_j), \quad k = 1, \dots, n$$

$$b_k = \frac{2}{2n+1} \sum_{j=-n}^n y_j \sin(kx_j), \quad k = 1, 2, \dots, n$$

Приклад 3.10. Тригонометрична інтерполяція

Нехай функція, що описує деякий коливальний процес, задана таблицею.

Таблиця 3.6

Таблиця значень функції

x_j	0	$\frac{\pi}{3}$	$\frac{2\pi}{3}$	π	$\frac{4\pi}{3}$	$\frac{5\pi}{3}$	2π
$f(x_j)$	-2	-0,92	0,83	2	2,32	-1,11	-2

Потрібно побудувати тригонометричний многочлен другого степеня.

$$p(x) = a_0 + \sum_{m=1}^3 a_m \cos(mx) + \sum_{m=1}^3 b_m \sin(mx)$$

Розв'язок. Визначимо вузли інтерполяції з формули:

$$x_j = \frac{2(j-1)\pi}{2 \cdot 3} \quad \text{при } j \in \{1, 2, 3, 4, 5, 6, 7\}$$

$$x_1 = 0, x_2 = \frac{\pi}{3}, x_3 = \frac{2\pi}{3}, x_4 = \pi, x_5 = \frac{4}{3}\pi, x_6 = \frac{5\pi}{3}, x_7 = 2\pi$$

$$x_1 = 0^\circ, x_2 = 60^\circ, x_3 = 120^\circ, x_4 = 180^\circ, x_5 = 240^\circ, x_6 = 300^\circ, x_7 = 360^\circ,$$

Функція $f(x)$ визначена на дискретній множині рівновіддалених точок із кроком $h = \frac{\pi}{3} = 60^\circ$.

Запишемо вирази для коефіцієнтів a_0, a_1, a_2, b_1, b_2 :

$$a_0 = \frac{1}{2 \cdot 3 + 1} \sum_{j=1}^{2 \cdot 3 + 1} y_j, \quad a_0 = \frac{1}{7} (y_1 + y_2 + y_3 + y_4 + y_5 + y_6 + y_7);$$

$$a_1 = \frac{2}{2 \cdot 3 + 1} \sum_{j=1}^{2 \cdot 3 + 1} y_j \cos(x_j), \quad a_2 = \frac{2}{2 \cdot 3 + 1} \sum_{j=1}^{2 \cdot 3 + 1} y_j \cos(2x_j), \quad a_3 = \frac{2}{2 \cdot 3 + 1} \sum_{j=1}^{2 \cdot 3 + 1} y_j \cos(3x_j),$$

$$a_1 = \frac{2}{7} (y_1 \cos x_1 + y_2 \cos x_2 + y_3 \cos x_3 + y_4 \cos x_4 + y_5 \cos x_5 + y_6 \cos x_6 + y_7 \cos x_7)$$

$$b_1 = \frac{2}{2 \cdot 3 + 1} \sum_{j=1}^{2 \cdot 3 + 1} y_j \sin(x_j), \quad b_2 = \frac{2}{2 \cdot 3 + 1} \sum_{j=1}^{2 \cdot 3 + 1} y_j \sin(2x_j), \quad b_3 = \frac{2}{2 \cdot 3 + 1} \sum_{j=1}^{2 \cdot 3 + 1} y_j \sin(3x_j),$$

$$b_1 = \frac{2}{7} (y_1 \sin x_1 + y_2 \sin x_2 + y_3 \sin x_3 + y_4 \sin x_4 + y_5 \sin x_5 + y_6 \sin x_6 + y_7 \sin x_7);$$

Контрольні запитання

1. Запишіть загальний вигляд тригонометричного полінома для функції, яка проходить через 4 точки.

2. За яких умов задача побудови тригонометричного полінома має єдиний розв'язок?

3. Запишіть загальну формулу тригонометричного полінома.

4. Для функції, яка має значення: $f(0) = -2$, $f(\pi/2) = 3$, $f(\pi) = 1$, побудувати тригонометричний поліном та визначити її значення в точці $f(\pi/4)$.

5. Для функції, яка має значення: $f(0) = -1$, $f(\pi) = 1$, $f(\pi) = -1$, побудувати тригонометричний поліном та визначити її значення в точці $f(\pi/2)$.

3.8. Чисельне диференціювання

3.8.1. Постановка задачі

При розв'язуванні практичних задач часто доводиться обчислювати похідні функції $y = f(x)$ різних порядків. Якщо функція задана аналітично і її аналітичний вираз не занадто складний, то задача вирішується звичайними методами математичного аналізу [27].

До чисельного диференціювання звертаються тоді, коли функція задана таблицею або коли залежність $y = f(x)$ представлена складним аналітичним виразом [28]. У першому випадку методи математичного аналізу незастосовні, а в другому – обчислення похідних пов'язане зі значними труднощами. У цих випадках зазвичай проводиться заміна даної функції, найчастіше, інтерполяційним поліномом, користуючись виразом:

$$f(x) = P_n(x) + R_n(x) \quad (3.27)$$

де $P_n(x)$ – інтерполяційний поліном,

$R_n(x)$ – залишковий член інтерполяційної формули.

Нехай функція $f(x)$ має похідні аж до порядку k включно. Тоді диференціюємо вираз (3.27) і знаходимо похідні:

$$\begin{aligned} f'(x) &= P'_n(x) + R'_n(x) \\ f''(x) &= P''_n(x) + R''_n(x) \\ &\dots\dots\dots \\ f^{(k)}(x) &= P_n^{(k)}(x) + R_n^{(k)}(x) \end{aligned} \quad (3.28)$$

Наближені значення цих похідних обчислюють як перші доданки у правій частині рівнянь:

$$\begin{aligned} f'(x) &\approx P'_n(x) \\ f''(x) &\approx P''_n(x) \\ &\dots\dots\dots \\ f^{(k)}(x) &\approx P_n^{(k)}(x) \end{aligned} \quad (3.29)$$

Залишкові члени $R'_n(x), R''_n(x), \dots, R_n^{(k)}(x)$ виражають похибки цих наближених рівнянь.

При заміні функції $f(x)$ інтерполяційним многочленом $P_n(x)$ необхідно мати на увазі, що залишковий член $R_n(x)$ повинен бути досить малим, але із цього *не випливає той факт*, що $R'_n(x), R''_n(x), \dots, R_n^{(k)}(x)$ будуть досить малими. Похибки, одержувані при обчисленні похідних (особливо вищих порядків), можуть виявитися значними.

3.8.2. Застосування формули чисельного диференціювання на основі інтерполяційного полінома Ньютона з нерівновіддаленими вузлами

Будемо використовувати для чисельного диференціювання поліном Ньютона для нерівновіддалених вузлів. Але будемо вважати, що функція задана у певних вузлах:

$$x_i = x_0 + ih, \quad h > 0, \quad i = 0, \pm 1, \pm 2, \dots$$

Значення функції та значення її похідних у вузлах будемо позначати так: $f(x_i) = f_i, f'(x_i) = f'_i, f''(x_i) = f''_i, \dots$

Нехай функція задана у двох точках x_0 і $x_1 = x_0 + h$, а її значення f_0, f_1 .

Побудуємо інтерполяційний поліном першого степеня:

$$N_1(x) = f_0 + (x - x_0)f(x_0; x_1)$$

Похідна $N'_1(x)$ дорівнює $N'_1(x) = f(x_0; x_1) = \frac{f_1 - f_0}{h}$.

Наступний крок полягає у заміні похідної від функції $f(x)$ у точці x_0 відповідною похідною полінома. Отже,

$$f'_0(x) \approx \frac{f_1 - f_0}{h} \tag{3.30}$$

Величину $\frac{f_1 - f_0}{h}$ називають першою різницевою похідною.

Нехай тепер функція $f(x)$ задана в трьох точках:

$$x_0, x_1 = x_0 + h, x_{-1} = x_0 - h$$

Інтерполяційний поліном Ньютона другого степеня має вигляд:

$$N_2(x) = f(x_0) + (x - x_0)f(x_0; x_1) + (x - x_0)(x - x_1)f(x_0; x_1; x_{-1}).$$

Похідна $N_2'(x)$ дорівнює $N_2'(x) = f(x_0; x_1) + (2x - x_0 - x_1)f(x_0; x_1; x_{-1})$

У точці x_0 вона дорівнює

$$N_2'(x_0) = \frac{f_1 - f_0}{x_1 - x_0} + (x_0 - x_1) \left[\frac{f_0}{(x_0 - x_1)(x_0 - x_{-1})} + \frac{f_1}{(x_1 - x_0)(x_1 - x_{-1})} + \frac{f_{-1}}{(x_{-1} - x_0)(x_{-1} - x_1)} \right] = \frac{f_1 - f_{-1}}{2h}$$

Одержуємо наближену формулу:

$$f_0' \approx \frac{f_1 - f_{-1}}{2h} \quad (3.31)$$

Величину $\frac{f_1 - f_{-1}}{2h}$ називають центральною різницевою похідною.

Розглянемо другу похідну для трьох точок: $x_0, x_1 = x_0 + h, x_{-1} = x_0 - h$

$$\begin{aligned} N_2''(x) &= 2f(x_0; x_1; x_{-1}) = \\ &= 2 \left[\frac{f_0}{(x_0 - x_1)(x_0 - x_{-1})} + \frac{f_1}{(x_1 - x_0)(x_1 - x_{-1})} + \frac{f_{-1}}{(x_{-1} - x_0)(x_{-1} - x_1)} \right] = \\ &= \frac{f_1 - 2f_0 + f_{-1}}{h^2} \end{aligned}$$

У результаті одержуємо наближену формулу:

$$f_0'' \approx \frac{f_1 - 2f_0 + f_{-1}}{h^2} \quad (3.32)$$

Величину $f_0'' \approx \frac{f_1 - 2f_0 + f_{-1}}{h^2}$ називають другою різницевою похідною.

Оцінки похибок чисельного диференціювання за формулами (3.30), (3.31) і (3.32) визначаються відповідними виразами:

$$\left| f_0' - \frac{f_1 - f_0}{h} \right| \leq \frac{h}{2} \max_{[x_0, x_1]} |f''(x)|$$

$$\left| f'_0 - \frac{f_1 - f_{-1}}{2h} \right| \leq \frac{h^2}{6} \max_{[x_{-1}, x_1]} |f'''(x)|$$

$$\left| f''_0 - \frac{f_{-1} - 2f_0 + f_1}{h^2} \right| \leq \frac{h^2}{12} \max_{[x_{-1}, x_1]} |f^{IV}(x)|$$

3.8.3. Застосування формули чисельного диференціювання на основі інтерполяційного полінома Ньютона з рівновіддаленими вузлами

Нехай функція $f(x)$ задана на відрізку $[a, b]$ таблицею значеннями в $(n + 1)$ рівновіддалених вузлах:

Таблиця 3.7

Таблиця значень функції

x_i	x_0	x_1	x_2	\dots	x_n
y_i	y_0	y_1	y_2	\dots	y_n

де $x_i = x_0 + ih$ при $i = 0, 1, 2, \dots, n$.

Необхідно обчислити значення похідної для значень, близьких до x_0 .

Функцію $f(x)$ замінимо наближено першим інтерполяційним поліномом Ньютона:

$$f(x) = y_0 + m\Delta y_0 + \frac{m(m-1)}{2!} \Delta^2 y_0 + \frac{m(m-1)(m-2)}{3!} \Delta^3 y_0 + \frac{m(m-1)(m-2)(m-3)}{4!} \Delta^4 y_0 + \dots \quad (3.33)$$

де $m = \frac{x - x_0}{h}$, $h = x_{i+1} - x_i$.

Розкривши дужки в чисельнику, формулу (3.33) можна записати у вигляді:

$$f(x) = y_0 + m\Delta y_0 + \frac{m^2 - m}{2} \Delta^2 y_0 + \frac{m^3 - 3m^2 + 2m}{6} \Delta^3 y_0 + \frac{m^4 - 6m^3 + 11m^2 - 6m - 3}{24} \Delta^4 y_0 + \dots \quad (3.34)$$

Помітимо, що похідна $\frac{df(x)}{dx} = \frac{df(x)}{dt} \cdot \frac{dt}{dx} = \frac{1}{h} \cdot \frac{df(x)}{dt}$.

Диференціюючи рівність (3.34) двічі, одержимо:

$$f'(x) = \frac{1}{h} \left(\Delta y_0 + \frac{2m-1}{2} \Delta^2 y_0 + \frac{3m^2-6m+2}{6} \Delta^3 y_0 + \frac{2m^3-9m^2+11m-3}{12} \Delta^4 y_0 + \dots \right) \quad (3.35)$$

$$f''(x) = \frac{1}{h^2} \left(\Delta^2 y_0 + (m-1) \Delta^3 y_0 + \frac{6m^2-18m+11}{12} \Delta^4 y_0 + \dots \right) \quad (3.36)$$

У такий же спосіб можна обчислити й похідні будь-якого порядку.

Для того, щоб одержати значення похідних у точці x , що знаходиться наприкінці таблиці (справа), слід скористатися другою інтерполяційною формулою Ньютона. Застосовуючи той же прийом, одержимо:

$$f'(x) = \frac{1}{h} \left(\Delta y_{n-1} + \frac{2m+1}{2} \Delta^2 y_{n-2} + \frac{3m^2+6m+2}{6} \Delta^3 y_{n-3} + \frac{2m^3+9m^2+11m+3}{12} \Delta^4 y_{n-4} + \dots \right) \quad (3.37)$$

$$f''(x) = \frac{1}{h^2} \left(\Delta^2 y_{n-2} + (m+1) \Delta^3 y_{n-3} + \dots \right) \quad (3.38)$$

Формули наближеного диференціювання значно спрощуються, якщо значення похідних обчислюються у вузлах інтерполяції. Поклавши $m = 0$ ($x = x_0$), одержимо:

$$f'(x) = \frac{1}{h} \left(\Delta y_0 - \frac{\Delta^2 y_0}{2} + \frac{\Delta^3 y_0}{3} - \frac{\Delta^4 y_0}{4} + \frac{\Delta^5 y_0}{5} - \dots \right) \quad (3.39)$$

$$f''(x_0) = \frac{1}{h^2} \left(\Delta^2 y_0 - \Delta^3 y_0 + \frac{11}{12} \Delta^4 y_0 - \frac{5}{6} \Delta^5 y_0 + \dots \right) \quad (3.40)$$

Залишковий член при обчисленні першої похідної визначають за формулою:

$$R'_k(x_0) \approx \frac{(-1)^k}{h} \frac{\Delta^{k+1} y_0}{k+1} \quad (3.41)$$

Приклад 3.11. Чисельне диференціювання функції

Знайти $y'(50)$ для функції $y = \lg x$, яка задана таблично.

Таблиця 3.8

Таблиця значень функції та її скінченних різниць

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$
50	1.6990	0.0414	-0.0036	0.0005
55	1.7404	0.0378	-0.0031	
60	1.7782	0.0347		
65	1.8129			

Розв'язок. Виходячи з таблиці $x_{i+1} - x_i = h = 5$. Доповнимо таблицю скінченними різницями. Використавши таблицю та (3.39) запишемо:

$$y'(50) = \frac{1}{h} \left(\Delta y_0 - \frac{\Delta^2 y_0}{2} + \frac{\Delta^3 y_0}{6} \right) = \frac{1}{5} \left(0.0414 - \frac{0.0036}{2} + \frac{0.0005}{6} \right) = 0.2 \cdot (0.0414 + 0.0018 + 0.00008) = 0.0087$$

Оскільки ми обчислювали похідну таблично заданої функції $y = \lg x$,

то $y' = (\lg x)' = \frac{1}{x \cdot \ln 10} = \frac{0.43429}{x}$. Звідки $y'(50) = \frac{0.43429}{50} = 0.0087$.

Результати співпадають з точністю до четвертого десяткового знака.

Приклад 3.11. Нехай $y = f(t)$ – шлях, який пройшла рухома точка за час t . Відповідна функція задана таблично.

Таблиця 3.9

Таблиця значень функції $y = f(t)$

i	Час t_i у сек	Шлях $y(t_i)$ у см
0	0.00	0.000
1	0.01	1.519
2	0.02	6.031
3	0.03	13.397
4	0.04	23.396
5	0.05	35.721
6	0.05	50.000
7	0.07	65.798
8	0.08	82.635

i	Час t_i у сек	Шлях $y(t_i)$ у см
9	0.09	100.000

Використовуючи скінченні різниці до 5-го порядку включно, наближено знайти швидкість $V = \frac{dy}{dt}$ та прискорення $W = \frac{d^2y}{dt^2}$ для моментів часу $t = 0; 0.01; 0.002; 0.03; 0.04$.

Розв'язок. Складемо таблицю скінченних різниць.

Таблиця 3.10

Скінченні різниці функції $y = f(t)$

l	Δy_i	$\Delta^2 y_i$	$\Delta^3 y_i$	$\Delta^4 y_i$	$\Delta^5 y_i$
0	1.519	2.993	-0.139	-0.082	-0.004
1	4.512	2.854	-0.221	-0.086	0.021
2	7.366	2.633	-0.307	-0.065	0.002
3	9.999	2.326	-0.372	-0.063	0.018
4	12.325	1.954	-0.435	-0.045	0.014
5	14.279	1.519	-0.480	-0.031	-
6	15.798	1.039	-0.511	-	
7	16.837	0.528	-		
8	17.365	-			

Прийнявши $h = 0.01$ та застосовуючи формули (3.39) та (3.40) одержимо наближені значення величин швидкості V (см/сек) та величини прискорення W (см/сек²).

Наприклад:

$$V(0) = 100(1.519 - 1.496 - 0.046 + 0.020 - 0.001) = 0.4 \text{ см/сек}$$

$$W(0) = 10000(2.993 + 0.139 - 0.075 + 0.003) = 30600 \text{ см/сек}^2$$

Відповідні значення V та W занесемо в таблицю.

Значення швидкості V і прискорення W для закону руху $y = f(t)$

Таблиця 3.11

Обчислені та точні значення швидкості і прискорення

t	V	W	\tilde{V}	\tilde{W}
0.00	0.4	30600	0.00	30462
0.01	303.6	29780	303.08	30001
0.02	596.3	28780	596.98	28625
0.03	873.2	26250	872.66	26381
0.04	1121.7	23360	1121.9	23340

Представлений таблицею закон руху задано виразом:

$$y = 100 \left(1 - \cos \frac{50\pi t}{9} \right)$$

$$\text{Звідси } V = \frac{dy}{dt} = \frac{5000\pi}{9} \sin \frac{50\pi t}{9} \quad \text{та} \quad W = \frac{d^2y}{dt^2} = \frac{250000\pi^2}{81} \cos \frac{50\pi t}{9}$$

Для порівняння точні значення \tilde{V} і \tilde{W} наведено в таблиці.

Контрольні запитання

1. Шляхом чисельного диференціювання знайти значення похідної $y'(20)$ від функції $y = \ln x$, якщо значення функції відомі тільки в точках $x = 20, 25, 30, 35$.

2. Знайти центральну різницеву похідну $y'(2)$, якщо відомі такі значення функції: $y(1) = 3$, $y(3) = 1$ і $h = 1$.

3. Знайти другу різницеву похідну $y''(2)$, якщо відомі такі значення функції: $y(0) = 0.5$, $y(2) = 2.3$, $y(4) = 6.2$.

4. Запишіть формулу чисельного диференціювання на основі полінома Ньютона, якщо значення похідних обчислюються у вузлах інтерполяції.

5. Запишіть формулу для обчислення залишкового члена при обчисленні похідної за допомогою многочлена Ньютона.

3.9. Методи розв'язування нелінійних рівнянь

3.9.1. Постановка задачі

Нехай потрібно розв'язати рівняння $f(x) = 0$, де $f(x)$ – неперервна функція в скінченному або нескінченному інтервалі.

Розв'язати рівняння – означає знайти всі його корені, тобто ті значення x , у яких $f(x)$ набуває значення нуль, або довести, що кореня не існує [29].

Будь-яке значення $x = x^$, у якому $f(x)$ набуває значення нуль, називають коренем цього рівняння.*

Завдання полягає у відшуванні такого наближеного значення кореня x_{np} , яке мало відрізняється від точного значення кореня x^* , так що виконується нерівність

$$|x^* - x_{np}| < \varepsilon,$$

де ε – мала додатна величина – припустима помилка, яку ми можемо заздалегідь задати на свій розсуд.

Якщо корінь знайдений з точністю ε , то прийнято писати $x^* = x_{np} \pm \varepsilon$.

Припустимо, що рівняння $f(x) = 0$ має лише ізольований корінь.

Визначення. Корінь рівняння називають ізольованим, якщо існує окіл, у якому цей корінь єдиний.

3.9.2. Етапи наближеного розв'язування нелінійних рівнянь

Наближене розв'язування рівняння складається із двох етапів:

1. *Відокремлення кореня*, тобто знаходження інтервалів з області визначення функції $f(x)$, у кожному з яких міститься тільки один корінь рівняння $f(x) = 0$ [30].

2. *Уточнення кореня до заданої точності.*

Відокремлення кореня можна проводити графічно й аналітично.

Існують два способи графічного відокремлення кореня.

Для того, щоб графічно відокремити корінь рівняння $f(x)=0$ у перший спосіб, необхідно побудувати графік функції $y=f(x)$. Абсциси точок його перетину з віссю Ox є дійсними коренями рівняння (рис. 3.4).

Перший спосіб графічного відокремлення коренів

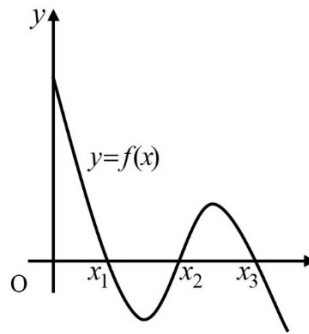


Рис. 3.4. Графічне відокремлення коренів (1-й спосіб).

На практиці ж буває зручніше замінити рівняння $f(x)=0$ рівносильним йому рівнянням $\varphi(x)=\psi(x)$, де $\varphi(x)$ й $\psi(x)$ – більш прості функції, ніж $f(x)$.

Другий спосіб графічного відокремлення коренів

Абсциси точок перетину графіків функцій $y=\varphi(x)$ і $y=\psi(x)$ дають корені рівняння $\varphi(x)=\psi(x)$, а отже, і початкового рівняння $f(x)=0$.

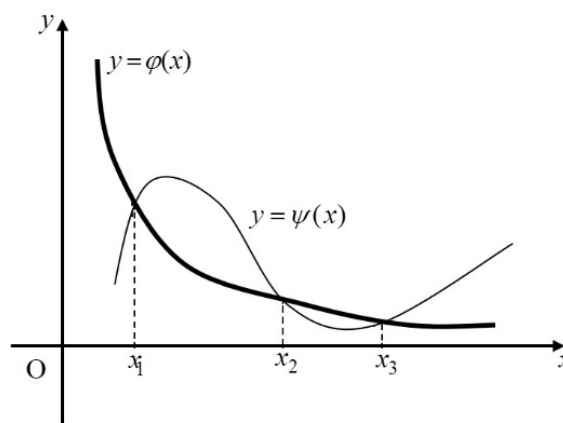


Рис. 3.5. Графічне відокремлення коренів (2-й спосіб).

Приклад 3.12. Відокремити графічно корінь рівняння $1-x^2+\frac{1}{6}x^3=0$

першим способом.

Розв'язок. Для розв'язування задачі побудуємо графік функції (рис. 3.6)

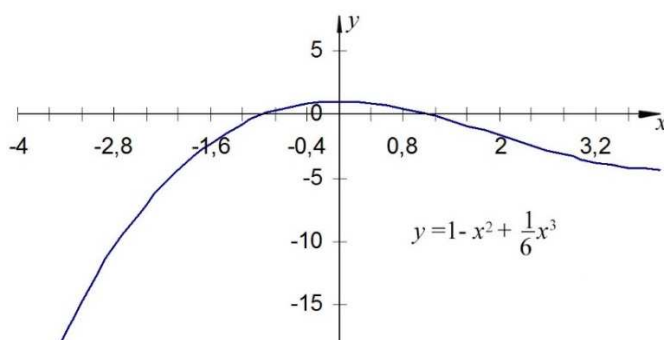


Рис. 3.6. Графік функції $y = 1 - x^2 + \frac{1}{6}x^3$.

З рисунка видно, що один з коренів рівняння належить відрізку $[-1,2; -0,8]$, другий – відрізку $[0,8; 1,2]$. Оскільки розглянуте рівняння третього степеня, то повинен існувати ще один корінь на інтервалі $(3,2; +\infty)$

Приклад 3.13. Відокремити графічно корінь рівняння

$$(x - 1)^2 - \frac{1}{2}e^x = 0 \text{ у другий спосіб.}$$

Розв'язок. Перетворимо рівняння до виду $(x - 1)^2 = \frac{1}{2}e^x$ й побудуємо

графіки функцій $y = (x - 1)^2$ і $y = \frac{1}{2}e^x$ (рис. 3.7).

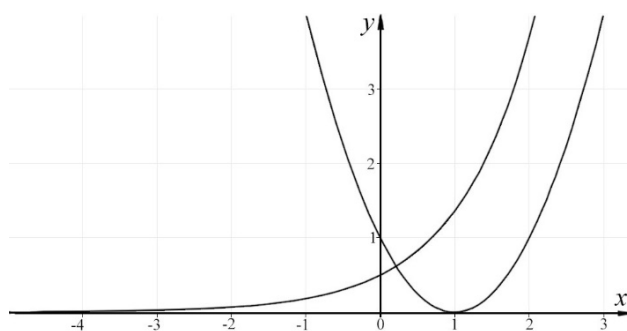


Рис. 3.7. Графічне відокремлення коренів

З рисунка видно, що абсциса точки перетину цих графіків належить відрізку $[0;1]$.

Аналитичне відділення коренів

Теорема 3.6. Про існування кореня рівняння

Якщо неперервна функція $f(x)$ набуває значення різних знаків на кінцях відрізка $[a; b]$, тобто $f(a) \cdot f(b) < 0$, то усередині цього відрізка існує щонайменше один корінь рівняння $f(x) = 0$, тобто таке значення x^* , що $f(x^*) = 0$.

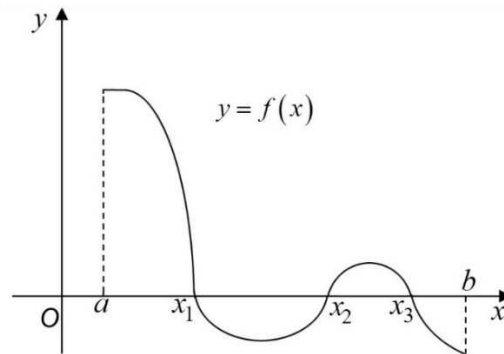


Рис. 3.8. Існування кореня на відрізку

Теорема 3.7. Про існування єдиного кореня. Якщо неперервна на відрізку $[a, b]$ функція $y = f(x)$ набуває на кінцях відрізка значення різних знаків, а похідна $f'(x)$ зберігає знак усередині відрізка $[a, b]$, то усередині відрізка існує єдиний корінь рівняння $f(x) = 0$ (рис. 3.9).

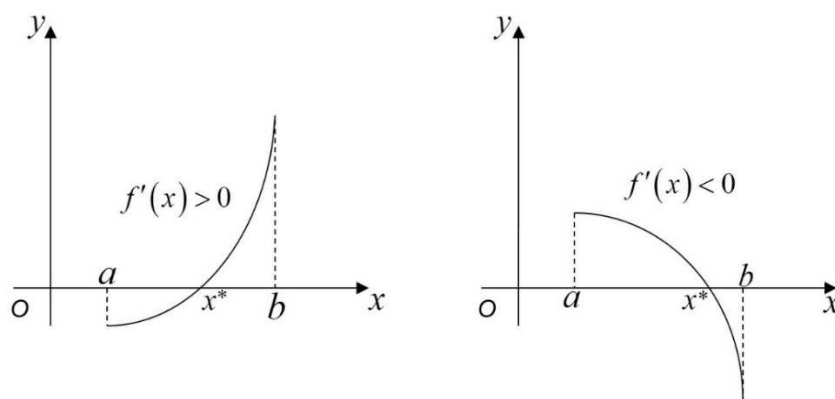


Рис. 3.9. Існування єдиного кореня на відрізку

Застосування теорем для відокремлення коренів

1) Визначаємо граничні точки $x = a$ й $x = b$ з області визначення функції $f(x)$.

2) Обчислюємо значення $f(x)$ на $[a;b]$ через проміжки довільної довжини h до зміни знака при переході від $f(x)$ до $f(x+h)$

Приклад 3.14. Підтвердити аналітично правильність знаходження відрізка ізоляції кореня рівняння $(x-1)^2 - \frac{1}{2}e^x = 0$.

Розв'язок. Для відрізка $[0;1]$ маємо: $f(0) = (0-1)^2 - \frac{1}{2}e^0 = 0.5$;
 $f(1) = (1-1)^2 - \frac{1}{2}e^1 = -\frac{1}{2}e = -1.359$. Оскільки $f(0) \cdot f(1) < 0$, то корінь відокремлений правильно.

Приклад 3.15. Відокремити корені рівняння $f(x) = \ln x + x^2 - 0.5 = 0$.

Розв'язок. З урахуванням області визначення ($x > 0$) складемо схему знаків функції.

Таблиця 3.12

Знаки значення функції

x	0.1	0.5	1
$\text{sign } f(x)$	(-)	(-)	(+)

$f'(x) = \frac{1+2x^2}{x} > 0$ при $x > 0$, тобто єдиний корінь належить відрізку $[0.5;1]$.

Розглянемо деякі алгоритми уточнення кореня.

3.9.3. Метод половинного ділення

Нехай потрібно уточнити єдиний корінь рівняння $f(x) = 0$, що належить відрізку $[a;b]$. Точка $c = \frac{a+b}{2}$ – середина відрізка $[a;b]$.

Якщо $f(c) = 0$, то корінь знайдений.

Якщо ні, то для подальшого розгляду залишаємо ту половину – $[a;c]$ або $[c;b]$ – на кінцях якої знаки функції $f(x)$ різні.

При цьому отримуємо послідовність вкладених відрізків, що містять шуканий корінь.

На кожному кроці довжина відрізка зменшується вдвічі. Метод сходиться завжди [31, 32].

Умова завершення алгоритму

Умовою закінчення пошуку кореня може бути, наприклад,

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

де ε – точність, $[a;b]$ – початковий відрізок, n – число ітерацій.

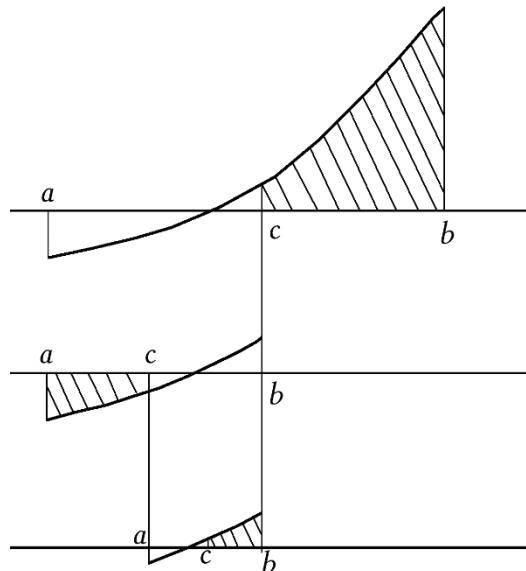


Рис. 3.10. Геометрична ілюстрація методу половинного ділення

Алгоритм методу половинного ділення

Вхідні дані: $f(x)$ – функція; ε – необхідна точність;

a, b – границі заданого інтервалу (границі пошуку кореня).

Результат: x_{np} – наближений корінь рівняння $f(x) = 0$.

Розв'язок

Step1: $c = \frac{a+b}{2};$

Step2: **If** ($f(c) = 0$) { Print; $x_{nb} = c$; Exit }

Step3: *If* $\left(\frac{|b-a|}{2^n} \leq \varepsilon\right)$ {Print; $x_{nb} = \frac{b+a}{2}$; Exit}

Step4: *If* $(f(a)f(c) < 0)$ { $b = c$ }
 else { $a = c$ }

goto Step1.

Приклад 3.16. Методом половинного ділення зробити шість ітерацій уточнення кореня рівняння $f(x) \equiv x^4 + 2x^3 - x - 1 = 0$, що належить відрізку $[0;1]$.

Таблиця 3.13

Проміжні обчислення методом половинного ділення

n	a	c	b	$f(a)$	$f(c)$	$f(b)$
1	0	0.5	1	-1	-1.19	1
2	0.5	0.75	1	-1.19	-0.59	1
3	0.75	0.875	1	-0.59	0.05	1
4	0.75	0.812	0.875	-0.59	-0.304	0.05
5	0.812	0.844	0.875	-0.304	-0.137	-0.05
6	0.844	0.860	0.875	-0.137	-0.041	0.05

Після шести ітерацій корінь локалізований на відрізку $[0.86; 0.875]$.

3.9.4. Метод пропорційних частин (метод хорд)

Нехай на відрізку $[a,b]$ функція неперервна, набуває на кінцях відрізка значення різних знаків, а похідна $f'(x)$ зберігає знак. Залежно від знака другої похідної можливі наступні випадки розташування кривих.

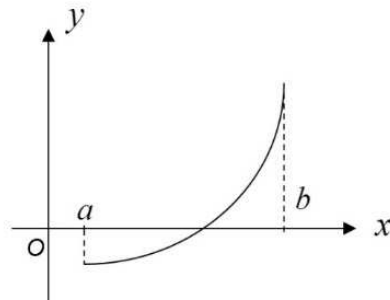
1) $f(a)f(b) < 0$

2) $sign(f'(x)) = const$

3) $sign(f''(x)) = const$

$f(a) < 0, f(b) > 0, f'(x) > 0$ – функція зростає

$f''(x) > 0$
крива опукла вниз



$f''(x) < 0$
крива опукла вверх

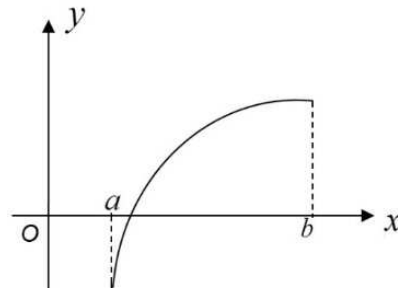
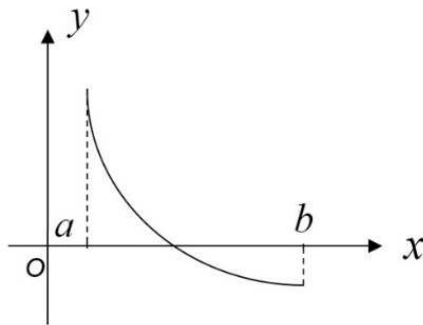


Рис. 3.11. Графік функції при $f(a) < 0, f(b) > 0, f'(x) > 0$

$f(a) > 0, f(b) < 0, f'(x) < 0$ – функція спадає

$f''(x) > 0$
крива опукла вниз



$f''(x) < 0$
крива опукла вверх

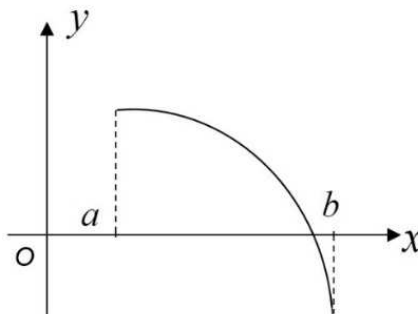


Рис. 3.12. Графік функції при $f(a) > 0, f(b) < 0, f'(x) < 0$

1. Випадок, коли $f'(x)$ та $f''(x)$ мають однакові знаки

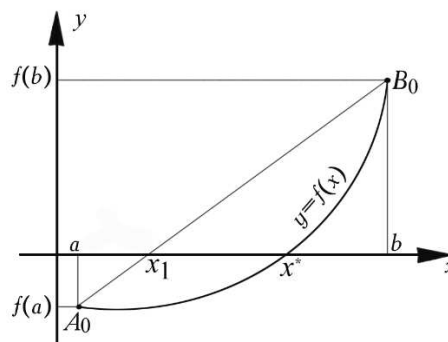


Рис. 3.13. Знаходження першого наближення графічно

Графік функції проходить через точки $A_0(a, f(a))$ й $B_0(b, f(b))$.

Шуканий корінь рівняння (точка x^*) нам невідомий, замість нього візьмемо

точку x_1 перетину хорди A_0B_0 з віссю абсцис. Це й буде наближене значення кореня.

У аналітичній геометрії виводиться формула, що задає рівняння прямої, що проходить через дві точки з координатами (x_1, y_1) та (x_2, y_2) :

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1}.$$

Нехай $(x_1, y_1) = (a, f(a))$ і $(x_2, y_2) = (b, f(b))$.

Тоді рівняння хорди A_0B_0 запишеться у вигляді: $\frac{x - a}{b - a} = \frac{y - f(a)}{f(b) - f(a)}$.

Знайдемо значення $x = x_1$, для якого $y = 0$: $x_1 = a - \frac{f(a)(b - a)}{f(b) - f(a)}$.

Тепер корінь знаходиться на відрізку $[x_1; b]$.

Застосуємо метод хорд до відрізка $[x_1; b]$. Проведемо хорду, що з'єднає точки $A_1(x_1, f(x_1))$ і $B_0(b, f(b))$, і знайдемо x_2 – точку перетину

хорди A_1B_0 з віссю Ox : $x_2 = x_1 - \frac{f(x_1)(b - x_1)}{f(b) - f(x_1)}$.

Продовжуючи цей процес, знаходимо: $x_3 = x_2 - \frac{f(x_2)(b - x_2)}{f(b) - f(x_2)}$.

Рекурентна формула обчислення наближень до кореня

Одержуємо рекурентну формулу обчислення наближень до кореня [33].

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

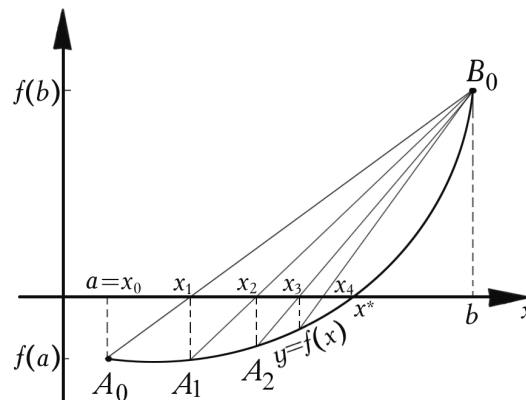


Рис. 3.14. Процес послідовного наближення до кореня зліва

Кінець b відрізка $[a, b]$ нерухомий, а кінець a переміщується.

Таким чином, одержуємо розрахункові формули методу хорд:

$$x_{n+1} = x_n - \frac{f(x_n)(b - x_n)}{f(b) - f(x_n)}; \quad x_0 = a. \quad (3.42)$$

$$f(a) < 0, f(b) > 0, \\ f'(x) > 0, f''(x) > 0$$

$$f(a) > 0, f(b) < 0, \\ f'(x) < 0, f''(x) < 0$$

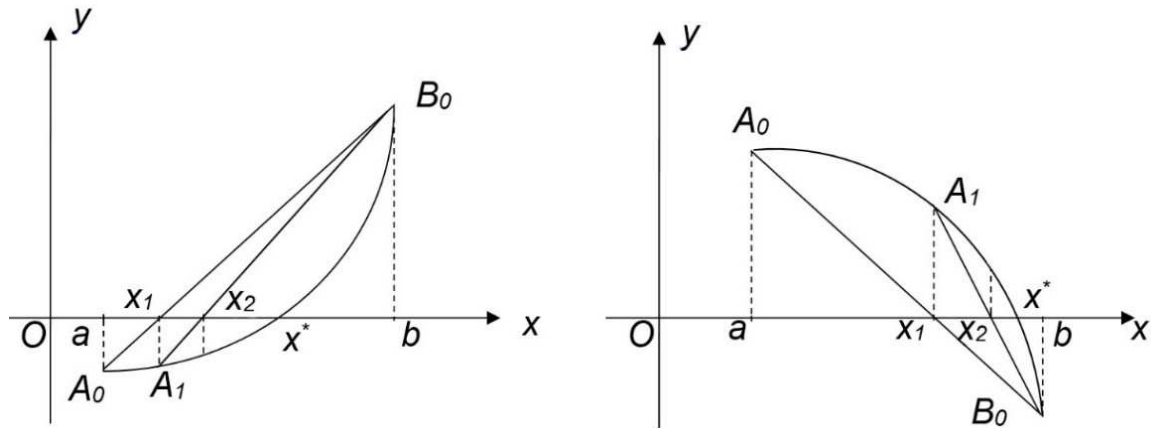


Рис. 3.15. Метод пропорційних частин для випадку $f'(x)f''(x) > 0$

Обчислення чергових наближень до точного кореня рівняння триває доти, поки не досягнемо заданої точності, тобто повинна виконуватися умова: $|x_{n+1} - x_n| < \varepsilon$, де ε – задана точність.

2. Випадок, коли $f'(x)$ й $f''(x)$ мають різні знаки

а. З'єднаємо точки $A_0(a, f(a))$ й $B_0(b, f(b))$ хордою A_0B_0 .

б. Точку перетину хорди з віссю Ox будемо вважати першим наближенням кореня.

в. У цьому випадку нерухомим кінцем відрізка буде кінець a .

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} \Rightarrow (x_1, y_1) = (b, f(b)) \quad (x_2, y_2) = (a, f(a))$$

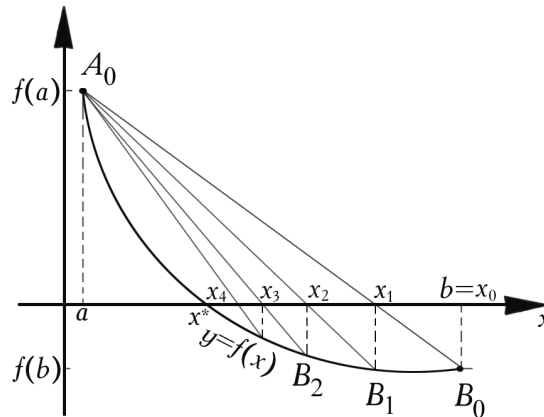


Рис. 3.16. Процес послідовного наближення до кореня справа Рівняння хорди A_0B_0 :

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

Звідси знайдемо x_1 , вважаючи, що $y = 0$:

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

Тепер корінь рівняння $x^* \in [a; x_1]$. Знайдемо $(x_1, f(x_1)) = B_1$.

Застосовуючи метод хорд до цього відрізка, одержимо:

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

Продовжуючи і т. д., одержимо:

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

Розрахункові формули методу:

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - a)}{f(x_n) - f(a)}, \quad x_0 = b. \quad (3.43)$$

Умова закінчення обчислень: $|x_{n+1} - x_n| < \varepsilon$.

Тоді $x_{np} = x_{n+1}$ з точністю ε .

Підсумок. Отже, якщо $f'(x) \cdot f''(x) > 0$, наближене значення кореня знаходять по формулі:

$$x_{n+1} = x_n - \frac{f(x_n)(b - x_n)}{f(b) - f(x_n)}; \quad x_0 = a.$$

Якщо ж $f'(x) \cdot f''(x) < 0$ – то по формулі:

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - a)}{f(x_n) - f(a)}, \quad x_0 = b.$$

$$f(a) < 0, f(b) > 0, \\ f'(x) > 0, f''(x) < 0$$

$$f(a) > 0, f(b) < 0, \\ f'(x) < 0, f''(x) > 0$$

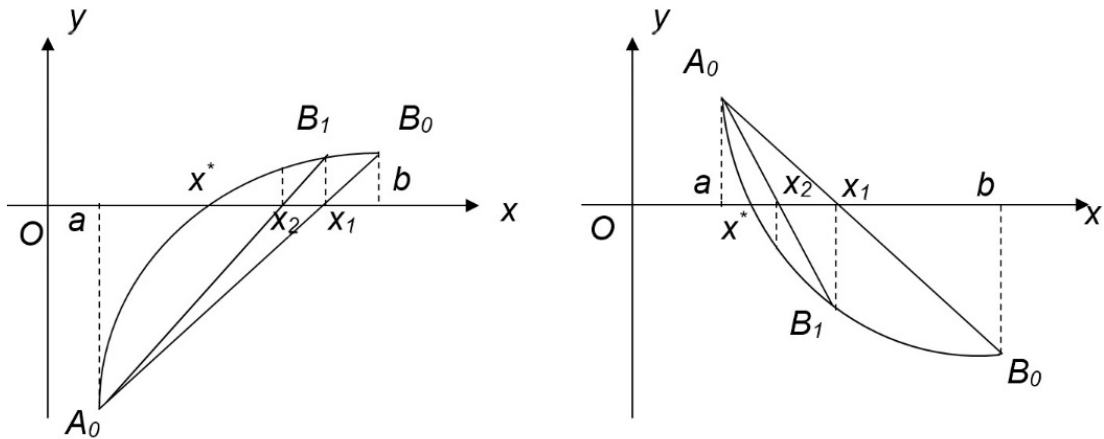


Рис. 3.17. Метод пропорційних частин для випадку $f'(x)f''(x) < 0$

Практичний вибір тієї або іншої формули здійснюють, користуючись наступним **правилом: нерухомим кінцем** відрізка є той, для якого **знак функції збігається зі знаком другої похідної** [34].

Оцінка точності наближення

Оцінку точності наближення за методом хорд визначають з виразу:

$$|x_n - x^*| \leq \frac{|f(x_n)|}{m_1},$$

де $|f'(x_n)| \geq m_1$ при $a \leq x_n \leq b$.

У випадку, коли необхідно оцінити абсолютну похибку по двох наступних наближеннях:

$$|x^* - x_n| \leq \frac{M_1 - m_1}{m_1} |x_n - x_{n-1}| \quad (3.44)$$

де m_1 й M_1 – відповідно найменше й найбільше значення модуля похідної $f'(x)$ на відрізку $[a; b]$.

Якщо відрізок $[a, b]$ настільки вузький, що справджується нерівність

$$M_1 \leq 2m_1,$$

то з формули $|x^* - x_n| \leq \frac{M_1 - m_1}{m_1} |x_n - x_{n-1}|$ одержуємо:

$$|x^* - x_n| \leq |x_n - x_{n-1}|.$$

Таким чином, у цьому випадку, як тільки буде виявлено, що

$$|x_n - x_{n-1}| < \varepsilon,$$

де ε – задана гранична абсолютна похибка, то гарантовано, що

$$|x^* - x_n| < \varepsilon.$$

Приклад 3.17. Вибрати формулу для розв'язування рівняння $(x-1)\ln(x)-1=0$, якщо відрізок ізоляції кореня $[2;3]$.

Розв'язок. Для цього прикладу $f(x) = (x-1)\ln(x)-1$;
 $f(2) = (2-1)0.69-1 = -0.31$; $f(3) = (3-1)1.1-1 = 1.2$

$$f'(x) = \ln(x) + \frac{x-1}{x}; \quad f'(2) = 0.69 + 0.5 = 1.19; \quad f'(3) = 1.1 + 0.75 = 1.85$$

$$f''(x) = \frac{1}{x} + \frac{1}{x^2}. \quad f''(2) = 0.5 + 0.25 = 0.75; \quad f''(3) = \frac{1}{3} + \frac{1}{9} = 0.44$$

Друга похідна в цьому прикладі додатна на відрізку ізоляції кореня $[2;3]$: $f''(x) > 0$, $f(3) > 0$, то $f(b) \cdot f''(x) > 0$.

Нерухома точка – це точка на кінці відрізка, в якій знак функції співпадає зі знаком другої похідної.

Тому для уточнення кореня вибираємо:

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

Приклад 3.18. Знайти додатний корінь рівняння

$$f(x) \equiv x^3 - 0.2x^2 - 0.2x - 1.2 = 0 \text{ з точністю до } 0.002.$$

Розв'язок.

Етап 1. Спочатку відокремлюємо корінь. Оскільки

$$f(1) = -0.6 < 0 \text{ і } f(1.5) = 1.43 > 0,$$

то шуканий корінь x^* лежить на відрізку $[1; 1.5]$. Тобто $a = 1$, $b = 1.5$.

Етап 2. Обчислюємо другу похідну: $f'(x) = 3x^2 - 0.4x - 0.2$

$$f''(x) = 6x - 0.4; \quad f''(1) = 6 - 0.4 = 5.6; \quad f''(1.5) = 9 - 0.4 = 8.6; \quad f''(x) > 0$$

Етап 3. Вибираємо нерухому точку:

$$f(1.5) = 1.425 > 0; \quad f''(1.5) = 8.6 > 0$$

Нерухома точка – це точка на кінці відрізка, у якій знак функції співпадає зі знаком другої похідної [35].

У цьому випадку – це точка b .

Етап 4. Тому для обчислення послідовних наближень застосуємо

формулу: $x_{n+1} = x_n - \frac{f(x_n)(b - x_n)}{f(b) - f(x_n)}$

$$a = 1; \quad b = 1.5; \quad x_0 = a; \quad f(1) = -0.6; \quad f(1.5) = 1.425$$

$$x_1 = x_0 - \frac{f(x_0)(b - x_0)}{f(b) - f(x_0)} = 1 - \frac{-0.6(1.5 - 1)}{1.425 + 0.6} = 1 + 0.15 = 1.15$$

$$f(x_1) = f(1.15) = -0.173;$$

$$x_2 = x_1 - \frac{f(x_1)(b - x_1)}{f(b) - f(x_1)} = 1.15 + \frac{0.173(1.5 - 1.15)}{1.425 + 0.173} = 1.15 + 0.040 = 1.190;$$

$$f(x_2) = f(1.190) = -0.036;$$

$$x_3 = x_2 - \frac{f(x_2)(b - x_2)}{f(b) - f(x_2)} = 1.190 + \frac{0.036(1.5 - 1.190)}{1.425 + 0.036} = 1.190 + 0.008 = 1.198$$

$$f(x_3) = f_3(1.198) = -0.0072.$$

$$x_4 = x_3 - \frac{f(x_3)(b - x_3)}{f(b) - f(x_3)} = 1.198 + \frac{0.0072(1.5 - 1.198)}{1.425 + 0.0072} = 1.198 + 0.0015 = 1.1995$$

$$f(x_4) = f(1.1995) = -0.0018.$$

Етап 5. Обчислюємо похибку методу:

Оскільки $f'(x) = 3x^2 - 0.4x - 0.2$ і при $x_3 < x < b$, тобто при $x_3 < x < 1.5$ маємо $|f'(x_3)| \geq 3 \cdot 1.1995^2 - 0.4 \cdot 1.1995 - 0.2 = 3.63 = m_1$

$$|f'(b)| \geq 3 \cdot 1.5^2 - 0.4 \cdot 1.5 - 0.2 = 5.95 = M_1, \quad M_1 < 2m_1.$$

Тому можемо застосувати оцінку: $|x_n - x_{n-1}| < \varepsilon$

Одержуємо оцінку: $|1.198 - 1.1995| < 0.0015$.

Оцінка через значення функції:

$$|x_n - x^*| \leq \frac{|f(x_n)|}{|f'(x_n)|}$$

то можна прийняти:

$$0 < x^* - x_3 < \frac{0.0018}{3.63} \approx 0.0005.$$

3.9.5. Метод Ньютона (метод дотичних)

Нехай корінь x^* рівняння $f(x) = 0$ відділений на відрізку $[a; b]$, причому $f'(x)$ й $f''(x)$ неперервні й зберігають певні знаки при $a \leq x \leq b$.

1. Розглянемо випадок, коли $f'(x)$ й $f''(x)$ мають однакові знаки.

Тоді можливі два випадки побудови кривої на відрізку $[a, b]$

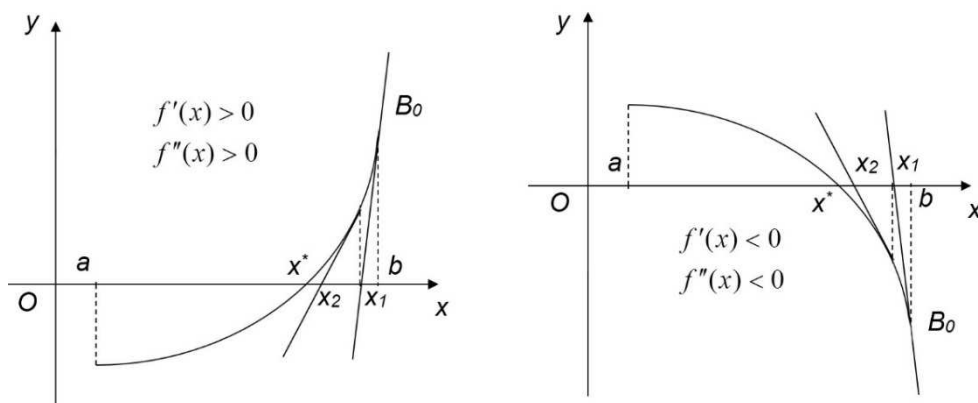


Рис. 3.18. Метод дотичних для випадку $f'(x)f''(x) > 0$

Пошук кореня на відрізку $[a, b]$

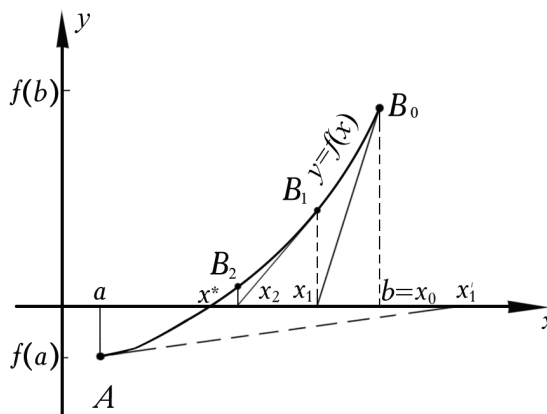


Рис. 3.19. Процес послідовного наближення до кореня справа

Проведемо дотичну до кривої $y = f(x)$ в точці $B_0(b, f(b))$.

У курсі алгебри виводиться рівняння дотичної.

Рівняння дотичної в точці B_0 має вигляд

$$y - f(b) = f'(b)(x - b).$$

Як чергове наближення до кореня рівняння беремо точку перетину дотичної з віссю Ox . Вважаючи $y = 0$ у рівнянні дотичної, знайдемо

$$x_1 = b - \frac{f(b)}{f'(b)}. \quad \text{Тепер } x^* \in [a, x_1].$$

Пошук кореня на відрізку $[a, x_1]$

Застосовуючи метод ще раз для відрізка $[a, x_1]$, одержимо:

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

Для загального випадку одержуємо рекурентну формулу обчислення наближень до кореня [36]:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (3.45)$$

Зверніть увагу, що в цьому випадку як початкове наближення до кореня вибираємо точку $x_0 = b$.

Наближення до кореня відбувається з правої сторони, тому одержуємо наближене значення кореня з надлишком.

2. Розглянемо випадок, коли коли $f'(x)$ й $f''(x)$ мають різні знаки

При цьому можливі два випадки побудови кривої на відрізку $[a, b]$.

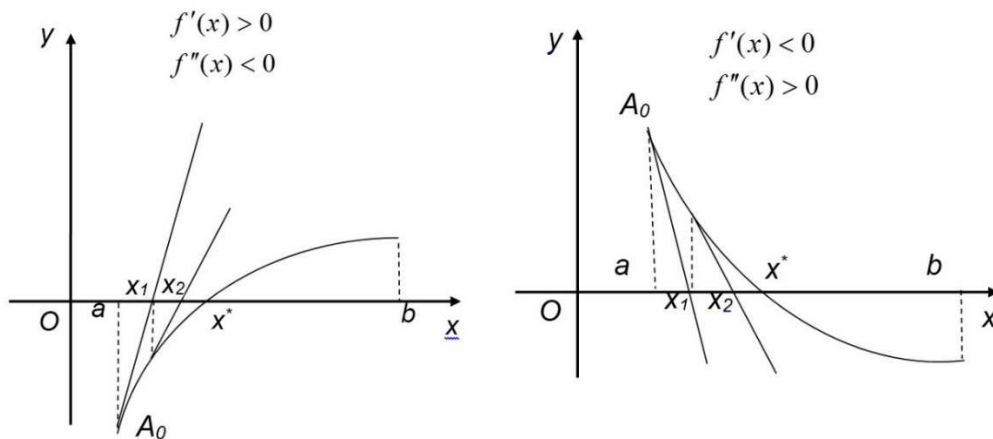


Рис. 3.20. Метод дотичних для випадку $f'(x)f''(x) < 0$

Пошук кореня на відрізку $[a, b]$

Якщо знову провести дотичну до кривої в точці B_0 , то вона перетне вісь Ox у точці, що не належить відрізку $[a, b]$. Тому проведемо дотичну в точці $A_0(a, f(a))$.

Рівняння дотичної: $y - f(a) = f'(a)(x - a)$.

Знаходимо x_1 , вважаючи $y = 0$: $x_1 = a - \frac{f(a)}{f'(a)}$. Корінь $x^* \in [x_1; b]$.

Застосовуючи метод ще раз для відрізка $[x_1; b]$, одержимо:

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

Одержуємо рекурентну формулу обчислення наближень до кореня, аналогічну першому випадку [37]:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \text{ Початкове наближення – точка } x_0 = a.$$

Підсумок. Розглянуто два випадки:

1. За x_0 приймаємо кінець b відрізка $[a, b]$,
2. За x_0 приймаємо початок a відрізка $[a, b]$.

При виборі початкового наближення кореня можна керуватися правилом: *за початкову точку слід вибрати той кінець відрізка $[a, b]$, у якому знак функції співпадає зі знаком другої похідної.*

Умова закінчення обчислювального процесу:

$$|x_{n-1} - x_n| < \varepsilon, \text{ де } \varepsilon - \text{ задана точність. Тоді } x^* = x_n \pm \varepsilon.$$

Приклад 3.19. Обчислити методом Ньютона корінь рівняння $f(x) \equiv x^4 - 3x^2 - 75x - 10000 = 0$ з п'ятьма вірними знаками.

Розв'язок

Етап 1. Вважаючи у лівій частині $x = 0, 10, 100, \dots$, одержимо

$$f(0) = -10000, \quad f(10) = -1050, \quad f(100) \approx +10^8.$$

Отже, шуканий корінь x^* перебуває в інтервалі $10 < x^* < 100$.

Етап 2. Звизимо знайдений інтервал.

Оскільки $f(11) = 3453$, то $10 < x^* < 11$.

На інтервалі $(10, 11)$: $f(11) > 0$ і $f''(11) > 0$; $f(10) < 0$ і $f''(10) > 0$

Оскільки $f''(x) > 0$ на інтервалі $[10,11]$, то можемо прийняти за початкове наближення $b = x_0 = 11$. Послідовні наближення $x_n (n = 1, 2, \dots)$ обчислюємо за схемою, яка наведена в таблиці 3.14.

Таблиця 3.14

Проміжні обчислення методом Ньютона

n	x_n	$f(x_n)$	$f'(x_n)$	$ x_{n+1} - x_n $	
0	11	3453	5183		
1	10.3	164.3	4234	0.7	$0.7 > \varepsilon$
2	10.262	4.33	4186	0.038	$0.3 > \varepsilon$
3	10.261	0.15	4184	0.001	$0.001 < \varepsilon$

Чергове наближення в даному випадку обчислюємо за формулою:

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

3.9.6. Видозмінений метод Ньютона

Якщо похідна $f'(x)$ мало змінюється на відрізку $[a; b]$, то у формулі $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ покладемо $f'(x_n) \approx f'(x_0)$.

Звідси для кореня x^* рівняння $f(x) = 0$ одержуємо послідовні наближення

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)} \quad (n = 0, 1, \dots)$$

Геометрично цей спосіб означає, що ми заміняємо дотичні в точках $B_n [x_n, f(x_n)]$ прямими, паралельними дотичній до кривої $y = f(x)$, у її фіксованій точці $B_0 [x_0, f(x_0)]$.

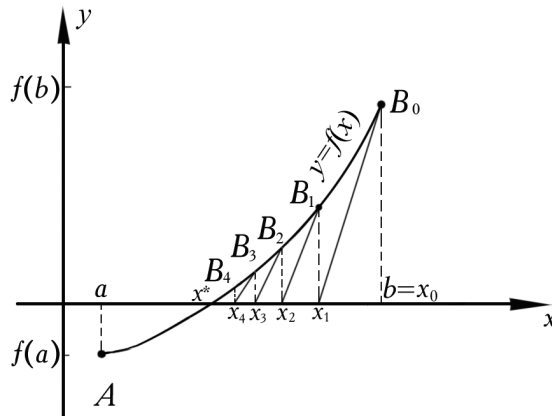


Рис. 3.21. Процес наближення до кореня видозміненим методом Ньютона

Формула $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ ($n = 0, 1, \dots$) позбавляє необхідності

обчислювати щоразу значення похідної $f'(x_n)$. Тому формула вельми корисна у тому випадку, коли для обчислення $f'(x)$ потрібні більші витрати часу.

3.9.7. Комбінований метод

Нехай на відрізку $[a, b]$

1. $f(a)f(b) < 0$,
2. $\text{sign}(f'(x)) = \text{const}$,
3. $\text{sign}(f''(x)) = \text{const}$.

Поєднаємо метод хорд і метод дотичних.

Одержуємо метод, на кожному етапі якого знаходимо:

- а) значення з нестачею,
- б) значення з надлишком точного кореня x^* рівняння $f(x) = 0$.

Теоретично тут можливі чотири випадки:

$$f'(x) > 0; f''(x) > 0; f'(x) < 0; f''(x) < 0; f'(x) \cdot f''(x) > 0$$

$$f'(x) < 0; f''(x) > 0; f'(x) > 0; f''(x) < 0; f'(x) \cdot f''(x) < 0$$

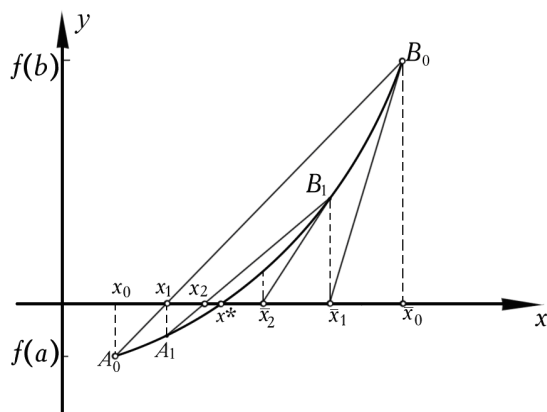


Рис. 3.22. Процес наближення до кореня комбінованим методом

Розглянемо випадок: $f(a)f(b) < 0$, $f'(x) > 0$; $f''(x) > 0$

Інші випадки можна звести до даного, увівши заміну в рівняння

$$f(x) = 0:$$

$$1) -f(x) = 0.$$

$$2) \pm f(z) = 0 \text{ де } z = -x.$$

Отже, нехай $f'(x) > 0$ і $f''(x) > 0$ при $a \leq x \leq b$. Вважаємо $x_0 = a$;

$\bar{x}_0 = b$ й

$$\bar{x}_1 = b - \frac{f(b)}{f'(b)} = \bar{x}_0 - \frac{f(\bar{x}_0)}{f'(\bar{x}_0)}, \text{ (метод дотичних)}$$

$$x_1 = a - \frac{f(a)(b-a)}{f(b)-f(a)} = x_0 - \frac{f(x_0)(\bar{x}_0 - x_0)}{f(\bar{x}_0) - f(x_0)}; \text{ (метод хорд)}$$

$$(n = 0, 1, 2, \dots)$$

$$\bar{x}_2 = \bar{x}_1 - \frac{f(\bar{x}_1)}{f'(\bar{x}_1)}, \text{ (} n = 0, 1, 2, \dots \text{). (метод дотичних)}$$

$$x_2 = x_1 - \frac{f(x_1)}{f(\bar{x}_1) - f(x_1)} (\bar{x}_1 - x_1); \text{ (метод хорд)}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f(\bar{x}_n) - f(x_n)} (\bar{x}_n - x_n); \text{ (метод хорд)}$$

$$\bar{x}_{n+1} = \bar{x}_n - \frac{f(\bar{x}_n)}{f'(\bar{x}_n)}, \text{ (} n = 0, 1, 2, \dots \text{). (метод дотичних)}$$

Справедлива нерівність: $x_n < x^* < \bar{x}_n$.

Звідси $0 < x^* - x_n < \bar{x}_n - x_n$.

Нехай ε – припустима похибка x_{np} , тоді процес зближення припиняється при $\bar{x}_n - x_n < \varepsilon$.

По закінченні процесу за значення кореня x^* найкраще брати середнє арифметичне отриманих останніх значень:

$$\bar{x}^* = \frac{1}{2}(x_n + \bar{x}_n).$$

Приклад 3.20. Обчислити з точністю до 0.0005 єдиний додатний корінь рівняння

$$f(x) \equiv x^5 - x - 0.2 = 0.$$

Розв'язок

Етап 1. Відокремлюючи корені, знаходимо $f(1) < 0$ й $f(1.1) > 0$

Етап 2. Визначаємо, що ізольований корінь міститься на відріжку $[1, 1.1]$.

Етап 3. Обчислюємо першу й другу похідну

$$f'(x) = 5x^4 - 1, \quad f'(x) > 0.$$

$$f''(x) = 20x^3, \quad f''(x) > 0.$$

Оскільки $b = 1.1$; $f(b) > 0$; $f''(b) > 0$, прийmemo $x_0 = a$ й $\bar{x}_0 = b$.

Виходячи з попередніх розрахунків $x_0 = 1$ і $\bar{x}_0 = 1.1$.

Обчислимо значення функції й похідної у точці x_0 .

$$f(x_0) = f(1) = -0.2;$$

$$f(\bar{x}_0) = f(1.1) = 0.3105;$$

$$f'(\bar{x}_0) = f'(1.1) = 6.3205.$$

$$x_1 = x_0 - \frac{f(x_0)(\bar{x}_0 - x_0)}{f(\bar{x}_0) - f(x_0)} = 1 - \frac{-0.2(1.1 - 1)}{0.3105 + 0.2} = 1.039;$$

$$\bar{x}_1 = \bar{x}_0 - \frac{f(\bar{x}_0)}{f'(\bar{x}_0)} = 1.1 - \frac{0.3105}{6.3205} = 1.051$$

Через те, що $\bar{x}_1 - x_1 = 1.051 - 1.039 = 0.012$, то точність недостатня.

Знаходимо наступну пару наближень:

$$x_2 = x_1 - \frac{f(x_1)(\bar{x}_1 - x_1)}{f(\bar{x}_1) - f(x_1)} = 1.039 + \frac{0.012 \cdot 0.0282}{0.0595} \approx 1.04469$$

$$\bar{x}_2 = \bar{x}_1 - \frac{f(\bar{x}_1)}{f'(\bar{x}_1)} = 1.051 - \frac{0.0313}{5.1005} \approx 1.04487.$$

Тут $\bar{x}_2 - x_2 = 0.00018$, тобто потрібний ступінь точності досягнутий.

Можна покласти

$$\bar{x}^* = \frac{1}{2}(1.04469 + 1.04487) = 1.04478 \approx 1.045$$

3.9.8. Метод ітерації

Нехай дано рівняння $f(x) = 0$ й відрізок $[a; b]$, функція $f(x)$ задовольняє умовам:

- $f(a) \cdot f(b) < 0$, тобто $f(x) = 0$ має різні знаки на кінцях відрізка,
- $f'(x) \neq 0$ на $[a; b]$.

Замінімо рівняння $f(x) = 0$ рівносильним йому рівнянням $x = \varphi(x)$.

Вибираючи довільне значення $x_0 \in [a; b]$, обчислимо

$$x_1 = \varphi(x_0), \quad x_2 = \varphi(x_1), \quad x_{n+1} = \varphi(x_n), \quad n = 0, 1, \dots$$

Процес збігається до кореня, якщо виконані умови теореми про збіжність (достатні умови збіжності) [38].

Теорема 3.8. *Про збіжність ітераційного процесу*

Якщо функція $\varphi(x)$: 1. визначена й диференційована на $[a, b]$,

2. $\varphi(x) \in [a; b]$.

Тоді, якщо існує правильний дріб q , такий, що $|\varphi'(x)| \leq q < 1$ при $x \in (a; b)$, то процес ітерації сходиться незалежно від початкового наближення $x_0 \in [a; b]$ і граничне значення $x^* = \lim_{n \rightarrow \infty} x_n$ є єдиним коренем рівняння $x = \varphi(x)$ на відрізку $[a; b]$.

Приведемо один зі способів рівносильного перетворення рівняння $f(x) = 0$ до виду $x = \varphi(x)$.

$$\text{Побудуємо функцію: } \varphi(x) = \lambda \cdot f(x) + x, \quad (3.46)$$

$$\text{де } \begin{cases} -\frac{1}{r} < \lambda < 0, \text{ если } f'(x) > 0, \\ 0 < \lambda < \frac{1}{r}, \text{ если } f'(x) < 0, \end{cases}$$

$$r = \max(|f'(a)|, |f'(b)|).$$

Тоді процес ітерації $x_{n+1} = \varphi(x_n)$, $n = 0, 1, \dots$ сходиться до кореня x^* при кожному $x_0 \in [a; b]$.

Швидкість збіжності оцінюється так:

$$|x^* - x| \leq \frac{m}{1-q} q^n,$$

$$m = |x_0 - \varphi(x_0)|, \quad q = \lambda \cdot r + 1 \text{ або } q = \max|\varphi'(x)|, \quad x \in [a; b].$$

Прикладом умов закінчення ітераційного процесу можуть бути:

$$\begin{cases} \Delta x_n = |x_n - x_{n-1}| < \varepsilon, \\ |f(x_n) - f(x_{n-1})| < \varepsilon, \end{cases}$$

або

$$\frac{1-q}{q} \Delta x_n < \varepsilon, \text{ де } \varepsilon \text{ – необхідна точність.}$$

Геометрично процес ітерації – знаходження абсциси точки перетину кривої $y = \varphi(x)$ і прямої $y = x$.

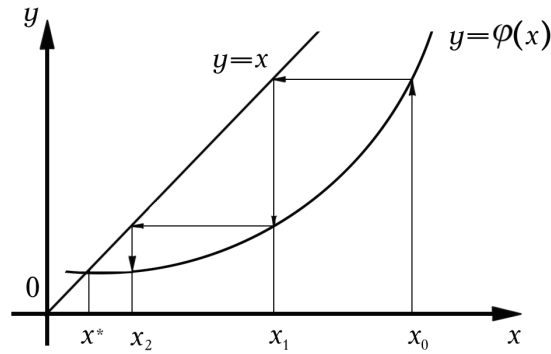


Рис. 3.23. Процес наближення до кореня методом ітерацій

Приклад 3.21.

Знайти методом ітерацій корінь рівняння

$$f(x) = 4x - 5\ln(x) - 5 = 0,$$

який належить відрізку $[0,25;0.75]$.

Розв'язок. Виберемо $x_0 = 0.5$. Обчислимо значення функції і її похідної:

$$f(a) = 1 - 5\ln(0.25) - 5 = 1 + 5 \cdot 1.3863 - 5 > 0,$$

$$f(b) = 3 - 5\ln(0.75) - 5 = 3 + 5 \cdot (0.2877 - 1) < 0,$$

$$f'(x) = 4 - \frac{5}{x} < 0 \text{ на відрізку } [0,25;0.75],$$

$$f'(0.25) = -16, \quad f'(0.75) = -\frac{8}{3}.$$

Згідно з формулою $0 < \lambda < \frac{1}{r}$, якщо $f'(x) < 0$, обчислимо $\lambda : r = 16$,

$$0 < \lambda < \frac{1}{16}, \text{ візьмемо } \lambda = 0.05.$$

Використовуючи вираз $\varphi(x) = \lambda \cdot f(x) + x$, запишемо

$$\varphi(x) = 0.2x - 0.25\ln(x) - 0.25 + x = 1.2x - 0.25(1 + \ln(x))$$

Звідси одержуємо рекомендовану формулу за методом ітерацій:

$$x_{n+1} = 1.2x_n - 0.25(1 + \ln(x_n)), \quad n = 0, 1, 2, \dots$$

Використовуючи цю формулу, обчислимо:

$$x_1 = 1.2x_0 - 0.25(1 + \ln(x_0)) = 0.525$$

Модифікований метод ітерації. Використаний в попередньому випадку вираз для формування ітерацій

$$x_{n+1} = \varphi(x_n), \quad n = 0, 1, \dots$$

перепишемо в такому вигляді: $x_{n+1} = x_n + \alpha \Delta x$, $\Delta x = f(x_n) - x_n$.

Ітераційний процес представимо у вигляді $x_{n+1} = x_n + \alpha \Delta x_n$,

де α – шуканий коефіцієнт, що покращує збіжність ітерацій.

Найкращим вибором α буде той, при якому $x_{n+1} = x^*$.

У цьому випадку ітерації визначають по формулах

$$x_{n+1} = x_n + \alpha(f(x_n) - x_n),$$

$$\alpha = \frac{1}{1 - f'(\xi)}, \quad x_n \leq \xi < x^*$$

Для $f'(\xi)$ можна прийняти наближення $f'(\xi) = \frac{f(x_n) - x_n}{x_n - x_{n-1}}$.

Правило вибору α :

$$\alpha = \begin{cases} a < 0, \text{ якщо } f'(x) > 1, \\ 0 < \alpha < \frac{1}{2}, \text{ якщо } f'(x) < -1, \\ \frac{1}{2} < \alpha < 1, \text{ якщо } -1 < f'(x) < 0, \\ \alpha > 0, \text{ если } 0 < f'(x) < 1. \end{cases}$$

Вибираючи α згідно із правилом, одержимо збіжний процес ітерацій. Ітераційні методи мають властивість самовиправлення – помилка на якійсь з ітерацій може бути прийнята за чергове наближення, що впливає лише на швидкість обчислення кореня. Помилки округлення не накопичуються.

Контрольні запитання

1. Методом пропорційних частин обчислити один з коренів рівняння:

$2x^3 - x^2 + 2x = 5$ на відрізку $[1; 2]$.

2. Методом ітерацій обчислити один з коренів рівняння $3x^3 - 5x^2 + 5x = 4$. Корінь знаходиться на відрізку $[1; 2]$.

3. У рівнянні $0.1x^3 - 0.1x^2 - x = 2$ уточнити значення кореня методом хорд на відрізку $[4, 5]$.

4. Методом Ньютона обчислити один з коренів рівняння: $0.1x^3 + x^2 + 2x = 4$ на відрізку $[1; 2]$. Виконати дві ітерації.

5. Дано рівняння $0.4 \cdot x^3 - x^2 = 3$. Ізольований корінь перебуває на відрізку $[3; 4]$. Уточнити значення кореня методом Ньютона.

3.10. Операції з матрицями та векторами

3.10.1. Матриці і операції з матрицями

Визначення матриці. Таблицю дійсних чисел виду

$$A = \begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{vmatrix}$$

називають *матрицею розміру $m \times n$* або *$m \times n$ -матрицею*.

Числа a_{ij} називають *елементами* цієї матриці. Індекс i указує номер рядка, а індекс j – номер стовпця, на перетині яких розташований даний елемент a_{ij} . Для явної вказівки елементів матриці використовують позначення $A = (a_{ij})$ [39].

Сума матриць. Нехай a_{ij}, b_{ij}, c_{ij} – елементи $m \times n$ -матриць A, B, C відповідно. Матрицю C називають *сумою матриць A і B* ($C = A + B$), якщо $c_{ij} = a_{ij} + b_{ij}, 1 \leq i \leq m, 1 \leq j \leq n$.

Приклад 3.22. Необхідно обчислити суму матриць A і B , де

$$A = \begin{pmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \\ 31 & 32 & 33 \end{pmatrix} \text{ і } B = \begin{pmatrix} 89 & 88 & 87 \\ 79 & 78 & 77 \\ 69 & 68 & 67 \end{pmatrix}$$

Розв'язок.

$$C = A + B = \begin{pmatrix} 11+89 & 12+88 & 13+87 \\ 21+79 & 22+78 & 23+77 \\ 31+69 & 32+68 & 33+67 \end{pmatrix} = \begin{pmatrix} 100 & 100 & 100 \\ 100 & 100 & 100 \\ 100 & 100 & 100 \end{pmatrix}$$

Різниця матриць. Аналогічно визначають різницю матриць, а саме, $C = A - B$, якщо $c_{ij} = a_{ij} - b_{ij}$, $1 \leq i \leq m$, $1 \leq j \leq n$. Додавати й віднімати можна тільки матриці однакових розмірів.

Приклад 3.23. Необхідно обчислити різницю матриць A і B , де

$$A = \begin{pmatrix} 10 & 7 & 5 \\ 15 & 9 & 6 \\ 20 & 11 & 7 \end{pmatrix} \text{ і } B = \begin{pmatrix} 5 & 3 & 2 \\ 10 & 5 & 3 \\ 15 & 7 & 4 \end{pmatrix}$$

Розв'язок.

$$C = \begin{pmatrix} 10-5 & 7-3 & 5-2 \\ 15-10 & 9-5 & 6-3 \\ 20-15 & 11-7 & 7-4 \end{pmatrix} = \begin{pmatrix} 5 & 4 & 3 \\ 5 & 4 & 3 \\ 5 & 4 & 3 \end{pmatrix}.$$

Добуток матриці на дійсне число. Матрицю C називають *добутком* $m \times n$ -матриці $A = (a_{ij})$ на дійсне число α ($C = \alpha A$), якщо кожний з її елементів формується відповідно до формули: $c_{ij} = \alpha a_{ij}$, $1 \leq i \leq m$, $1 \leq j \leq n$.

Приклад 3.24. Обчислити добуток матриці $A = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 2 \\ 1 & 2 & 1 \end{pmatrix}$ на число $\alpha = 3$.

Розв'язок. $C = \alpha \cdot A = \begin{pmatrix} 3 \cdot 1 & 3 \cdot 2 & 3 \cdot 1 \\ 3 \cdot 2 & 3 \cdot 1 & 3 \cdot 2 \\ 3 \cdot 1 & 3 \cdot 2 & 3 \cdot 1 \end{pmatrix} = \begin{pmatrix} 3 & 6 & 3 \\ 6 & 3 & 6 \\ 3 & 6 & 3 \end{pmatrix}$

Добуток двох матриць. Добуток $m \times n$ -матриці $A = (a_{ij})$ на $s \times k$ -матрицю $B = (b_{ij})$ ($C = AB$) визначений лише при $s = n$ як $m \times k$ -матриця C

з елементами c_{ij} , $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \dots + a_{in}b_{nj} = \sum_{p=1}^n a_{ip}b_{pj}$, $1 \leq i \leq m$, $1 \leq j \leq k$.

Добуток матриць визначений, якщо число стовпців матриці A дорівнює числу рядків матриці B .

Приклад 3.25. Знайти добуток матриць

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \text{ і } B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix}$$

Розв'язок

$$C = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} & a_{11}b_{13} + a_{12}b_{23} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} & a_{21}b_{13} + a_{22}b_{23} \\ a_{31}b_{11} + a_{32}b_{21} & a_{31}b_{12} + a_{32}b_{22} & a_{31}b_{13} + a_{32}b_{23} \end{pmatrix}$$

Приклад 3.26. Знайти добуток матриць

$$A = \begin{pmatrix} 7 & 8 \\ 9 & 10 \end{pmatrix} \text{ і матриці } B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}.$$

Розв'язок

$$C = \begin{pmatrix} 7 \cdot 1 + 8 \cdot 4 & 7 \cdot 2 + 8 \cdot 5 & 7 \cdot 3 + 8 \cdot 6 \\ 9 \cdot 1 + 10 \cdot 4 & 9 \cdot 2 + 10 \cdot 5 & 9 \cdot 3 + 10 \cdot 6 \end{pmatrix} = \begin{pmatrix} 39 & 54 & 69 \\ 49 & 68 & 87 \end{pmatrix}$$

Транспонування матриць. Транспонованою до $m \times n$ -матриці $A = (a_{ij})$ називають $n \times m$ -матрицю $A^T = (a'_{ij})$ з елементами $a'_{ij} = a_{ji}$, $1 \leq i \leq m, 1 \leq j \leq n$. Для транспонованої матриці справедливі наступні властивості:

1. $(A^T)^T = A$, тобто двічі транспонована матриця співпадає з початковою.
2. $(A + B)^T = A^T + B^T$. Транспонована сума матриць дорівнює сумі транспонованих матриць.
3. $(AB)^T = B^T A^T$. Транспонований добуток матриць дорівнює добутку транспонованих матриць.

3.10.2. Вектори і операції з векторами

Вектор-стовпець. Під вектором-стовпцем x розміром $n \times 1$ розуміють матрицю розміром $n \times 1$ [40], тобто

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

Вектор-рядок. Вектором-рядком x розміром $1 \times n$ називають матрицю розміром $1 \times n$, тобто

$$x = (x_1, x_2, \dots, x_n).$$

Рівність векторів. Два вектори вважають рівними, якщо вони мають однакове число компонентів, і якщо їх відповідні компоненти рівні, тобто $x = y$, якщо співпадають їх розмірності та $x_1 = y_1, \dots, x_n = y_n$.

Додавання векторів. Вектори однакоких розмірів, що є одночасно стовпцями або рядками, можна додавати як матриці однакоких розмірів.

Нехай дано два вектори-рядка $x = (x_1, x_2, \dots, x_n)$ й $y = (y_1, y_2, \dots, y_n)$. Суму цих векторів визначають за формулою:

$$x + y = x_1 + y_1, x_2 + y_2, \dots, x_n + y_n$$

У випадку векторів-стовпців

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \text{ та } y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \text{ їх сума } x + y = \begin{pmatrix} x_1 + y_1 \\ x_2 + y_2 \\ \vdots \\ x_n + y_n \end{pmatrix}$$

Приклад 3.27. Нехай $x = (1, 2, 3, 4, 5)$ і $y = (6, 7, 8, 9, 10)$.

Обчислити $x + y$.

Розв'язок.

$$x + y = (1 + 6, 2 + 7, 3 + 8, 4 + 9, 5 + 10) = (7, 9, 11, 13, 15).$$

Різниця векторів. Дано два вектори-рядка $x = (x_1, x_2, \dots, x_n)$ й $y = (y_1, y_2, \dots, y_n)$. Різниця цих векторів: $x - y = x_1 - y_1, x_2 - y_2, \dots, x_n - y_n$

Різниця векторів-стовпців:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \text{ та } y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \text{ їх різниця } x - y = \begin{pmatrix} x_1 - y_1 \\ x_2 - y_2 \\ \vdots \\ x_n - y_n \end{pmatrix}$$

Добуток матриці на вектор. Добуток $m \times n$ -матриці A на вектор-стовпець x визначають як $y = Ax$, або

$$y_i = a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = \sum_{p=1}^n a_{ip}x_p, 1 \leq i \leq m,$$

де a_{ip} – елементи матриці A , x_p – компоненти вектора x , y_i – компоненти вектора y .

Приклад 3.28.

$$\text{Обчислити добуток матриці } A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \text{ на вектор } x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

Розв'язок

$$y = Ax = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} a_{11}x_1 + a_{12}x_2 + a_{13}x_3, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 \end{pmatrix}$$

Приклад 3.29.

$$\text{Обчислити добуток матриці } A = \begin{pmatrix} 1 & 2 & 5 \\ 3 & 4 & 1 \\ 7 & 4 & 2 \end{pmatrix} \text{ на вектор-стовпець } x = \begin{pmatrix} 3 \\ 2 \\ 4 \end{pmatrix}.$$

Розв'язок

$$y = Ax = \begin{pmatrix} 1 & 2 & 5 \\ 3 & 4 & 1 \\ 7 & 4 & 2 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 2 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 \cdot 3 + 2 \cdot 2 + 5 \cdot 4 \\ 3 \cdot 3 + 4 \cdot 2 + 1 \cdot 4 \\ 7 \cdot 3 + 4 \cdot 2 + 2 \cdot 4 \end{pmatrix} = \begin{pmatrix} 27 \\ 21 \\ 37 \end{pmatrix}$$

Скалярний добуток векторів. На множині векторів однакових розмірів визначена операція скалярного добутку. Нехай

$$x = (x_1, x_2, \dots, x_n)^T, y = (y_1, y_2, \dots, y_n)^T \text{ або } x = (x_1, x_2, \dots, x_n)$$

та $y = (y_1, y_2, \dots, y_n)$, тоді скалярний добуток векторів x та y позначають через (x, y) і обчислюють за формулою

$$(x, y) = x_1 y_1 + x_2 y_2 + \dots + x_n y_n = \sum_{p=1}^n x_p y_p.$$

Скалярний добуток задовольняє наступним властивостям:

1. $(x, x) \geq 0$, $(x, x) = 0 \Leftrightarrow x = 0$;

Властивість комутативності: $x \cdot y = y \cdot x$

2. $(x, y) = (y, x)$ для будь-яких векторів x, y ;

Властивість дистрибутивності: $(x + y)z = x \cdot z + y \cdot z$

3. $(x + y, z) = (x, z) + (y, z)$ для будь-яких векторів x, y і z .

Властивість асоціативності: $(\lambda x \cdot y) = \lambda(x \cdot y)$

4. $(\lambda x, y) = \lambda(x, y)$ для будь-яких векторів x, y і числа λ .

Приклад 3.30. Обчислити скалярний добуток векторів $x = (x_1, x_2, x_3, x_4)$ та

$$y = (y_1, y_2, y_3, y_4)$$

Розв'язок. $(x, y) = (x_1 \cdot y_1 + x_2 \cdot y_2 + x_3 \cdot y_3 + x_4 \cdot y_4)$

Ортогональні вектори

Вектори x та y називають ортогональними, якщо їх скалярний добуток дорівнює нулю, тобто $(x, y) = 0$.

Нульовий вектор 0 ортогональний будь-якому вектору x , тобто $(0, x) = 0$.

Приклад 3.31. Перевірити, чи є ортогональними вектори

$$x = (5, -2, 2) \text{ і } y = (-10, 7, 32)$$

Розв'язок. Обчислимо скалярний добуток двох векторів.

$$(x, y) = (-5 \cdot 10 - 2 \cdot 7 + 2 \cdot 32) = (-50 - 14 + 64) = 0$$

Отже, вектори x та y є ортогональними.

Вектор $x = (x_1, x_2, \dots, x_n)$ називають *невід'ємним*, якщо всі його компоненти $x_i \geq 0, i = 1, 2, \dots, n$.

Вектор 0 згідно цього визначення є *невід'ємним*.

Вектор $x = (x_1, x_2, \dots, x_n)$ називають *додатним*, якщо його компоненти $x_i > 0, i = 1, 2, \dots, n$.

3.10.3. Різновидності квадратних матриць

Матрицю A розміру $n \times n$ називають *квадратною*, а число n – порядком цієї матриці. Для множини квадратних матриць порядку n виконуються всі операції, описані вище.

Для будь-яких двох квадратних матриць того самого порядку n завжди визначений їх добуток.

Матрицю $A^k = \underbrace{A \cdot A \cdot A \cdots A}_k$ називають *k -тим степенем* матриці A , де

k – ціле число.

Приклад 3.32.

Нехай дана 3×3 -матриця $A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$. Обчислити 2-ий

ступінь матриці A

Розв'язок

$$A^2 = A \cdot A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} =$$

$$= \begin{pmatrix} a_{11}a_{11} + a_{12}a_{21} + a_{13}a_{31} & a_{11}a_{12} + a_{12}a_{22} + a_{13}a_{32} & a_{11}a_{13} + a_{12}a_{23} + a_{13}a_{33} \\ a_{21}a_{11} + a_{22}a_{21} + a_{23}a_{31} & a_{21}a_{12} + a_{22}a_{22} + a_{23}a_{32} & a_{21}a_{13} + a_{22}a_{23} + a_{23}a_{33} \\ a_{31}a_{11} + a_{32}a_{21} + a_{33}a_{31} & a_{31}a_{12} + a_{32}a_{22} + a_{33}a_{32} & a_{31}a_{13} + a_{32}a_{23} + a_{33}a_{33} \end{pmatrix}$$

Справедливі правила:

$$1. A^m \cdot A^k = A^{m+k},$$

$$2. (A^m)^k = A^{m \cdot k}$$

Нижньотрикутна матриця

Матрицю $A = (a_{ij})$ називають *нижньотрикутною*, якщо $a_{ij} = 0$ при $i < j$.

$$\text{Наприклад, } A = \begin{pmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}.$$

Верхньотрикутна матриця

Матрицю $B = (b_{ij})$ називають *верхньотрикутною*, якщо $b_{ij} = 0$ при $i > j$.

$$\text{Наприклад, } B = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ 0 & b_{22} & b_{23} & b_{24} \\ 0 & 0 & b_{33} & b_{34} \\ 0 & 0 & 0 & b_{44} \end{pmatrix}.$$

Діагональні матриці

Матрицю $C = (c_{ij})$ називають *діагональною*, якщо $c_{ij} \neq 0$ при $i = j$, а інші елементи дорівнюють нулю.

Діагональну матрицю записують як $C = \text{diag}(c_{11}, c_{22}, c_{33}, \dots, c_{nn})$.

$$\text{Наприклад, } C = \text{diag}(c_{11}, c_{22}, c_{33}, c_{44}) = \begin{pmatrix} c_{11} & 0 & 0 & 0 \\ 0 & c_{22} & 0 & 0 \\ 0 & 0 & c_{33} & 0 \\ 0 & 0 & 0 & c_{44} \end{pmatrix}.$$

Діагональну матрицю називають *одиничною* E , якщо $e_{ij} = 1$ при $i = j$, а інші елементи дорівнюють нулю.

$$\text{Наприклад, } E = \text{diag}(1,1,1,1) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Симетричні матриці

Матрицю A називають *симетричною*, якщо $A = A^T$, тобто якщо $(a_{ij}) = (a_{ji})$

$$\text{Наприклад, } A = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 1 \\ 3 & 2 & 1 & 0 \end{pmatrix}$$

3.10.4. Визначник квадратної матриці та його властивості

Визначник (або *детермінант*) – одне з основних понять лінійної алгебри. Визначник матриці є многочленом від елементів квадратної матриці. Визначник $\det A$ квадратної матриці $A = (a_{ij})$ порядку n вводять рекурентно [41].

1. *Визначником матриці першого порядку* $A = (a_{11})$ назвемо число $\det A = (a_{11})$.

2. *Визначником матриці другого порядку*

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}.$$

назвемо число, яке дорівнює

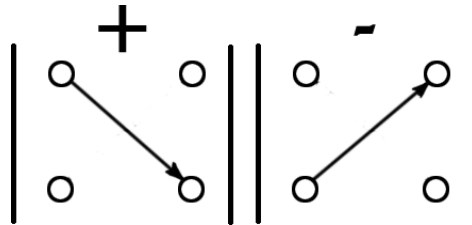
$$\det A = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}M_{11} - a_{12}M_{12},$$

$M_{11} = a_{22}$ – визначник (мінор), отриманий викреслюванням з матриці A рядка 1 і стовпця 1.

$M_{12} = a_{21}$ – визначник (мінор), отриманий викреслюванням з матриці A рядка 1 і стовпця 2.

$$\text{Тоді } \det A = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}.$$

Геометрична інтерпретація правила обчислення визначника другого порядку:



Визначником матриці третього порядку

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

назвемо число, яке дорівнює

$$\det A = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}M_{11} - a_{12}M_{12} + a_{13}M_{13}, \quad (3.47)$$

$$\text{де } M_{11} = \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} = a_{22}a_{33} - a_{23}a_{32},$$

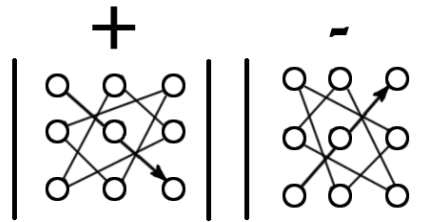
$$M_{12} = \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} = a_{21}a_{33} - a_{23}a_{31},$$

$$M_{13} = \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} = a_{21}a_{32} - a_{22}a_{31}.$$

Підставивши отримані вирази для обчислення M_{11} , M_{12} , M_{13} у формулу (3.47), одержимо кінцевий вираз для визначника:

$$\begin{aligned} \det A = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} &= a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{21}a_{32}a_{13} - \\ &- a_{13}a_{22}a_{31} - a_{11}a_{32}a_{23} - a_{21}a_{12}a_{33}. \end{aligned}$$

Геометрична інтерпретація правила обчислення визначника третього порядку:



Визначник $n \times n$ -матриці

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}$$

будемо обчислювати за формулою:

$$\det A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} = a_{11}M_{11} - a_{12}M_{12} + \dots + (-1)^n a_{1n}M_{1n} \quad (3.48)$$

де $M_{11}, M_{12}, \dots, M_{1j}, \dots, M_{1n}$ – визначники (мінори), отримані викреслюванням з матриці A рядка 1 і стовпця j .

Мінор. Узагальнюючи розглянуте, введемо поняття мінору й алгебраїчного доповнення.

Мінором M_{ij} елемента a_{ij} $n \times n$ -матриці A називають визначник $(n-1)$ -го порядку, отриманий з матриці A викреслюванням i -го рядка та j -го стовпця.

Алгебраїчне доповнення

Алгебраїчним доповненням A_{ij} елемента a_{ij} $n \times n$ -матриці A називають мінор M_{ij} , узятий зі знаком $(-1)^{i+j}$: $A_{ij} = (-1)^{i+j} M_{ij}$.

Використовуючи алгебраїчне доповнення, формулу (3.48) можна переписати у вигляді:

$$\det A = a_{11}M_{11} - a_{12}M_{12} + \dots + (-1)^n a_{1n}M_{1n} = \\ = a_{11}A_{11} + a_{12}A_{12} + \dots + a_{1n}A_{1n}.$$

Формула розкладання визначника по першому рядку:

$$\det A = \sum_{j=1}^n (-1)^{j+1} a_{1j}M_{1j} = \sum_{j=1}^n a_{1j}A_{1j}.$$

Формула розкладання визначника по першому стовпцю:

$$\det A = \sum_{i=1}^n (-1)^{i+1} a_{i1}M_{i1} = \sum_{i=1}^n a_{i1}A_{i1}$$

Властивості визначника

1. Визначник з нульовим рядком або нульовим стовпцем дорівнює нулю.

$$\det A = \begin{vmatrix} 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = 0, \quad \det B = \begin{vmatrix} 0 & b_{12} & b_{12} \\ 0 & b_{22} & b_{23} \\ 0 & b_{32} & b_{33} \end{vmatrix} = 0$$

Для доведення цієї властивості достатньо розкласти визначник по нульовому рядку або по нульовому стовпцю.

2. Множення визначника на число рівносильне множенню будь-якого рядка або стовпця визначника на це число.

$$\alpha \cdot \det A = \alpha \cdot \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = \begin{vmatrix} \alpha \cdot a_{11} & a_{12} & a_{13} \\ \alpha \cdot a_{21} & a_{22} & a_{23} \\ \alpha \cdot a_{31} & a_{32} & a_{33} \end{vmatrix} = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ \alpha \cdot a_{21} & \alpha \cdot a_{22} & \alpha \cdot a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

Помножимо будь-який рядок або стовпець початкового визначника на число, розкладемо визначник по цьому рядку або стовпцю, винесемо це число за дужки й згорнемо вираз, що залишився в дужках, у початковий визначник.

3. При транспонуванні матриці величина її визначника не змінюється:

$$\det A = \det A^T$$

Розкладемо визначник $\det A$ по 1-ому рядку, транспонуємо його. Розкладемо отриманий визначник $\det A^T$ по 1-му стовпцю. З доведеної вище теореми випливає, що результат буде однаковий.

4. При перестановці двох рядків або стовпців визначник змінює знак.

$$\text{У визначнику } \det A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}$$

переставимо, наприклад, перший і другий рядки. Одержимо

$$\det A = \begin{vmatrix} a_{21} & a_{22} & \dots & a_{2n} \\ a_{11} & a_{12} & \dots & a_{1n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}$$

Розкладемо визначник $\det A_1$, по другому рядку, а визначник $\det A_2$ – по першому. Одержимо

$$\det A_1 = -a_{21}M_{21} + a_{22}M_{22} + \dots + (-1)^{n+2} a_{2n}M_{2n},$$

$$\det A_2 = a_{21}M_{21} - a_{22}M_{22} + \dots + (-1)^{n+1} a_{2n}M_{2n},$$

звідки випливає $\det A_1 = -\det A_2$.

Аналогічно можна показати, що перестановка будь-яких двох рядків або будь-яких двох стовпців приводить до зміни знака визначника.

5. Визначник матриці із двома однаковими рядками (стовпцями) дорівнює нулю.

$$\det A = \begin{vmatrix} a_{11} & a_{11} \\ a_{21} & a_{21} \end{vmatrix} = a_{11}a_{21} - a_{21}a_{11} = 0$$

При перестановці двох рядків визначник змінює знак. Переставимо місцями однакові рядки. Визначник залишиться таким же. Отже, $\det A = -\det A$. Звідси випливає, що $\det A = 0$.

6. Визначник, який містить два пропорційні рядки (стовпця), дорівнює нулю.

$$\det A = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ \alpha a_{11} & \alpha a_{12} & \alpha a_{13} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = \alpha \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{11} & a_{12} & a_{13} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = \alpha \cdot 0 = 0.$$

Винесемо коефіцієнт пропорційності за знак визначника. У ньому утворюються два однакових рядки. Тому такий визначник дорівнює нулю.

7. Визначник можна розкласти на суму визначників. Представимо елементи i -го рядка визначника у вигляді суми двох доданків.

Одержимо

$$\det A = \begin{vmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ \alpha b_1 + \beta c_1 & \dots & \alpha b_n + \beta c_n \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{vmatrix}$$

де α, β – деякі коефіцієнти, які дорівнюють в окремому випадку одиниці.

Розкладемо визначник $\det A$ по i -ому рядку, використовуючи алгебраїчні доповнення, перетворимо отриману суму. Тоді

$$\begin{aligned} \det A &= \sum_{j=1}^n (\alpha b_j + \beta c_j) A_{ij} = \sum_{j=1}^n \alpha b_j A_{ij} + \sum_{j=1}^n \beta c_j A_{ij} \\ &= \alpha \sum_{j=1}^n b_j A_{ij} + \beta \sum_{j=1}^n c_j A_{ij} = \alpha \det A_1 + \beta \det A_2 \end{aligned} \quad \text{де}$$

$$\det A_1 = \begin{vmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ b_1 & \dots & b_n \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{vmatrix}, \quad \det A_2 = \begin{vmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ c_1 & \dots & c_n \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{vmatrix}$$

8. Визначник добутку квадратних матриць дорівнює добутку визначників цих матриць, тобто $\det(A \cdot B) = \det A \cdot \det B$

Через громіздкість перетворень обмежимося розглядом матриць другого порядку. Нехай

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}.$$

$$\text{Тоді } A \cdot B = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \cdot \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} = \begin{pmatrix} a_1 b_1 + a_2 b_3 & a_1 b_2 + a_2 b_4 \\ a_3 b_1 + a_4 b_3 & a_3 b_2 + a_4 b_4 \end{pmatrix}$$

Обчислимо величини визначників трьох матриць, а також величину $\det A \cdot \det B$.

$$\det A = a_1 a_4 - a_2 a_3, \quad \det B = b_1 b_4 - b_2 b_3,$$

$$\begin{aligned} \det(AB) &= (a_1 b_1 + a_2 b_3)(a_3 b_2 + a_4 b_4) - (a_1 b_2 + a_2 b_4)(a_3 b_1 + a_4 b_3) = \\ &= \underline{a_1 a_3 b_1 b_2} + a_1 a_4 b_1 b_4 + a_2 a_3 b_2 b_3 + \underline{a_2 a_4 b_3 b_4} - \underline{a_1 a_3 b_1 b_2} - \\ &- a_1 a_4 b_2 b_3 - a_2 a_3 b_1 b_4 - \underline{a_2 a_4 b_3 b_4} \end{aligned}$$

$$\begin{aligned} \det A \cdot \det B &= (a_1 a_4 - a_2 a_3)(b_1 b_4 - b_2 b_3) = \\ &= a_1 a_4 b_1 b_4 - a_1 a_4 b_2 b_3 - a_2 a_3 b_1 b_4 + a_2 a_3 b_2 b_3 \end{aligned}$$

Після скорочення подібних (вони підкреслені) одержуємо праворуч однакові вирази для $\det(AB)$ й $\det A \cdot \det B$

9. Визначник трикутної й діагональної матриць дорівнює добутку її

діагональних елементів, тобто $\det A = a_{11} \cdot a_{22} \cdot \dots \cdot a_{nn} = \prod_{k=1}^n a_{kk}$,

Обчислення визначника. Існує кілька способів обчислення величини визначника. Вибір способу диктує вид і порядок визначника. Вдало обраний спосіб дозволяє суттєво скоротити обчислення. Розглянемо їх на прикладі визначника матриці третього порядку.

Приклад 3.33. Обчислити визначник матриці

$$A = \begin{pmatrix} 2 & 2 & -1 \\ 2 & -4 & 1 \\ -2 & -2 & 2 \end{pmatrix}.$$

1-й спосіб. Цей спосіб полягає в розкладанні визначника по будь-якому рядку або стовпцю. Для прикладу розкладемо визначник заданої вище матриці A по 3-му стовпцю:

$$\begin{vmatrix} 2 & 2 & -1 \\ 2 & -4 & 1 \\ -2 & -2 & 2 \end{vmatrix} = -1(-1)^{1+3} \begin{vmatrix} 2 & -4 \\ -2 & -2 \end{vmatrix} + 1(-1)^{2+3} \begin{vmatrix} 2 & 2 \\ -2 & -2 \end{vmatrix} + \\ + 2(-1)^{3+3} \begin{vmatrix} 2 & 2 \\ 2 & -4 \end{vmatrix} = -(-4-8) - (-4+4) + 2(-8-4) = -12$$

2-й спосіб. Використання правила трикутників:

$$\begin{vmatrix} 2 & 2 & -1 \\ 2 & -4 & 1 \\ -2 & -2 & 2 \end{vmatrix} = 2(-4)2 + 2 \cdot 1(-2) + 2(-2)(-1) \\ - (-1)(-4)(-2) - 2 \cdot 2 \cdot 2 - (-2)1 \cdot 2 = -12$$

3-й спосіб. Використання властивостей визначника для перетворення його до вигляду, коли він містить рядок або стовпець із максимальною кількістю нулів. Розкладання визначника по цьому рядку (стовпцю):

$$\begin{vmatrix} 2 & 2 & -1 \\ 2 & -4 & 1 \\ -2 & -2 & 2 \end{vmatrix} \Rightarrow \text{додаємо до 1-го рядка 3-й} \Rightarrow \begin{vmatrix} 0 & 0 & 1 \\ 2 & -4 & 1 \\ -2 & -2 & 2 \end{vmatrix} \Rightarrow$$

\Rightarrow розкладемо визначник по першому рядку \Rightarrow

$$\Rightarrow 1(-1)^{1+3} \begin{vmatrix} 2 & -4 \\ -2 & -2 \end{vmatrix} = -4 - 8 = -12$$

4-й спосіб. Використання властивостей визначника для перетворення його до трикутного виду. Величина визначника обчислюється як добуток елементів, що стоять на головній діагоналі:

$$\begin{vmatrix} 2 & 2 & -1 \\ 2 & -4 & 1 \\ -2 & -2 & 2 \end{vmatrix} \Rightarrow \text{1-ий рядок множимо на -1 і додаємо до 2-го рядка,}$$

$$\text{помістивши результат на місце 2-го рядка} \Rightarrow \begin{vmatrix} 2 & 2 & -1 \\ 0 & -6 & 2 \\ -2 & -2 & 2 \end{vmatrix} \Rightarrow \text{1-й рядок}$$

$$\text{додаємо до 3-го і помістимо результат на місце 3-го рядка} \Rightarrow \begin{vmatrix} 2 & 2 & -1 \\ 0 & -6 & 2 \\ 0 & 0 & 1 \end{vmatrix} \Rightarrow$$

перемножимо елементи головної діагоналі $\Rightarrow = -12$.

3.10.5. Обернена матриця

Нехай A і B – квадратні матриці порядку n і $AB = E$, тоді матрицю B називають *оберненою* до A і позначають як A^{-1} , тобто $AA^{-1} = A^{-1}A = E$, де E – *одична матриця*.

Якщо A^{-1} існує, то матрицю A називають *невиродженою*, якщо ні, то матрицю A називають *виродженою*, або *сингулярною*. Основні властивості оберненої матриці [42]:

$$1. \det A^{-1} \det A = \det E = 1 \Rightarrow \det A^{-1} = \frac{1}{\det A}$$

$$2. (AB)^{-1} = B^{-1}A^{-1}$$

і, у загальному випадку, $(A_1A_2\dots A_k)^{-1} = A_k^{-1}A_{k-1}^{-1}\dots A_1^{-1}$.

$$3. (A^{-1})^T = (A^T)^{-1}.$$

Теорема 3.9. *Про єдиність оберненої матриці*

Кожна невивроджена квадратна матриця має обернену матрицю, і притому єдину.

Визначення. Матрицю A^* , отриману з матриці A заміною її елементів їх алгебраїчними доповненнями з наступним транспонуванням, називають матрицею, приєднаною до матриці A :

$$A^* = \begin{pmatrix} A_{11} & A_{21} & \dots & A_{n1} \\ A_{12} & A_{22} & \dots & A_{n2} \\ \dots & \dots & \dots & \dots \\ A_{1n} & A_{2n} & \dots & A_{nn} \end{pmatrix}$$

Теорема 3.10. *Про обчислення оберненої матриці*

Якщо матриця A невивроджена, то її обернена матриця має вигляд:

$$A^{-1} = \frac{A^*}{\det A} \quad (3.49)$$

Приклад 3.34.

Дана матриця $A = \begin{pmatrix} 3 & 2 & 2 \\ 1 & 3 & 1 \\ 5 & 3 & 4 \end{pmatrix}$. Знайти обернену матрицю A^{-1} .

Розв'язок: Обчислюємо визначник матриці A :

$$\det A = \begin{vmatrix} 3 & 2 & 2 \\ 1 & 3 & 1 \\ 5 & 3 & 4 \end{vmatrix} = 3 \begin{vmatrix} 3 & 1 \\ 3 & 4 \end{vmatrix} - 2 \begin{vmatrix} 1 & 1 \\ 5 & 4 \end{vmatrix} + 2 \begin{vmatrix} 1 & 3 \\ 5 & 3 \end{vmatrix} = 27 + 2 - 24 = 5$$

Знаходимо алгебраїчні доповнення елементів цього визначника:

$$\begin{aligned} A_{11} &= \begin{vmatrix} 3 & 1 \\ 3 & 4 \end{vmatrix} = 9 & A_{21} &= \begin{vmatrix} 2 & 2 \\ 3 & 4 \end{vmatrix} = -2 & A_{31} &= \begin{vmatrix} 2 & 2 \\ 3 & 1 \end{vmatrix} = -4 \\ A_{12} &= \begin{vmatrix} 1 & 1 \\ 5 & 4 \end{vmatrix} = 1 & A_{22} &= \begin{vmatrix} 3 & 2 \\ 5 & 4 \end{vmatrix} = 2 & A_{32} &= - \begin{vmatrix} 3 & 2 \\ 1 & 1 \end{vmatrix} = -1 \\ A_{13} &= \begin{vmatrix} 1 & 3 \\ 5 & 3 \end{vmatrix} = -12 & A_{23} &= - \begin{vmatrix} 3 & 2 \\ 5 & 3 \end{vmatrix} = 1 & A_{33} &= \begin{vmatrix} 3 & 2 \\ 1 & 3 \end{vmatrix} = 7 \end{aligned}$$

Отже,

$$A^{-1} = \begin{pmatrix} \frac{9}{5} & -\frac{2}{5} & -\frac{4}{5} \\ \frac{1}{5} & \frac{2}{5} & -\frac{1}{5} \\ -\frac{12}{5} & \frac{1}{5} & \frac{7}{5} \end{pmatrix}$$

3.10.6. Ортогональні матриці

Визначення. Квадратну матрицю O називають ортогональною, якщо вона задовольняє умові [43]:

$$O^T O = E \tag{3.50}$$

де E – одинична матриця.

Приклад 3.35. Найпростішою ортогональною матрицею є одинична матриця E , оскільки $E^T E = EE = E$.

$$E^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

$$E \cdot E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Приклад 3.36. Визначити, чи є матриця U ортогональною:

$$U = \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix}.$$

Розв'язок. Перевіримо виконання умови ортогональності $U^T U = E$.

1. Визначимо транспоновану матрицю $U^T = \begin{pmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{pmatrix}$.

2. Обчислимо добуток матриць

$$U^T U = \begin{pmatrix} \cos \varphi & \sin \varphi \\ -\sin \varphi & \cos \varphi \end{pmatrix} \cdot \begin{pmatrix} \cos \varphi & -\sin \varphi \\ \sin \varphi & \cos \varphi \end{pmatrix} =$$

$$\begin{pmatrix} \cos^2 \varphi + \sin^2 \varphi & \cos \varphi \sin \varphi - \cos \varphi \sin \varphi \\ \cos \varphi \sin \varphi - \cos \varphi \sin \varphi & \cos^2 \varphi + \sin^2 \varphi \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = E$$

Таким чином, справедливе співвідношення $U^T U = E$, отже, матриця U – ортогональна.

Властивості ортогональних матриць

Властивість 1. Визначник ортогональної матриці O може мати одне із двох можливих значень: $\det O = \pm 1$.

Згідно з рівністю (3.50) маємо $\det(O^T O) = \det E$.

Оскільки визначник добутку матриць дорівнює добутку їх визначників, а при транспонуванні матриці її визначник не змінюється, одержимо $\det(O^T O) = \det O^T \det O = (\det O)^2$.

Оскільки $\det E = 1$, то й $(\det O)^2 = 1$. Отже, $\det O = \pm 1$.

Властивість 2. Матриця, обернена до ортогональної матриці O , співпадає з її транспонованою матрицею, тобто $O^{-1} = O^T$.

Згідно із властивістю 1, ортогональна матриця не вироджена, і тому має обернену матрицю O^{-1} . Помноживши (3.50) на O^{-1} , одержуємо $(O^T O)O^{-1} = EO^{-1}$, звідки $O^T(OO^{-1}) = O^{-1}$. Але $OO^{-1} = E$, тому $O^T = O^{-1}$.

Властивість 3. Добуток ортогональної матриці O на транспоновану до неї дорівнює одиничній матриці, тобто $OO^T = E$.

Згідно із властивістю 2 і визначенням оберненої матриці, $OO^T = OO^{-1} = E$.

Властивість 4. Матриця, транспонована до ортогональної матриці, теж є ортогональною.

Доведемо для довільної ортогональної матриці O рівність

$$(O^T)^T O^T = E, \quad (3.51)$$

що є записом співвідношення (3.50) для матриці O^T . Тому що, згідно із властивістю операції транспонування, $(O^T)^T = O$, рівність (3.50) еквівалентна рівності $OO^T = E$, яка є вірною відповідно до властивості 3.

Властивість 5. Добуток двох ортогональних матриць O та Q одного порядку є ортогональною матрицею.

Для доведення достатньо перевірити виконання рівності (3.50) для матриці OQ :

$$(OQ)^T OQ = (Q^T O^T)OQ = Q^T(O^T O)Q = Q^T E Q = Q^T Q = E$$

Властивість 6. Матриця, обернена до ортогональної матриці, теж є ортогональною.

Згідно з властивістю (3.50), ортогональна матриця не вироджена, а тому має обернену. Згідно із властивістю 2, матриця, обернена до ортогональної, співпадає із транспонованою. Нарешті, згідно з властивістю 4, матриця, транспонована до ортогональної, є ортогональною.

3.10.7. Лінійні простори

Лінійним (або векторним) простором називають множину деяких об'єктів L (далі ці об'єкти будемо називати *векторами*), для яких визначено дві операції [44]:

1. Операція множення довільного вектора $x \in L$ на довільне число a , у результаті якої отримуємо деякий вектор

$$z = ax \in L.$$

2. Операція додавання двох довільних векторів $x \in L$ і $y \in L$, у результаті якої отримуємо деякий вектор

$$s = x + y \in L.$$

При цьому дані операції задовольняють наступним властивостям:

1. Для будь-яких векторів $x, y \in L$ виконується умова комутативності:

$$x + y = y + x$$

2. Для будь-яких векторів $x, y, z \in L$ виконується умова асоціативності:

$$x + (y + z) = (x + y) + z$$

3. Існує нульовий вектор 0 такий, що для будь-якого вектора $x \in L$ виконується

$$x + 0 = 0 + x = x$$

4. Для будь-якого вектора $x \in L$ існує вектор $y \in L$, такий що

$$x + y = y + x = 0.$$

Вектор y називають оберненим до вектора x і позначають $y = -x$.

5. Для будь-якого вектора $x \in L$ виконується

$$1x = x$$

Тобто при множенні вектора на число 1 отримуємо той самий вектор.

6. Для будь-якого вектора $x \in L$ і будь-яких чисел a, b виконується

$$(ab)x = a(bx)$$

7. Для будь-яких векторів $x, y \in L$ і будь-якого числа a виконується

$$a(x + y) = ax + ay$$

8. Для будь-якого вектора $x \in L$ і будь-яких чисел a, b виконується

$$(a + b)x = ax + bx$$

3.10.8. Базис. Координати

Нехай дано вектори x_1, x_2, \dots, x_n й числа a_1, a_2, \dots, a_n , тоді вираз

$$a_1x_1 + a_2x_2 + \dots + a_nx_n$$

називають лінійною комбінацією векторів x_1, x_2, \dots, x_n .

Якщо вектор y такий, що $y = a_1x_1 + a_2x_2 + \dots + a_nx_n$, то говоримо, що вектор y можна *лінійно виразити* через вектори x_1, x_2, \dots, x_n .

Нехай 0 – нульовий вектор у просторі. Вектори x_1, x_2, \dots, x_n називають лінійно незалежними, якщо вираз

$$0 = a_1x_1 + a_2x_2 + \dots + a_nx_n$$

можливий тільки при $a_1 = \dots = a_n = 0$.

Якщо вектори x_1, x_2, \dots, x_n такі, що існують числа a_1, a_2, \dots, a_n , не всі одночасно рівні нулю, для яких

$$0 = a_1x_1 + a_2x_2 + \dots + a_nx_n,$$

то говоримо, що вектори x_1, x_2, \dots, x_n – *лінійно залежні*.

Базисом у лінійному просторі L називають набір векторів $B = \{e_1, e_2, \dots, e_n\}$, такий що:

1. Вектори e_1, e_2, \dots, e_n – лінійно незалежні.

2. Будь-який вектор $\alpha \in L$ можна лінійно виразити через вектори x_1, x_2, \dots, x_n , тобто існують числа a_1, a_2, \dots, a_n , такі що

$$x = a_1e_1 + a_2e_2 + \dots + a_n e_n.$$

Числа a_1, a_2, \dots, a_n називають *координатами вектора x* у базисі e_1, e_2, \dots, e_n .

Число n (кількість базисних векторів) називають *розмірністю простору L* .

У n -вимірному просторі дійсних чисел R^n є базис, названий *природним або стандартним*.

Цей базис утворений одиничними векторами

$$e_1 = (1, 0, 0, \dots, 0), e_2 = (0, 1, 0, \dots, 0), \dots, e_n = (0, 0, 0, \dots, 1).$$

Особливість цього базису полягає в тому, що координатами деякого вектора $z = (z_1, z_2, \dots, z_n)$ у цьому базисі будуть самі числа z_1, z_2, \dots, z_n .

Систему лінійно незалежних векторів $\{b_1, b_2, \dots, b_n\}$ з R^n називають *ортогональним базисом R^n* , якщо вектори $\{b_1, b_2, \dots, b_n\}$ попарно ортогональні один до одного, і кожний з них має довжину, що дорівнює одиниці, тобто

$$(b_i, b_j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

Зокрема, стандартний базис (e_1, e_2, \dots, e_n) є ортогональним.

$$e_1 = (1, 0, 0, \dots, 0), e_2 = (0, 1, 0, \dots, 0), \dots, e_n = (0, 0, 0, \dots, 1)$$

Визначення для $\{b_1, b_2\}$ й $\{b_1, b_2, b_3\}$. Два вектори називають ортогональними, якщо кут між ними дорівнює прямому куту, тобто 90° .

Позначення: $b_1 \perp b_2$ – вектори b_1 й b_2 ортогональні.

$b_1 \perp b_2 \perp b_3, b_1 \perp b_3$ – вектори b_1, b_2 і b_3 попарно ортогональні

Визначення. n векторів $\{b_1, b_2, \dots, b_n\}$ називають ортогональними, якщо ці вектори попарно ортогональні один одному, тобто $b_1 \perp b_2, b_2 \perp b_3, \dots, b_{n-1} \perp b_n, b_1 \perp b_3, \dots, b_1 \perp b_n$.

$b_i \perp b_j$ якщо $\forall (b_i, b_j) = k$, $k \neq 0$ при $i = j$, $k = 0$ при $i \neq j$.

Визначення. n векторів $\{b_1, b_2, \dots, b_n\}$ називають ортонормованими, якщо вони ортогональні, і довжини всіх векторів дорівнюють одиниці:
 $|b_1| = |b_2| = \dots = |b_n| = 1$.

3.10.9. Норми векторів і матриць

Норма вектора. У просторі R^n скалярною характеристикою вектора є норма, яка дає можливість порівнювати вектори за величиною, і яка, у загальному випадку, узагальнює поняття довжини вектора. Поняття норми визначають аксіоматично [45].

Нормою вектора x називають дійсне число $\|x\|$, що задовольняє наступним умовам:

1. $\|x\| \geq 0$, $\|x\| = 0$ тільки при $x = 0$ (невід'ємність);
2. $\|yx\| = \|y\| \|x\|$, для довільних x і y .
3. $\|x + y\| \leq \|x\| + \|y\|$ для довільних x і y .

Норму вектора визначають у різний спосіб залежно від цілей дослідження й умов задачі, але при будь-якому визначенні норма повинна задовольняти умовам 1), 2), 3), вказаним вище.

Найпоширенішими в практиці обчислень є наступні норми:

1. Природна норма $\|x\| = \sqrt{(x, x)}$, $x \in L$,

де (x, x) – скалярний добуток вектора x самого на себе:

$$(x, x) = x_1^2 + x_2^2 + \dots + x_n^2.$$

2. Гельдерові норми n -вимірних векторів (сімейство):

$$\|x\|_p = \sqrt[p]{\sum_{i=1}^n |x_i|^p},$$

де $p \geq 1$ – натуральне число.

Зокрема:

$$\|x\|_1 = \sum_{i=1}^n |x_i|$$

$$\|x\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2} \text{ – евклідова норма,}$$

.....

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i| \text{ – гранична норма при } p \rightarrow \infty$$

Просте пояснення змісту норми

Так що ж таке норма?

Ми бачимо, що задані аксіоми, які повинні виконуватися, але навіщо ця норма потрібна в принципі?

Є кілька розповсюджених варіантів відповіді:

1. Норма матриці або вектора є деяким числом, відмінним від нуля. Норма потрібна для того, щоб порівнювати, яка матриця або вектор «більше», а яка (який) «менше».

2. Норма матриці або вектора є деякою числовою характеристикою відхилення їх від нуля (але це швидше стосується норми взагалі).

Дві векторні норми $\|\cdot\|_1$ й $\|\cdot\|_2$ називають *еквівалентними*, якщо існують такі додатні дійсні числа c_1 й c_2 , що для будь-якого вектора x виконуються нерівності

$$c_1 \|x\|_1 \leq \|x\|_2 \leq c_2 \|x\|_1, \text{ причому } c_1 \text{ й } c_2 \text{ не залежать від вибору } x.$$

Нерівність Коші-Буняковського

Нехай даний лінійний простір L зі скалярним добутком двох векторів (x, y) . Нехай $\|x\|$ – норма, породжена скалярним добутком, тобто $\|x\| = \sqrt{(x, x)} \forall x \in L$. Тоді для будь-яких x і y маємо: $|(x, y)| \leq \|x\| \cdot \|y\|$.

Норма матриць

Розглянемо тепер множину всіх $n \times n$ -матриць. Для матриць поняття норми також вводиться аксіоматично.

Нормою квадратної матриці A називають дійсне число $\|A\|$, що задовольняє наступним умовам:

1. $\|A\| \geq 0$, $\|A\| = 0$ тільки коли $A = 0$ (нульова матриця)
2. $\|\alpha A\| = |\alpha| \cdot \|A\|$, α – довільне число (скаляр).
3. $\|A + B\| \leq \|A\| + \|B\|$ для довільних A і B .
4. $\|AB\| \leq \|A\| \cdot \|B\|$ для довільних A і B .

Якщо виконується також і четверта властивість, норму називають *мультіплікативною*.

Якщо в математичному виразі *спільно використовують матриці та вектори*, то матрична норма повинна співвідноситися з векторною, тобто матрична норма повинна задовольняти *умові погодженості*, а саме, якщо для будь-яких x та A справедлива нерівність $\|Ax\| \leq \|A\| \cdot \|x\|$, тоді норма матриці $\|A\|$ погоджена з нормою вектора $\|x\|$.

Матричні норми, відповідні до векторних норм, мають вигляд:

$$\|A\|_l = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}| \quad - \quad l\text{-норма,}$$

$$\|A\|_m = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}| \quad - \quad m\text{-норма,}$$

$$\|A\|_2 = \sqrt{\sum_{i,j} |a_{ij}|^2} \quad - \quad \text{норма Фробеніуса}$$

$$\|A\|_p = \sqrt[p]{\sum |a_{ij}|^p} \quad - \quad (p > 0) \quad - \quad p\text{-норма.}$$

Матричну норму можна інтерпретувати як максимальну довжину вектора, отриманого в результаті перетворення A до вектора.

Зазначимо, що з усіх наведених матричних норм найменше значення має m -норма.

Число λ будемо називати власним значенням матриці A , яке відповідає власному вектору x за умови, що справедлива рівність:

$$Ax = \lambda x$$

Множину всіх власних значень матриці $(\lambda_1, \lambda_2, \dots, \lambda_n)$ називають спектром матриці A і позначають як $S(A)$.

Спектральний радіус $r(A)$ квадратної матриці A дорівнює максимальному по модулю власному значенню матриці A .

Сингулярним числом σ називають власне значення матриці $A^T A$.

Вираз $Ax = \lambda x$ перепишемо у вигляді $(A - \lambda E)x = 0$. Одержане рівняння має нетривіальний розв'язок у випадку, коли $\det(A - \lambda E) = 0$. За умови, що матриця A має розмірність $n \times n$, отримуємо алгебраїчне рівняння n -го степеня відносно λ . Таким чином, корені даного рівняння і будуть власними значеннями матриці A .

Контрольні запитання

1. Знайти $A \cdot B$ за умови, що $A = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 1 & -1 \end{pmatrix}$, $B = (2 \ 1 \ 3)^T$
2. Знайти $5 \cdot C$ за умови, що $C = (12 \ 11.3 \ 3.71 \ -121)^T$
3. Перевірити, чи є ортогональними вектори $A = (3, -4, 2)$ та $B = (2, 3, 3)$.
4. Запишіть властивості скалярного добутку векторів.
5. Визначити l -норму матриці $A = \begin{pmatrix} 1 & 4 \\ 2 & 3 \end{pmatrix}$.

3.11. Розв'язування систем лінійних алгебраїчних рівнянь

3.11.1. Основні визначення

$$\text{Систему виду} \quad \begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2, \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m. \end{cases} \quad (3.52)$$

де a_{ij} ($1 \leq i \leq m, 1 \leq j \leq n$) та b_1, \dots, b_m – задані числа, x_1, \dots, x_n – невідомі, називають *системою лінійних алгебраїчних рівнянь (СЛАР)* [46].

Матричне представлення системи лінійних рівнянь

Лінійна система (3.52) має також матричну форму запису:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix} \text{ або } Ax = b,$$

де $A = (a_{ij})$ – $m \times n$ -матриця системи, $x = (x_1, x_2, \dots, x_n)^T$ – вектор-стовпець невідомих, $b = (b_1, b_2, \dots, b_m)^T$ – вектор-стовпець вільних коефіцієнтів.

Векторне представлення системи лінійних рівнянь

$$(a_i, x) = a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n$$

$$(a_i, x) = b_i \Rightarrow \sum_{p=1}^n a_{ip}x_p = b_i, \quad i = 1, 2, \dots, m,$$

$$\text{або} \quad \begin{cases} (a_1, x) = b_1, \\ (a_2, x) = b_2, \\ \dots \\ (a_m, x) = b_m. \end{cases}$$

де $x = (x_1, \dots, x_n)$ – вектор рядок, $a_i = (a_{i1}, a_{i2}, \dots, a_{in})$ – вектор-рядок, $i = 1, 2, \dots, m$

3.11.2. Види систем лінійних алгебраїчних рівнянь

Однорідна система

Визначення. Систему лінійних алгебраїчних рівнянь

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix}$$

називають *однорідною*, якщо всі $b_i, i = 1, 2, \dots, m$ дорівнюють нулю.

$$Ax = 0,$$

де $A = (a_{ij})$ – $m \times n$ -матриця системи,

$x = (x_1, x_2, \dots, x_n)^T$ – вектор-стовпець невідомих.

Неоднорідна система

Визначення. Систему лінійних алгебраїчних рівнянь

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix}$$

у випадку, коли $b \neq 0$, називають *неоднорідною*.

$$Ax = b,$$

де $A = (a_{ij})$ – $m \times n$ -матриця системи,

$x = (x_1, x_2, \dots, x_n)^T$ – вектор-стовпець невідомих,

$b = (b_1, b_2, \dots, b_m)^T$ – вектор-стовпець вільних коефіцієнтів.

3.11.3. Сумісні й несумісні системи алгебраїчних рівнянь

Визначення. Систему лінійних алгебраїчних рівнянь $Ax = b$ називають

сумісною, якщо існує вектор $x^0 = (x_1^0, x_2^0, \dots, x_n^0)^T$ такий, що $Ax^0 = b$, тоді

вектор x^0 називають *розв'язком системи* [47].

Визначення. Якщо для системи $Ax = b$ не існує вектора $x^0 = (x_1^0, x_2^0, \dots, x_n^0)^T$ такого, що $Ax^0 = b$, то систему називають *несумісною*.

3.11.4. Множина розв'язків системи алгебраїчних рівнянь

Множина розв'язків системи $Ax = b$ характеризується в такий спосіб.

Порожня множина розв'язків

Якщо множина розв'язків алгебраїчних рівнянь порожня, то в цьому випадку система *несумісна*.

Множина містить єдиний розв'язок

Якщо множина складається з єдиного розв'язку, то система сумісна й називається *визначеною*.

Множина містить більше, ніж один розв'язок

Якщо множина включає більше, ніж один розв'язок, то систему називають *невизначеною*.

3.11.5. Розширена матриця системи лінійних алгебраїчних рівнянь

Якщо приєднати стовпець b до матриці A , то отриману матрицю B називають *розширеною* матрицею системи $Ax = b$.

$$B = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_m \end{pmatrix}.$$

Умови сумісності для лінійної системи алгебраїчних рівнянь $Ax = b$ визначаються теоремою Кронекера-Капеллі.

3.11.6. Ранг матриці (системи)

Визначення. Рангом матриці A , що складається з m рядків та n стовпців, називають максимальне число лінійно незалежних рядків (стовпців).

Кілька рядків називають лінійно незалежними, якщо жоден з них не можна виразити лінійно через інші.

Ранг системи завжди дорівнює рангу матриці.

Ранг матриці – найвищий з порядків мінорів цієї матриці, відмінних від нуля.

Формальне визначення рангу матриці

Нехай $A_{m \times n}$ – прямокутна матриця. Тоді за визначенням рангом матриці A є:

1. $\text{rang } A = 0$, якщо матриця A нульова;
2. $\text{rang } A = r : \exists M_r \neq 0, \forall M_{r+1} = 0$, де M_r – базисний мінор матриці A порядку r , а M_{r+1} – мінор $(r+1)$ порядку, що оточує його (якщо він існує).

3.11.7. Базисний мінор

Базисний мінор матриці A – будь-який ненульовий мінор M_r матриці A порядку r , де $r = \text{rang } A$.

Рядки й стовпці, на перетині яких міститься базисний мінор, називають базисними рядками й стовпцями (вони визначені неоднозначно через неоднозначність базисного мінору) [48].

Теорема 3.11. *Про базисний мінор*

Нехай $r = \text{rang } A$ і M_r – базисний мінор матриці A , тоді:

1. базисні рядки й базисні стовпці лінійно незалежні;
2. будь-який рядок (стовпець) матриці A є лінійною комбінацією базисних рядків (стовпців).

Приклад 3.37. Нехай дана матриця B .

$$B = \begin{pmatrix} 2 & 3 & 1 & 2 \\ 0 & 2 & -1 & 1 \\ 4 & 0 & 5 & 1 \end{pmatrix}$$

Знайти базисний мінор матриці. На матриці $A_{m \times n}$ існують мінори $k \leq \min(m, n)$. Оскільки $m = 3$, $n = 4$ то $k \leq 3$. Почнемо з $k = 3$

$$B = \begin{pmatrix} 2 & 3 & 1 & 2 \\ 0 & 2 & -1 & 1 \\ 4 & 0 & 5 & 1 \end{pmatrix}$$

$$m_1 = \begin{vmatrix} 3 & 1 & 2 \\ 2 & -1 & 1 \\ 0 & 5 & 1 \end{vmatrix} = 0$$

$$B = \begin{pmatrix} 2 & 3 & 1 & 2 \\ 0 & 2 & -1 & 1 \\ 4 & 0 & 5 & 1 \end{pmatrix}$$

$$m_2 = \begin{vmatrix} 2 & 1 & 2 \\ 0 & -1 & 1 \\ 4 & 5 & 1 \end{vmatrix} = 0$$

$$B = \begin{pmatrix} 2 & 3 & 1 & 2 \\ 0 & 2 & -1 & 1 \\ 4 & 0 & 5 & 1 \end{pmatrix}$$

$$m_3 = \begin{vmatrix} 2 & 3 & 2 \\ 0 & 2 & 1 \\ 4 & 0 & 1 \end{vmatrix} = 0$$

$$B = \begin{pmatrix} 2 & 3 & 1 & 2 \\ 0 & 2 & -1 & 1 \\ 4 & 0 & 5 & 1 \end{pmatrix}$$

$$m_4 = \begin{vmatrix} 2 & 3 & 1 \\ 0 & 2 & -1 \\ 4 & 0 & 5 \end{vmatrix} = 0$$

Усі мінори третього порядку дорівнюють нулю. Продовжимо для $k = 2$.

Перший же мінор другого порядку не дорівнює нулю.

Тому $r = \text{rang } B = 2$

$$B = \begin{pmatrix} 2 & 3 & 1 & 2 \\ 0 & 2 & -1 & 1 \\ 4 & 0 & 5 & 1 \end{pmatrix}$$

$$m_4 = \begin{vmatrix} 2 & 3 \\ 0 & 2 \end{vmatrix} = 2 \cdot 2 - 3 \cdot 0 = 4$$

Методи знаходження рангу матриці

1. Метод елементарних перетворень

Ранг матриці дорівнює числу ненульових рядків у матриці після приведення її до східчастої форми за допомогою елементарних перетворень над рядками матриці.

2. Метод оточуючих мінорів

А) Нехай у матриці A знайдений ненульовий мінор r -го порядку M_r .

Б) Розглянемо всі мінори $(r + 1)$ -го порядку, що включають у себе (що оточують) мінор M_r ; якщо всі вони дорівнюють нулю, то ранг матриці дорівнює r .

В) А якщо не дорівнюють нулю, то серед оточуючих мінорів знайдеться ненульовий, і вся процедура повторюється.

3.11.8. Теорема Кронекера-Капеллі

Теорема. 3.12. *Про сумісну систему лінійних алгебраїчних рівнянь*

Система лінійних алгебраїчних рівнянь $Ax = b$ сумісна тоді й тільки тоді, коли ранг її основної матриці $\text{rang } A$ дорівнює рангу її розширеної матриці $\text{rang } B$, причому система має єдиний розв'язок, якщо ранг дорівнює числу невідомих, і нескінченну множину розв'язків, якщо ранг менше числа невідомих.

Наслідки

1. Кількість головних змінних СЛАР дорівнює рангу системи.
2. Спільна СЛАР буде *визначена* (її розв'язок єдиний), якщо ранг системи дорівнює числу всіх її змінних [49].

3.11.9. Системи лінійних алгебраїчних рівнянь із квадратною матрицею

Нехай матриця системи $Ax = b$ квадратна, $n = m$.

Нехай також матриця A не вироджена, тобто $\det A \neq 0$ (ця умова забезпечує сумісність і визначеність). Тоді єдиний розв'язок x_1, x_2, \dots, x_n може бути отриманий за формулами Крамера [50]:

$$x_1 = \frac{\Delta_{x_1}}{\det A}, \dots, x_n = \frac{\Delta_{x_n}}{\det A}, \text{ де}$$

$$\Delta_{x_1} = \begin{vmatrix} b_1 & a_{12} & \dots & a_{1n} \\ b_2 & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ b_n & a_{n2} & \dots & a_{nn} \end{vmatrix}, \dots, \Delta_{x_n} = \begin{vmatrix} a_{11} & a_{12} & \dots & b_1 \\ a_{21} & a_{22} & \dots & b_2 \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & b_n \end{vmatrix}$$

При великих n ці формули не використовують, тому що число арифметичних операцій зростає зі зростанням розмірності й становить $O(n) = n^2 n!$.

3.11.10. Однорідні системи лінійних алгебраїчних рівнянь

Нехай у системи $Ax = b$ всі $b_i, i = 1, 2, \dots, m$ дорівнюють 0, тоді система однорідна, тобто $Ax = 0$, де $A = (a_{ij})$, x – шуканий вектор.

Якщо $\det A \neq 0$, то в силу формул Крамера система $Ax = 0$ має єдиний розв'язок $x = 0$, який називають тривіальним.

Для того, щоб система $Ax = 0$ мала й ненульові розв'язки, необхідно, щоб $\det A = 0$. У цьому випадку сукупність усіх розв'язків системи $Ax = 0$ утворює векторний простір, який називають *простором розв'язків*.

Властивості й розмірність цього простору визначаються теоремами.

Теорема 3.13. *Про лінійний розв'язок однорідних систем*

Нехай $(x)_1, (x)_2, \dots, (x)_k$ – розв'язок однорідної системи $Ax = 0$ й c_1, c_2, \dots, c_k – довільні константи. Тоді $x_{OO} = c_1(x)_1 + c_2(x)_2 + \dots + c_k(x)_k$ також є розв'язком розглянутої системи.

Теорема 3.14. *Про структуру загального розв'язку*

Нехай $r = \text{rang } A$, тоді:

- якщо $r = n$, де n – число невідомих однорідної системи $Ax = 0$, то існує тільки тривіальний розв'язок;

- якщо $(r < n)$, то існує $k = n - r$ лінійно незалежних розв'язків даної системи $(x)_1, (x)_2, \dots, (x)_{n-r}$, а загальний розв'язок має вигляд:

$$x_{OO} = c_1(x)_1 + c_2(x)_2 + \dots + c_{n-r}(x)_{n-r}, \text{ де } c_1, c_2, \dots, c_{n-r} \text{ – константи.}$$

3.11.11. Фундаментальна система розв'язків системи однорідних лінійних алгебраїчних рівнянь

Нехай дана деяка однорідна система $A_{m \times n}x = 0$ й існує деякий набір векторів лінійно незалежних розв'язків даної системи $(x)_1, \dots, (x)_i, \dots, (x)_k$ таких, що $(x)_i = (x_1, x_2, \dots, x_n)_i$. Тоді справедлива теорема про фундаментальну систему розв'язків [51].

Теорема 3.15. *Про фундаментальну систему розв'язків (ФСР)*

Якщо ранг матриці $r = \text{rang } A$ менший за кількість невідомих $r < n$, тоді: існує ФСР $(x)_1, \dots, (x)_i, \dots, (x)_k$, ФСР складається з $k = n - r$ векторів, загальний розв'язок СЛАР презентовано лінійною комбінацією:
 $x_{OO} = c_1(x)_1 + c_2(x)_2 + \dots + c_{n-r}(x)_{n-r}$

Приклад 3.38. Розв'язати систему:

$$\begin{cases} x_1 + x_2 + 2x_3 + x_4 = 0, \\ 3x_1 + 2x_2 + x_3 + 3x_4 = 0, \\ 2x_1 + \frac{3}{2}x_2 + \frac{3}{2}x_3 + 2x_4 = 0. \end{cases} \Rightarrow \begin{pmatrix} 1 & 1 & 2 & 1 \\ 3 & 2 & 1 & 3 \\ 2 & \frac{3}{2} & \frac{3}{2} & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Домножимо другий рядок на 2, а третій – на 3 і від третього рядка віднімемо другий.

$$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 3 & 2 & 1 & 3 \\ 2 & \frac{3}{2} & \frac{3}{2} & 2 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 & 2 & 1 \\ 6 & 4 & 2 & 6 \\ 6 & \frac{9}{2} & \frac{9}{2} & 6 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 & 2 & 1 \\ 6 & 4 & 2 & 6 \\ 0 & -\frac{1}{2} & -\frac{5}{2} & 0 \end{pmatrix}$$

Домножимо перший рядок на 6, віднімемо другий рядок від першого, а потім поділимо другий рядок на 2.

$$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 6 & 4 & 2 & 6 \\ 0 & -\frac{1}{2} & -\frac{5}{2} & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 6 & 6 & 12 & 6 \\ 6 & 4 & 2 & 6 \\ 0 & -\frac{1}{2} & -\frac{5}{2} & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & -1 & -5 & 0 \\ 0 & -\frac{1}{2} & -\frac{5}{2} & 0 \end{pmatrix}$$

Домножимо другий рядок на $-1/2$ й складемо з третім

$$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & -1 & -5 & 0 \\ 0 & -\frac{1}{2} & -\frac{5}{2} & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & \frac{1}{2} & \frac{5}{2} & 0 \\ 0 & -\frac{1}{2} & -\frac{5}{2} & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & -1 & -5 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Таким чином, ранг системи (ранг її основної матриці) дорівнює двом. Це означає, що існує $n - r = 4 - 2 = 2$ лінійно незалежних розв'язків системи.

Таким чином, початкова система набуде вигляду:

$$\begin{cases} x_1 + x_2 + 2x_3 + x_4 = 0, \\ x_2 + 5x_3 = 0. \end{cases}$$

Виберемо x_1 та x_2 як головні змінні.

$$\text{Тоді: } \begin{cases} x_1 = 3x_3 - x_4 \\ x_2 = -5x_3 \end{cases}$$

Підставимо по черзі одиниці як одну з вільних змінних: x_3 та x_4 .

Одержимо вектори фундаментальної системи:

$$\begin{array}{ccccc} & x_1 & x_2 & x_3 & x_4 \\ (x)_1 & 3 & -5 & 1 & 0 \\ (x)_2 & -1 & 0 & 0 & 1 \end{array} \quad (x)_1 = \begin{pmatrix} 3 \\ -5 \\ 1 \\ 0 \end{pmatrix}, \quad (x)_2 = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Таким чином, загальний розв'язок розглянутої системи

$$\begin{cases} x_1 + x_2 + 2x_3 + x_4 = 0, \\ 3x_1 + 2x_2 + x_3 + 3x_4 = 0, \\ 2x_1 + \frac{3}{2}x_2 + \frac{3}{2}x_3 + 2x_4 = 0. \end{cases}$$

може бути записаний так:

$$x_{OO} = c_1 \begin{pmatrix} 3 \\ -5 \\ 1 \\ 0 \end{pmatrix} + c_2 \begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

або покоординатно: $x_1 = 3c_1 - c_2$, $x_2 = -5c_1$, $x_3 = c_1$, $x_4 = c_2$.

3.11.12. Загальний розв'язок неоднорідної системи лінійних рівнянь

У загальному випадку неоднорідної системи $Ax = b$ її загальний розв'язок визначають згідно з теоремою.

Теорема 3.16. *Про загальний розв'язок неоднорідної СЛАР*

Нехай система $Ax = b$ сумісна, тобто $\text{rang}A = \text{rang}B = r$, тоді:

- якщо $r = n$, де n – число змінних системи $Ax = b$, то розв'язок $Ax = b$ існує й він єдиний;

- якщо $r < n$, то загальний розв'язок системи $Ax = b$ має вигляд $(x)_{ZH} = (x)_{ZO} + (x)_{CH}$, де $(x)_{ZO}$ – загальний розв'язок системи $Ax = 0$, названий *загальним однорідним розв'язком*,

$(x)_{CH}$ – частинний розв'язок системи $Ax = b$, названий *частинним неоднорідним розв'язком*.

Приклад 3.39. Розв'яжемо систему.

Домножимо третє рівняння – на 3.

$$\begin{cases} x_1 + 2x_2 - 3x_3 + x_4 = 1, \\ 3x_3 + x_4 = 4, \\ x_3 + 2x_4 = 3. \end{cases} \Rightarrow \begin{cases} x_1 + 2x_2 - 3x_3 + x_4 = 1, \\ 3x_3 + x_4 = 4, \\ 3x_3 + 6x_4 = 9. \end{cases}$$

Відніmemo третє рівняння від другого й результат запишемо на місце третього рівняння, а третє – на місце другого

$$\begin{cases} x_1 + 2x_2 - 3x_3 + x_4 = 1, \\ x_3 + 2x_4 = 3, \\ 5x_4 = 5. \end{cases} \Rightarrow \begin{cases} x_1 + 2x_2 - 3x_3 + x_4 = 1, \\ x_3 + 2x_4 = 3, \\ x_4 = 1. \end{cases}$$

Визначимо частинний розв'язок даної системи рівнянь, прийнявши за головні змінні x_2, x_3, x_4 .

Прийmemo значення вільної змінної $x_1 = 1$.

Розв'язавши систему при $x_1 = 1$:

$$\begin{cases} x_1 + 2x_2 - 3x_3 + x_4 = 1, \\ x_3 + 2x_4 = 3, \\ x_4 = 1. \end{cases} \Rightarrow \begin{cases} 1 + 2x_2 - 3x_3 + x_4 = 1, \\ x_3 + 2x_4 = 3, \\ x_4 = 1. \end{cases}$$

одержимо частинний розв'язок $(x)_{\text{ЧН}} = (1, 1, 1, 1)^T$.

Однорідна система має вигляд:
$$\begin{cases} 1 + 2x_2 - 3x_3 + x_4 = 0, \\ x_3 + 2x_4 = 0, \\ x_4 = 0. \end{cases}$$

Ранг системи (ранг її основної матриці) дорівнює трьом. Це означає, що існує $n - r = 4 - 3 = 1$ лінійно незалежних розв'язків системи.

$$\begin{cases} 1 + 2x_2 - 3x_3 + x_4 = 0, \\ x_3 + 2x_4 = 0, \\ x_4 = 0. \end{cases}$$

Розв'яжемо дану систему, підставивши як вільну змінну $x_1 = 1$.

У результаті розв'язування одержимо фундаментальну систему розв'язків: $(x)_1 = \left(1, -\frac{1}{2}, 0, 0\right)$.

$$\text{Отже, } (x)_{\text{ОО}} = c_1 \begin{pmatrix} 1 \\ -\frac{1}{2} \\ 0 \\ 0 \end{pmatrix} \cdot (x)_{\text{ОН}} = (x)_{\text{ОО}} + (x)_{\text{ЧН}} = c_1 \begin{pmatrix} 1 \\ -\frac{1}{2} \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

$$x_1 = c_1 + 1, \quad x_2 = -\frac{c_1}{2}, \quad x_3 = 1, \quad x_4 = 1$$

3.11.13. Метод виключення Гауса

Метод виключення Гауса для розв'язування лінійних систем рівнянь є основою багатьох обчислювальних схем і зводиться до перетворення початкової системи до рівносильної системи з верхньотрикутною матрицею [52].

Опис методу. Розглянемо поелементну форму запису основної обчислювальної схеми методу Гауса.

Будемо розглядати роботу методу на прикладі системи чотирьох рівнянь із чотирма невідомими:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 = a_{15}, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 = a_{25}, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 = a_{35}, \\ a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 = a_{45}. \end{cases} \quad (3.53)$$

Виключаємо x_1 з системи (3.53).

1. Вибираємо як провідний (головний) елемент a_{11} , якщо $a_{11} \neq 0$.
2. Ділимо перше рівняння системи на провідний (головний) елемент:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= a_{15} \\ x_1 + b_{12}x_2 + b_{13}x_3 + b_{14}x_4 &= b_{15}, \end{aligned} \quad (3.54)$$

де $b_{1j} = \frac{a_{1j}}{a_{11}}$, ($j > 1$).

3. Виключаємо x_1 з 2-го рівняння:

Помножимо (3.54) на a_{21} : $a_{21}x_1 + a_{21}b_{12}x_2 + a_{21}b_{13}x_3 + a_{21}b_{14}x_4 = a_{21}b_{15}$

Віднімаємо його з 2-го рівняння: $a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 = a_{25}$, як показано в таблиці 3.15.

Таблиця 3.15

Віднімання від другого рівняння

$a_{21}x_1 +$	$a_{22}x_2 +$	$a_{23}x_3 +$	$a_{24}x_4$	$= a_{25}$
$a_{21}x_1 +$	$a_{21}b_{12}x_2 +$	$a_{21}b_{13}x_3 +$	$a_{21}b_{14}x_4$	$= a_{21}b_{15}$
$0 +$	$(a_{22} - a_{21}b_{12})x_2 +$	$(a_{23} - a_{21}b_{13})x_3 +$	$(a_{24} - a_{21}b_{14})x_4$	$= a_{25} - a_{21}b_{15}$

4. Виключаємо x_1 з 3-го рівняння:

Помножимо (3.54) на a_{31} : $a_{31}x_1 + a_{31}b_{12}x_2 + a_{31}b_{13}x_3 + a_{31}b_{14}x_4 = a_{31}b_{15}$

Віднімаємо його з 3-го рівняння:

$$\begin{aligned} a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 &= a_{35} \\ a_{31}x_1 + a_{31}b_{12}x_2 + a_{31}b_{13}x_3 + a_{31}b_{14}x_4 &= a_{31}b_{15} \end{aligned}$$

$$0x_1 + (a_{32} - a_{31}b_{12})x_2 + (a_{33} - a_{31}b_{13})x_3 + (a_{34} - a_{31}b_{14})x_4 = a_{35} - a_{31}b_{15}$$

5. Виключаємо x_1 з 4-го рівняння:

Помножимо (3.54) на a_{41} : $a_{41}x_1 + a_{41}b_{12}x_2 + a_{41}b_{13}x_3 + a_{41}b_{14}x_4 = a_{41}b_{15}$

Віднімаємо його з 4-го рівняння:

$$\begin{aligned} a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 &= a_{45} \\ a_{41}x_1 + a_{41}b_{12}x_2 + a_{41}b_{13}x_3 + a_{41}b_{14}x_4 &= a_{41}b_{15} \end{aligned}$$

$$0x_1 + (a_{42} - a_{41}b_{12})x_2 + (a_{43} - a_{41}b_{13})x_3 + (a_{44} - a_{41}b_{14})x_4 = a_{45} - a_{41}b_{15}$$

Об'єднаємо отримані рівняння в систему із трьох рівнянь:

$$\begin{cases} (a_{22} - a_{21}b_{12})x_2 + (a_{23} - a_{21}b_{13})x_3 + (a_{24} - a_{21}b_{14})x_4 = a_{25} - a_{21}b_{15} \\ (a_{32} - a_{31}b_{12})x_2 + (a_{33} - a_{31}b_{13})x_3 + (a_{34} - a_{31}b_{14})x_4 = a_{35} - a_{31}b_{15} \\ (a_{42} - a_{41}b_{12})x_2 + (a_{43} - a_{41}b_{13})x_3 + (a_{44} - a_{41}b_{14})x_4 = a_{45} - a_{41}b_{15} \end{cases}$$

Виконаємо заміни:

$$\begin{aligned} (a_{22} - a_{21}b_{12}) &\rightarrow a_{22}^{(1)} \quad (a_{23} - a_{21}b_{13}) \rightarrow a_{23}^{(1)} \quad (a_{24} - a_{21}b_{14}) \rightarrow a_{24}^{(1)} \quad (a_{25} - a_{21}b_{15}) \rightarrow a_{25}^{(1)} \\ (a_{32} - a_{31}b_{12}) &\rightarrow a_{32}^{(1)} \quad (a_{33} - a_{31}b_{13}) \rightarrow a_{33}^{(1)} \quad (a_{34} - a_{31}b_{14}) \rightarrow a_{34}^{(1)} \quad (a_{35} - a_{31}b_{15}) \rightarrow a_{35}^{(1)} \\ (a_{42} - a_{41}b_{12}) &\rightarrow a_{42}^{(1)} \quad (a_{43} - a_{41}b_{13}) \rightarrow a_{43}^{(1)} \quad (a_{44} - a_{41}b_{14}) \rightarrow a_{44}^{(1)} \quad (a_{45} - a_{41}b_{15}) \rightarrow a_{45}^{(1)} \end{aligned}$$

Одержимо систему:

$$\begin{cases} a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + a_{24}^{(1)}x_4 = a_{25}^{(1)}, \\ a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 + a_{34}^{(1)}x_4 = a_{35}^{(1)}, \\ a_{42}^{(1)}x_2 + a_{43}^{(1)}x_3 + a_{44}^{(1)}x_4 = a_{45}^{(1)}, \end{cases} \quad (3.55)$$

де $a_{ij}^{(1)} = a_{ij} - a_{i1}b_{1j}$ ($i, j \geq 2$).

Виключаємо x_2 з системи (3.55)

1. Ділимо перше рівняння системи (3.55) на головний елемент $a_{22}^{(1)} \neq 0$:

$$\begin{aligned} a_{22}^{(1)}x_2 + a_{23}^{(1)}x_3 + a_{24}^{(1)}x_4 &= a_{25}^{(1)} \\ x_2 + b_{23}^{(1)}x_3 + b_{24}^{(1)}x_4 &= b_{25}^{(1)} \end{aligned} \quad (3.56)$$

де $b_{2j}^{(1)} = \frac{a_{2j}^{(1)}}{a_{22}^{(1)}} \quad (j > 2)$.

2. Виключаємо x_2 з 2-го рівняння:

Помножимо (3.56) на $a_{32}^{(1)}$: $a_{32}^{(1)}x_2 + a_{32}^{(1)}b_{23}^{(1)}x_3 + a_{32}^{(1)}b_{24}^{(1)}x_4 = a_{32}^{(1)}b_{25}^{(1)}$

Віднімаємо його з 2-го рівняння:

$$\begin{aligned} a_{32}^{(1)}x_2 + a_{33}^{(1)}x_3 + a_{34}^{(1)}x_4 &= a_{35}^{(1)} \\ a_{32}^{(1)}x_2 + a_{32}^{(1)}b_{23}^{(1)}x_3 + a_{32}^{(1)}b_{24}^{(1)}x_4 &= a_{32}^{(1)}b_{25}^{(1)} \\ 0x_2 + \left(a_{33}^{(1)} - a_{32}^{(1)}b_{23}^{(1)}\right)x_3 + \left(a_{34}^{(1)} - a_{32}^{(1)}b_{24}^{(1)}\right)x_4 &= a_{35}^{(1)} - a_{32}^{(1)}b_{25}^{(1)} \end{aligned}$$

3. Виключаємо x_2 з 3-го рівняння:

Помножимо (3.56) на $a_{42}^{(1)}$: $a_{42}^{(1)}x_2 + a_{42}^{(1)}b_{23}^{(1)}x_3 + a_{42}^{(1)}b_{24}^{(1)}x_4 = a_{42}^{(1)}b_{25}^{(1)}$

Віднімаємо його з 3-го рівняння:

$$\begin{aligned} a_{42}^{(1)}x_2 + a_{43}^{(1)}x_3 + a_{44}^{(1)}x_4 &= a_{45}^{(1)} \\ a_{42}^{(1)}x_2 + a_{42}^{(1)}b_{23}^{(1)}x_3 + a_{42}^{(1)}b_{24}^{(1)}x_4 &= a_{42}^{(1)}b_{25}^{(1)} \\ 0x_2 + \left(a_{43}^{(1)} - a_{42}^{(1)}b_{23}^{(1)}\right)x_3 + \left(a_{44}^{(1)} - a_{42}^{(1)}b_{24}^{(1)}\right)x_4 &= a_{45}^{(1)} - a_{42}^{(1)}b_{25}^{(1)} \end{aligned}$$

Об'єднаємо отримані рівняння в систему із двох рівнянь:

$$\begin{cases} 0x_2 + \left(a_{33}^{(1)} - a_{32}^{(1)}b_{23}^{(1)}\right)x_3 + \left(a_{34}^{(1)} - a_{32}^{(1)}b_{24}^{(1)}\right)x_4 = a_{35}^{(1)} - a_{32}^{(1)}b_{25}^{(1)} \\ 0x_2 + \left(a_{43}^{(1)} - a_{42}^{(1)}b_{23}^{(1)}\right)x_3 + \left(a_{44}^{(1)} - a_{42}^{(1)}b_{24}^{(1)}\right)x_4 = a_{45}^{(1)} - a_{42}^{(1)}b_{25}^{(1)} \end{cases}$$

Виконаємо заміни:

$$\begin{aligned} \left(a_{33}^{(1)} - a_{32}^{(1)}b_{23}^{(1)}\right) &\rightarrow a_{33}^{(2)} & \left(a_{34}^{(1)} - a_{32}^{(1)}b_{24}^{(1)}\right) &\rightarrow a_{34}^{(2)} & \left(a_{35}^{(1)} - a_{32}^{(1)}b_{25}^{(1)}\right) &\rightarrow a_{35}^{(2)} \\ \left(a_{43}^{(1)} - a_{42}^{(1)}b_{23}^{(1)}\right) &\rightarrow a_{43}^{(2)} & \left(a_{44}^{(1)} - a_{42}^{(1)}b_{24}^{(1)}\right) &\rightarrow a_{44}^{(2)} & \left(a_{45}^{(1)} - a_{42}^{(1)}b_{25}^{(1)}\right) &\rightarrow a_{45}^{(2)} \end{aligned}$$

У результаті одержимо систему рівнянь:

$$\begin{cases} a_{33}^{(2)}x_3 + a_{34}^{(2)}x_4 = a_{35}^{(2)}, \\ a_{43}^{(2)}x_3 + a_{44}^{(2)}x_4 = a_{45}^{(2)}, \end{cases} \quad (3.57)$$

де $a_{ij}^{(2)} = a_{ij}^{(1)} - a_{i2}^{(1)}b_{2j}^{(1)}$ ($i, j \geq 3$)

Виключаємо x_3 з системи (3.57)

1. Ділимо перше рівняння системи (3.57) на головний елемент $a_{33}^{(2)} \neq 0$:

$$\begin{aligned} a_{33}^{(2)}x_3 + a_{34}^{(2)}x_4 &= a_{35}^{(2)} \\ x_3 + b_{34}^{(2)}x_4 &= b_{35}^{(2)} \end{aligned} \quad (3.58)$$

де $b_{3j}^{(2)} = \frac{a_{3j}^{(2)}}{a_{33}^{(2)}} (j > 3)$.

2. Виключаємо x_3 з 2-го рівняння:

Помножимо (3.58) на $a_{43}^{(2)}$: $a_{43}^{(2)}x_3 + a_{43}^{(2)}b_{34}^{(2)}x_4 = a_{43}^{(2)}b_{35}^{(2)}$

Віднімаємо його з 2-го рівняння:

$$\begin{aligned} a_{43}^{(2)}x_3 + a_{44}^{(2)}x_4 &= a_{45}^{(2)} \\ a_{43}^{(2)}x_3 + a_{43}^{(2)}b_{34}^{(2)}x_4 &= a_{43}^{(2)}b_{35}^{(2)} \\ 0x_3 + \left(a_{44}^{(2)} - a_{43}^{(2)}b_{34}^{(2)}\right)x_4 &= a_{45}^{(2)} - a_{43}^{(2)}b_{35}^{(2)} \end{aligned}$$

Виконаємо заміну

$$\left(a_{44}^{(2)} - a_{43}^{(2)}b_{34}^{(2)}\right) \rightarrow a_{44}^{(3)} \quad \left(a_{45}^{(2)} - a_{43}^{(2)}b_{35}^{(2)}\right) \rightarrow a_{45}^{(3)}$$

Одержимо рівняння

$$a_{44}^{(3)}x_4 = a_{45}^{(3)} \quad (3.59)$$

де $a_{ij}^{(3)} = a_{ij}^{(2)} - a_{i3}^{(2)}b_{3j}^{(2)}$ ($i, j \geq 4$).

Обчислимо x_4 рівняння (3.59).

$$x_4 = \frac{a_{45}^{(3)}}{a_{44}^{(3)}} = b_{45}^{(3)}.$$

Інші невідомі послідовно будемо обчислювати з рівнянь (3.58), (3.56), (3.54):

$$(3.59) \quad x_4 = b_{45}^{(3)}$$

$$(3.58) \quad x_3 = b_{35}^{(2)} - b_{34}^{(2)} x_4,$$

$$(3.56) \quad x_2 = b_{25}^{(1)} - b_{24}^{(1)} x_4 - b_{23}^{(1)} x_3,$$

$$(3.54) \quad x_1 = b_{15} - b_{14} x_4 - b_{13} x_3 - b_{12} x_2$$

Висновок.

За методом Гауса побудована еквівалентна система:

$$x_1 + b_{12} x_2 + b_{13} x_3 + b_{14} x_4 = b_{15}$$

$$x_2 + b_{23}^{(1)} x_3 + b_{24}^{(1)} x_4 = b_{25}^{(1)}$$

$$x_3 + b_{34}^{(2)} x_4 = b_{35}^{(2)}$$

$$x_4 = b_{45}^{(3)}$$

Система має трикутну матрицю. Необхідною й достатньою умовою застосовності методу є *не рівність нулю* всіх «головних елементів».

Узагальнення методу Гауса для довільної системи

Процес виключення невідомого x_k із усіх рівнянь $j > k$ називають прямим ходом методу Гауса. Формули для обчислення коефіцієнтів системи на k -му кроці мають вигляд [53]:

$$b_{1j}^{(0)} = a_{1j}^{(0)} / a_{11}^{(0)}, \quad (j > 1)$$

$$b_{2j}^{(1)} = a_{2j}^{(1)} / a_{22}^{(1)}, \quad (j > 2) \Rightarrow b_{kj}^{(k-1)} = \frac{a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}$$

$$b_{3j}^{(2)} = a_{3j}^{(2)} / a_{33}^{(2)} \quad (j > 3)$$

$$j = k, \dots, n + 1.$$

$$\begin{aligned}
a_{ij}^{(1)} &= a_{ij}^{(0)} - a_{j1}^{(0)} b_{1j}^{(0)} \quad (i, j \geq 2) \\
a_{ij}^{(2)} &= a_{ij}^{(1)} - a_{i2}^{(1)} b_{2j}^{(1)} \quad (i, j \geq 3) \Rightarrow a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{ik}^{(k-1)} b_{kj}^{(k-1)} \\
a_{ij}^{(3)} &= a_{ij}^{(2)} - a_{i3}^{(2)} b_{3j}^{(2)} \quad (i, j \geq 4) \\
& i = k + 1, \dots, n; \quad j = k, \dots, n + 1.
\end{aligned}$$

(i – номер рядка, j – номер стовпця елемента початкової матриці).

Визначення невідомих за формулами

$$x_i = b_{i,n+1}^{(i-1)} - \sum_{j=i+1}^n b_{ij}^{(i-1)} x_j, \quad i = n, n-1, \dots, 1$$

називають зворотним ходом методу Гауса.

3.11.14. Метод Гауса-Жордана

Суть методу – приведення матриці початкової системи до *діагонального вигляду* шляхом перетворення коефіцієнтів рівнянь, розташованих вище й нижче головного рівняння [54].

Застосування описаного методу ускладнюється, якщо в будь-якому рівнянні «головний елемент» дорівнює нулю. Однак ці труднощі можна обійти, змінивши порядок розташування рівнянь системи.

Найбільша точність досягається тоді, коли «головний елемент» має найбільше значення. Тому рядок з нульовим або малим головним елементом потрібно замінити на той з нижніх рядків, у якому у тому ж стовпці розміщений елемент, який має найбільше значення.

Опис методу. Розглянемо запис основної обчислювальної схеми методу Гауса-Жордана.

Будемо розглядати роботу методу на прикладі конкретної системи чотирьох рівнянь із чотирма невідомими:

$$\begin{aligned}
A_1 & \left\{ \begin{array}{l} x_1 + x_2 + x_3 - x_4 = 2 \\ x_1 - x_2 - x_3 + x_4 = 0 \\ 2x_1 + x_2 - x_3 + 2x_4 = 9 \\ 3x_1 + x_2 + 2x_3 - x_4 = 7 \end{array} \right.
\end{aligned}$$

Розв'язок

1. Оскільки четверте рівняння містить максимальний коефіцієнт при x_1 , то міняємо місцями перший рядок із четвертим.

$$\begin{array}{l} A_1 \\ A_2 \\ A_3 \\ A_4 \end{array} \left\{ \begin{array}{l} 3x_1 + x_2 + 2x_3 - x_4 = 7 \\ x_1 - x_2 - x_3 + x_4 = 0 \\ 2x_1 + x_2 - x_3 + 2x_4 = 9 \\ x_1 + x_2 + x_3 - x_4 = 2 \end{array} \right.$$

2. Ділимо рівняння першого рядка на «головний елемент» 3.

$$B_1: x_1 + \frac{1}{3}x_2 + \frac{2}{3}x_3 - \frac{1}{3}x_4 = \frac{7}{3}.$$

$$x_1 + 0.333x_2 + 0.666x_3 - 0.333x_4 = 2.333$$

3. Виключаємо змінну x_1 способом, аналогічним методу Гауса.

$$\begin{array}{l} B_1 \\ B_2 = A_2 - B_1 \\ B_3 = A_3 - 2B_1 \\ B_4 = A_4 - B_1 \end{array} \left\{ \begin{array}{l} x_1 + 0.333x_2 + 0.666x_3 - 0.333x_4 = 2.333 \\ 0x_1 - 1.333x_2 - 1.666x_3 + 1.333x_4 = -2.333 \\ 0x_1 + 0.333x_2 - 2.333x_3 + 2.666x_4 = 4.333 \\ 0x_1 + 0.666x_2 + 0.333x_3 - 0.666x_4 = -0.333 \end{array} \right.$$

4. Ділимо рівняння другого рядка на «головний елемент» -1.333 .

$$C_2: x_2 + 1.25x_3 - 1x_4 = 1.75$$

5. Виключаємо x_2 .

$$\begin{array}{l} C_1 = B_1 - C_2(0.333) \\ C_3 = B_3 - C_2(0.333) \\ C_4 = B_4 - C_2(0.666) \end{array} \left\{ \begin{array}{l} C_1 \left\{ \begin{array}{l} 1x_1 + 0x_2 + 0.25x_3 - 0x_4 = 1.75 \\ \mathbf{0x_1 + 1x_2 + 1.25x_3 - 1x_4 = 1.75} \\ 0x_1 + 0x_2 - 2.75x_3 + 3x_4 = 3.75 \\ 0x_1 + 0x_2 - 0.5x_3 + 0x_4 = -1.5 \end{array} \right. \end{array} \right.$$

6. Ділимо рівняння третього рядка на «головний елемент» -2.75 .

$$D: x_3 - 1.09x_4 = -1.363$$

7. Виключаємо x_3 . Головний елемент при x_3 міститься в третьому рівнянні.

$$\begin{array}{l}
 D_1 = C_1 - D_3(0.25) \\
 D_2 = C_2 - D_3(1.25) \\
 D_4 = C_4 - D_3(0.5)
 \end{array}
 \begin{array}{l}
 D_1 \\
 D_2 \\
 D_3 \\
 D_4
 \end{array}
 \begin{cases}
 1x_1 & +0x_2 & +0x_3 & +0.272x_4 & = & 2.0909 \\
 0x_1 & +1x_2 & +0x_3 & +0.363x_4 & = & 3.454 \\
 0x_1 & +0x_2 & +1x_3 & -1.09x_4 & = & -1.363 \\
 0x_1 & +0x_2 & +0x_3 & -0.545x_4 & = & -2.181
 \end{cases}$$

8. Ділимо рівняння четвертого рядка на «головний елемент» -0.545 .

$$F : 1x_4 = 4$$

9. Виключаємо x_4 із усіх рівнянь, крім останнього

$$\begin{array}{l}
 F_1 = D_1 - (0.272)F_4 \\
 F_2 = D_2 - (0.363)F_4 \\
 F_3 = D_3 - (-1.09)F_4
 \end{array}
 \begin{cases}
 1x_1 & +0x_2 & +0x_3 & +0x_4 & = & 1 \\
 0x_1 & +1x_2 & +0x_3 & +0x_4 & = & 2 \\
 0x_1 & +0x_2 & +1x_3 & -0x_4 & = & 3 \\
 0x_1 & +0x_2 & +0x_3 & -1x_4 & = & 4
 \end{cases}$$

$$x_1 = 1, \quad x_2 = 2, \quad x_3 = 3, \quad x_4 = 4$$

Відмінність методу Гауса-Жордана від методу Гауса

Перетворена матриця має діагональний вигляд і тому *немає необхідності* у зворотному ході.

3.11.15. Метод квадратного кореня

Метод застосовується тільки для спеціальних видів СЛАР [55].

Умови застосування методу:

1. Матриця системи A повинна бути невиродженою, тобто $(\det A \neq 0)$
2. Матриця системи A повинна бути симетричною, тобто $A = A^T$.
3. Для того, щоб не виконувати обчислень із комплексними числами, повинна виконуватися додаткова умова: матриця A повинна бути додатно визначена, тобто всі її головні мінори повинні бути додатними.

Матричний опис методу квадратного кореня

Теорема 3.17. Нехай дана система лінійних алгебраїчних рівнянь $Ax = b$ задовольняє умові застосовності методу квадратного кореня.

Тоді існує така верхньотрикутна матриця S , що:

$$S^T S = A.$$

У цьому випадку початкову систему можна записати у вигляді:

$$(S^T S)x = b \text{ або } S^T (Sx) = b.$$

Якщо позначити $Sx = y$, то весь процес знаходження розв'язку x можна розбити на три етапи:

1. Знайти матрицю S , що задовольняє виразу: $S^T S = A$;
2. Знайти вектор y , що відповідає умові: $S^T y = b$;
3. Знайти вектор x з умови: $Sx = y$.

Етап 1. Знаходження матриці S («квадратного кореня» з A)

Покажемо процес знаходження коефіцієнтів матриці S для матриці A розмірності 4×4 , а потім запишемо загальні формули.

Позначимо елементи матриці S :

$$S = \begin{pmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ 0 & s_{21} & s_{22} & s_{23} \\ 0 & 0 & s_{32} & s_{33} \\ 0 & 0 & 0 & s_{34} \end{pmatrix}.$$

Тоді повинно бути виконаним співвідношення $A = S^T S$, або

$$\begin{pmatrix} s_{11} & 0 & 0 & 0 \\ s_{12} & s_{22} & 0 & 0 \\ s_{13} & s_{23} & s_{33} & 0 \\ s_{14} & s_{24} & s_{34} & s_{44} \end{pmatrix} \begin{pmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ 0 & s_{22} & s_{23} & s_{24} \\ 0 & 0 & s_{33} & s_{34} \\ 0 & 0 & 0 & s_{44} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

За правилами множення матриць одержуємо систему:

$$\left\{ \begin{array}{l} s_{11} \cdot s_{11} = a_{11}, \\ s_{11} \cdot s_{12} = a_{12}, \\ s_{11} \cdot s_{13} = a_{13}, \\ s_{11} \cdot s_{14} = a_{14}, \\ s_{12} \cdot s_{12} + s_{22} \cdot s_{22} = a_{22}, \\ s_{12} \cdot s_{13} + s_{22} \cdot s_{23} = a_{23}, \\ s_{12} \cdot s_{14} + s_{22} \cdot s_{24} = a_{24}, \\ s_{13} \cdot s_{13} + s_{23} \cdot s_{23} + s_{33} \cdot s_{33} = a_{33}, \\ s_{13} \cdot s_{14} + s_{23} \cdot s_{24} + s_{33} \cdot s_{34} = a_{34}, \\ s_{14} \cdot s_{14} + s_{24} \cdot s_{24} + s_{34} \cdot s_{34} + s_{44} \cdot s_{44} = a_{44}. \end{array} \right.$$

$$s_{11} = \sqrt{a_{11}}, \quad s_{12} = \frac{a_{12}}{s_{11}}, \quad s_{13} = \frac{a_{13}}{s_{11}}, \quad s_{14} = \frac{a_{14}}{s_{11}}.$$

$$s_{22} = \sqrt{a_{22} - s_{12}^2}, s_{23} = \frac{a_{23} - s_{12}s_{13}}{s_{22}}, s_{24} = \frac{a_{24} - s_{12}s_{14}}{s_{22}},$$

$$s_{33} = \sqrt{a_{33} - s_{13}^2 - s_{23}^2}, s_{34} = \frac{a_{34} - s_{13}s_{14} - s_{23}s_{24}}{s_{33}}, s_{44} = \sqrt{a_{44} - s_{14}^2 - s_{24}^2 - s_{34}^2}.$$

Загальні формули для знаходження елементів матриці S мають

вигляд: $s_{11} = \sqrt{a_{11}}, s_{1j} = \frac{a_{1j}}{s_{11}} (j > 1),$

$$s_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} s_{ki}^2} (1 < i \leq n), s_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} t_{ki}t_{kj}}{t_{ii}} (i < j), s_{ij} = 0 (i > j)$$

Етан 2. Знаходження вектора y , що відповідає умові: $S^T y = b$.

Запишемо співвідношення $S^T y = b$ у поелементному вигляді:

$$\begin{pmatrix} s_{11} & 0 & 0 & 0 \\ s_{12} & s_{22} & 0 & 0 \\ s_{13} & s_{23} & s_{33} & 0 \\ s_{14} & s_{24} & s_{34} & s_{44} \end{pmatrix} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}.$$

Виконавши операцію множення матриці S^T на вектор y , одержимо:

$$s_{11}y_1 = b_1,$$

$$s_{12}y_1 + s_{22}y_2 = b_2,$$

$$s_{13}y_1 + s_{23}y_2 + s_{33}y_3 = b_3,$$

$$s_{14}y_1 + s_{24}y_2 + s_{34}y_3 + s_{44}y_4 = b_4.$$

Звідси послідовно знаходимо:

$$y_1 = \frac{b_1}{s_{11}}, y_2 = \frac{b_2 - s_{12}y_1}{s_{22}}, y_3 = \frac{b_3 - s_{13}y_1 - s_{23}y_2}{s_{33}},$$

$$y_4 = \frac{b_4 - s_{14}y_1 - s_{24}y_2 - s_{34}y_3}{s_{44}}.$$

Загальні формули для знаходження y мають вигляд:

$$y_1 = \frac{b_1}{s_{11}}, \quad y_i = \frac{b_i - \sum_{k=1}^{i-1} s_{ki} y_k}{s_{ii}} \quad (i > j)$$

Етап 3. Знаходження вектора x з умови: $Sx = y$.

Запишемо співвідношення $Sx = y$ в поелементному вигляді:

$$\begin{pmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ 0 & s_{22} & s_{23} & s_{24} \\ 0 & 0 & s_{33} & s_{34} \\ 0 & 0 & 0 & s_{44} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}.$$

Після виконання множення матриці на вектор:

$$s_{11}x_1 + s_{12}x_2 + s_{13}x_3 + s_{14}x_4 = y_1,$$

$$s_{22}x_2 + s_{23}x_3 + s_{24}x_4 = y_2,$$

$$s_{33}x_3 + s_{34}x_4 = y_3,$$

$$s_{44}x_4 = y_4.$$

$$x_4 = \frac{y_4}{s_{44}}, \quad x_3 = \frac{y_3 - s_{33}x_3}{s_{33}}, \quad x_2 = \frac{y_2 - s_{23}x_3 - s_{24}x_4}{s_{22}},$$

$$x_1 = \frac{y_1 - s_{12}x_2 - s_{13}x_3 - s_{14}x_4}{s_{11}},$$

Загальні формули мають вигляд: $x_i = \frac{y_i - \sum_{k=i+1}^n s_{ik}x_k}{s_{ii}} \quad (i < n), \quad x_n = \frac{y_n}{s_{nn}}.$

3.11.16. Метод прогонки

При моделюванні деяких задач, а також при чисельному розв'язуванні крайових задач для диференціальних рівнянь отримуємо системи з розрідженою матрицею – матриця A містить велику кількість нульових елементів [56].

Зокрема, становлять інтерес системи з трьохдіагональною матрицею.

У загальному випадку рекурентні співвідношення для будь-якого номера $i = 2, 3, \dots, n - 1$.

$$A_i = -\frac{c_i}{l_i}, B_i = \frac{d_i - a_i B_{i-1}}{l_i}, l_i = a_i A_{i-1} + b_i$$

Етап 2. Зворотна прогонка

Останнє рівняння системи має вигляд: $a_n x_{n-1} + b_n x_n = d_n$.

Виразимо з нього значення x_{n-1} : $x_{n-1} = \frac{d_n - b_n x_n}{a_n}$.

У той же час із $x_i = A_i x_{i+1} + B_i$, випливає, що при $i = n - 1$

$$x_{n-1} = A_{n-1} x_n + B_{n-1}.$$

Прирівняємо два отриманих вирази:

$$\frac{d_n - b_n x_n}{a_n} = A_{n-1} x_n + B_{n-1},$$

$$d_n - b_n x_n = a_n A_{n-1} x_n + a_n B_{n-1},$$

$$d_n - a_n B_{n-1} = a_n A_{n-1} x_n + b_n x_n.$$

З рівняння одержимо значення x_n : $x_n = \frac{d_n - a_n B_{n-1}}{b_n + a_n A_{n-1}}$.

Далі будемо послідовно обчислювати x_i зі зменшенням індексу:

$$x_{n-1} = A_{n-1} x_n + B_{n-1}$$

$$x_{n-2} = A_{n-2} x_{n-1} + B_{n-2}$$

.....

$$x_i = A_i x_{i+1} + B_i$$

$$x_1 = A_1 x_2 + B_1$$

$$A_i = -\frac{c_i}{l_i}, B_i = \frac{d_i - a_i B_{i-1}}{l_i}, l_i = a_i A_{i-1} + b_i$$

Контрольні запитання

1. Знайти загальний розв'язок неоднорідної системи лінійних рівнянь

$$\begin{cases} x_1 + 4x_2 - 3x_3 - 5x_4 = 1, \\ 3x_3 + 2x_4 = 5, \\ x_3 + 2x_4 = 3. \end{cases}$$

2. Розв'язати систему методом Гауса-Жордана.

$$\begin{cases} 2x_1 + x_2 - x_3 = 2 \\ 3x_1 + x_2 - 2x_3 = 3 \\ x_1 + x_3 = 3 \end{cases}$$

3. Розв'язати систему методом Гауса

$$\begin{cases} x_1 - x_3 = 2, \\ 4x_2 + 4x_3 = 12, \\ -x_1 + 4x_2 + 14x_3 = 19 \end{cases}$$

5. Визначити, при яких значеннях параметра λ існує матриця, зворотна

до матриці $A = \begin{pmatrix} 1 & -2 & 2 \\ \lambda & 3 & 0 \\ 2 & 2 & 1 \end{pmatrix}$

3.12. Ітераційні методи розв'язування систем алгебраїчних рівнянь

3.12.1. Точні й ітераційні методи

Методи розв'язування систем лінійних алгебраїчних рівнянь поділяються на дві групи:

1. Точні методи.
2. Ітераційні методи.

Точні методи розв'язування – це методи, що дозволяють одержати точне значення всіх невідомих у результаті скінченного числа арифметичних операцій.

Ітераційні методи – це методи, що дозволяють одержати розв'язок у вигляді границі послідовності векторів, побудову яких проводять однаковим процесом, названим процесом ітерації [57].

При великій кількості невідомих лінійної системи схема методу Гауса, що дає *точний* розв'язок, стає досить складною.

За цих умов для знаходження кореня системи використовують *наближені* чисельні методи.

3.12.2. Загальна схема побудови ітераційних методів

Нехай потрібно розв'язати систему $Ax = b$; $b, x \in R^m$, $A \in R^{m \times m}$.

Уведемо матрицю $Q \in R^{m \times m}$ як матрицю, що апроксимує матрицю A . Матриця Q повинна мати вигляд, що дозволяє спростити розв'язування наближеної системи: $Ax = b; \Rightarrow Qu = d$.

На практиці така матриця найчастіше є нижньотрикутною, верхньотрикутною, діагональною або добутком скінченного числа таких простих матриць.

Отже, розв'язування системи $Qu = d$, де Q – матриця розщеплення, не повинно представляти особливих труднощів, оскільки розв'язок може бути отриманий з використанням алгоритму, подібного до методу Гауса.

3.12.3. Побудова ітераційного процесу

Перепишемо систему $Ax = b$ у вигляді: $(Q - Q + A)x = b$.

Використовуючи еквівалентні перетворення, запишемо:

$$Qx = (Q - A)x + b$$

Введемо верхній індекс, як номер ітерації

$$Qx^{(n+1)} = (Q - A)x^{(n)} + b; \quad n = 0, 1, 2, \dots$$

Поділивши обидві частини виразу на Q , одержимо:

$$x^{(n+1)} = Bx^{(n)} + d; \quad n = 0, 1, 2, \dots, \quad (3.60)$$

де $B = \frac{Q - A}{Q} = I - Q^{-1}A$; $d = Q^{-1}b$, I – одинична матриця.

Це лінійна схема першого порядку. Оскільки ітераційна матриця B на ітераціях є постійною, то таку схему називають стаціонарною [58].

3.12.4. Теорема про достатню умову збіжності ітераційного методу

Метод простих ітерацій

$$x^{(n+1)} = Bx^{(n)} + d; \quad n = 0, 1, 2, \dots$$

сходиться до єдиного розв'язку початкової системи

$$Ax = b;$$

при будь-якому початковому наближенні $x^{(0)}$ зі швидкістю, не повільнішою за геометричну прогресію, якщо будь-яка норма матриці B є меншою за одиницю, тобто

$$\|B\| < 1, \text{ де}$$

$$\|B\|_l = \max_{1 \leq j \leq n} \sum_{i=1}^n |b_{ij}| \text{ – } l\text{-норма, } \|B\|_m = \max_{1 \leq i \leq n} \sum_{j=1}^n |b_{ij}| \text{ – } m\text{-норма,}$$

$$\|B\|_2 = \sqrt{\sum_{i,j} |b_{ij}|^2} \text{ – норма Фробеніуса}$$

3.12.5. Метод простої ітерації (метод Якобі)

Обчислювальну схему визначають способом вибору матриці розщеплення Q . Представимо

$$A = A^- + D + A^+,$$

де D – діагональна матриця з діагональними членами матриці A ;

A^- – нижньотрикутна матриця, що містить частину матриці A , що лежить нижче центральної діагоналі;

A^+ – верхньотрикутна матриця, що містить частину матриці A , що лежить вище центральної діагоналі.

$$A = A^- + D + A^+,$$

$$\text{де } A^- = \begin{pmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & 0 \end{pmatrix}, A^+ = \begin{pmatrix} 0 & a_{12} & \dots & a_{1m} \\ 0 & 0 & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{pmatrix},$$

$$D = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{mm} \end{pmatrix}. \text{ Виберемо як матрицю розщеплення } Q = D.$$

Підставимо в загальну ітераційну формулу

$$x^{(n+1)} = Bx^{(n)} + d; \quad n = 0, 1, 2, \dots$$

значення B й d з виразів $B = I - Q^{-1}A$; $d = Q^{-1}b$

Тоді одержимо:

$$x^{(n+1)} = (I - Q^{-1}A)x^{(n)} + Q^{-1}b; \quad n = 0, 1, 2, \dots$$

Якщо $Q = D$ то $Q^{-1} = D^{-1}$. Звідси

$$x^{(n+1)} = (I - D^{-1}A)x^{(n)} + D^{-1}b; \quad n = 0, 1, 2, \dots$$

Підставимо замість A її представлення у вигляді суми матриць:

$$A = A^- + D + A^+$$

Тоді одержимо

$$x^{(n+1)} = (I - D^{-1}(A^+ + D + A^-))x^{(n)} + D^{-1}b; \quad n = 0, 1, 2, \dots$$

У виразі

$$x^{(n+1)} = (I - D^{-1}(A^+ + D + A^-))x^{(n)} + D^{-1}b; \quad n = 0, 1, 2, \dots$$

виконаємо елементарні перетворення:

$$x^{(n+1)} = (\chi - D^{-1}A^+ - \chi - D^{-1}A^-)x^{(n)} + D^{-1}b; \quad n = 0, 1, 2, \dots$$

$$x^{(n+1)} = -D^{-1}A^-x^{(n)} - D^{-1}A^+x^{(n)} + D^{-1}b; \quad n = 0, 1, 2, \dots \quad (3.61)$$

або в неявній формі

$$Dx^{(n+1)} + A^-x^{(n)} + A^+x^{(n)} = b.$$

Додаємо й віднімаємо в лівій частині виразу $Dx^{(n)}$:

$$Dx^{(n+1)} + Dx^{(n)} - Dx^{(n)} + A^-x^{(n)} + A^+x^{(n)} = b.$$

Згрупуємо доданки:

$$D\left(x^{(n+1)} - x^{(n)}\right) + (A^+ + A^- + D)x^{(n)} = b$$

У підсумку одержуємо $D\left(x^{(n+1)} - x^{(n)}\right) + Ax^{(n)} = b$.

Вираз $D\left(x^{(n+1)} - x^{(n)}\right) + Ax^{(n)} = b$ $n = 0, 1, 2, \dots$ називають *неявним записом методу простої ітерації*.

Для одержання методу Якобі розглянемо отриманий раніше вираз:

$$x^{(n+1)} = -D^{-1}A^-x^{(n)} - D^{-1}A^+x^{(n)} + D^{-1}b; n = 0, 1, 2, \dots$$

Винесемо за дужки $-D^{-1}$.

$$x^{(n+1)} = -D^{-1}\left(A^- + A^+\right)x^{(n)} + D^{-1}b; n = 0, 1, 2, \dots$$

Для перетворення в неявну форму домножимо на D

$$Dx^{(n+1)} + \left(A^- + A^+\right)x^{(n)} = b, n = 0, 1, 2, \dots,$$

Вираз $Dx^{(n+1)} + \left(A^- + A^+\right)x^{(n)} = b; n = 0, 1, 2, \dots$ називають *неявним записом методу Якобі*.

Зведення методу Якобі до методу простої ітерації

Додаємо й віднімаємо $Dx^{(n)}$ до правої частини виразу

$$Dx^{(n+1)} + \left(A^- + A^+\right)x^{(n)} = b; n = 0, 1, 2, \dots$$

$$Dx^{(n+1)} = -\left(A^- + A^+\right)x^{(n)} + Dx^{(n)} - Dx^{(n)} + b,$$

$$Dx^{(n+1)} = -A^-x^{(n)} - A^+x^{(n)} + Dx^{(n)} - Dx^{(n)} + b,$$

$$Dx^{(n+1)} = -Ax^{(n)} + Dx^{(n)} + b$$

$$D\left(x^{(n+1)} - x^{(n)}\right) + Ax^{(n)} = b.$$

Таким чином, метод Якобі можна звести до того ж неявного виразу, що й метод простої ітерації.

3.12.6. Метод Гауса-Зейделя

Виберемо як матрицю розщеплення $Q = A^- + D$.

Підставимо в загальну ітераційну формулу

$$x^{(n+1)} = Bx^{(n)} + d; \quad n = 0, 1, 2, \dots$$

значення B й d з виразів $B = I - Q^{-1}A$; $d = Q^{-1}b$.

Тоді одержимо: $x^{(n+1)} = (I - Q^{-1}A)x^{(n)} + Q^{-1}b$; $n = 0, 1, 2, \dots$

Розкриємо дужки й домножимо на Q ліву й праву частину:

$$Qx^{(n+1)} = Qx^{(n)} - Ax^{(n)} + b; \quad n = 0, 1, 2, \dots$$

Введемо заміну $Q = A^- + D$:

$$\begin{aligned} A^-x^{(n+1)} + Dx^{(n+1)} &= \\ &= \cancel{A^-x^{(n)}} + \cancel{Dx^{(n)}} - A^+x^{(n)} - \cancel{Dx^{(n)}} - \cancel{A^-x^{(n)}} + b; \quad n = 0, 1, 2, \dots \end{aligned}$$

Зведемо подібні члени:

$$Dx^{(n+1)} = -A^+x^{(n)} - A^-x^{(n+1)} + b$$

Розділивши обидві частини рівняння на D , одержимо метод Гауса-Зейделя в явній формі:

$$x^{(n+1)} = -D^{-1}A^-x^{(n+1)} - D^{-1}A^+x^{(n)} + D^{-1}b; \quad n = 0, 1, \dots$$

Перетворимо явний ітераційний вираз для методу Гауса-Зейделя в неявну форму: $Dx^{(n+1)} = -A^-x^{(n+1)} - A^+x^{(n)} + b$,

Додамо й віднімемо у правій частині $(Dx^{(n)} + A^-x^{(n)})$:

$$Dx^{(n+1)} + A^-x^{(n+1)} = Dx^{(n)} + A^-x^{(n)} - Dx^{(n)} - A^-x^{(n)} - A^+x^{(n)} + b$$

$$(D + A^-)x^{(n+1)} = A^-x^{(n)} + Dx^{(n)} - A^-x^{(n)} - Dx^{(n)} - A^+x^{(n)} + b,$$

$$(D + A^-)x^{(n+1)} = (D + A^-)x^{(n)} - Ax^{(n)} + b,$$

$$(D + A^-)(x^{(n+1)} - x^{(n)}) + Ax^{(n)} = b.$$

Вираз $(D + A^-)(x^{(n+1)} - x^{(n)}) + Ax^{(n)} = b$ називають *неявним записом методу Гауса-Зейделя*.

Висновок

Розглянуті ітераційні методи зводяться до двох неявних виразів:

$$D(x^{(n+1)} - x^{(n)}) + Ax^{(n)} = b,$$

$$(D + A^-)(x^{(n+1)} - x^{(n)}) + Ax^{(n)} = b.$$

Якщо дані методи сходяться, то

$$D(x^{(n+1)} - x^{(n)}) \rightarrow 0 \text{ і } (D + A^-)(x^{(n+1)} - x^{(n)}) \rightarrow 0$$

Із цих виразів видно, що якщо ітераційний метод сходиться, то він сходиться до розв'язку початкової системи, оскільки при цьому $Ax^{(n)} = b$.

3.12.7. Ітераційний параметр

Ітераційний параметр – це параметр, який залежить від номера ітерації і введений в ітераційну формулу з метою прискорення збіжності ітераційного процесу [59].

Ітераційний параметр τ_{n+1} можна ввести в ітераційну формулу в такий спосіб:

$$D \frac{x^{(n+1)} - x^{(n)}}{\tau_{n+1}} + Ax^{(n)} = b,$$

$$(D + A^-) \frac{x^{(n+1)} - x^{(n)}}{\tau_{n+1}} + Ax^{(n)} = b.$$

Вибір ітераційних параметрів з'ясовується при дослідженні збіжності.

3.12.8. Канонічна форма запису однокрокових ітераційних методів

Однокроковим ітераційним методом називають такий метод, у якому для знаходження $x^{(n+1)}$ потрібно пам'ятати тільки одну попередню ітерацію $x^{(n)}$. Ітераційний метод може бути записаний у різний спосіб.

Тому вводиться стандартна (канонічна) форма запису ітераційних методів у матричній формі.

Канонічною формою однокрокового ітераційного методу розв'язування системи $Ax = b$ називають запис у вигляді:

$$B_{n+1} \frac{x^{(n+1)} - x^{(n)}}{\tau_{n+1}} + Ax^{(n)} = b, \quad n = 0, 1, \dots, n_0, \quad (3.62)$$

де B_{n+1} – матриця, яка задає той або інший ітераційний метод;

τ_{n+1} – ітераційний параметр.

Нехай задано початкове наближення $x^{(0)}$, існують матриці B_n^{-1} , $n = 1, 2, \dots, n_0$. Тоді з рівняння (3.62) можна послідовно визначити всі $x^{(n)}$, $n = 1, 2, \dots, n_0$.

3.12.9. Узагальнений розв'язок системи лінійних рівнянь

Нехай даний однокроковий метод у канонічній формі:

$$B_{n+1} \frac{x^{(n+1)} - x^{(n)}}{\tau_{n+1}} + Ax^{(n)} = b, \quad n = 0, 1, \dots, n_0$$

Домножимо ліву й праву частину рівняння на τ_{n+1} й згрупуємо доданки:

$$B_{n+1}x^{(n+1)} = B_{n+1}x^{(n)} - \tau_{n+1}Ax^{(n)} + \tau_{n+1}b, \quad n = 0, 1, \dots, n_0$$

Введемо заміну $F_n = (B_{n+1} - \tau_{n+1}A)x^{(n)} + \tau_{n+1}b$. Для знаходження $x^{(n+1)}$ по відомих b і $x^{(n)}$ достатньо розв'язати систему рівнянь:

$$B_{n+1}x^{(n+1)} = F_n,$$

Якщо $B_{n+1} = E$, то такий ітераційний метод називають *явним*. Якщо $B_{n+1} \neq E$, то ітераційний метод називають *неявним*. У цьому випадку E – одинична матриця.

3.12.10. Стаціонарний і нестаціонарний однокрокові методи

Ітераційний метод називають стаціонарним, якщо $B_{n+1} = B$ й $\tau_{n+1} = \tau$.

Наприклад, метод простої ітерації:

$$\frac{x^{(n+1)} - x^{(n)}}{\tau} + Ax^{(n)} = b.$$

Ітераційний метод називають нестаціонарним, якщо використовують змінний ітераційний параметр τ_{n+1} . Наприклад, метод Річардсона:

$$\frac{x^{(n+1)} - x^{(n)}}{\tau_{n+1}} + Ax^{(n)} = b.$$

Узагальненням методу Гауса-Зейделя є метод верхньої релаксації:

$$\left(D + \omega A^{-}\right) \frac{x^{(n+1)} - x^{(n)}}{\omega} + Ax^{(n)} = b,$$

де $\omega > 0$ – заданий числовий параметр.

3.12.11. Метод простої ітерації в координатній формі

Реалізація методу простих ітерацій складається з виконання наступних кроків.

Крок 1. Початкову $Ax = b$ перетворюють до рекурентного вигляду:

$$x = Bx + d,$$

де B – квадратна матриця порядку n ; d – вектор-стовпець.

До початку ітераційного процесу необхідно досягти виконання умови

$$\|B\| < 1, \text{ де}$$

$$\|B\|_l = \max_{1 \leq j \leq n} \sum_{i=1}^n |b_{ij}| \text{ – } l\text{-норма, } \|B\|_m = \max_{1 \leq i \leq n} \sum_{j=1}^n |b_{ij}| \text{ – } m\text{-норма.}$$

Крок 2. Стовпець d використовують як початкове наближення $x^{(0)} = d$ й далі багаторазово виконують дії з уточнення розв'язку згідно рекурентного співвідношення

$$x^{(n+1)} = Bx^{(n)} + d, \quad n = 0, 1, 2, \dots \quad (3.63)$$

Розглянемо принципи формування цього співвідношення.

Нехай дана лінійна система:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1m}x_m = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2m}x_m = b_2, \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mm}x_m = b_m \end{cases} \quad (3.64)$$

Припустимо, що діагональні коефіцієнти $a_{ii} \neq 0$ ($i = 1, 2, \dots, m$).

Запишемо рекурентне співвідношення $x^{(n+1)} = Bx^{(n)} + d$ для системи (3.64) у розгорнутому вигляді:

$$\begin{cases} x_1^{(n+1)} = \alpha_{11}x_1^{(n)} + \alpha_{12}x_2^{(n)} + \alpha_{13}x_3^{(n)} + \dots + a_{1m}x_m^{(n)} + \beta_1, \\ x_2^{(n+1)} = \alpha_{21}x_1^{(n)} + \alpha_{22}x_2^{(n)} + \alpha_{23}x_3^{(n)} + \dots + a_{2m}x_m^{(n)} + \beta_2, \\ \dots \\ x_m^{(n+1)} = \alpha_{m1}x_1^{(n)} + \alpha_{m2}x_2^{(n)} + \alpha_{m3}x_3^{(n)} + \dots + a_{mm}x_m^{(n)} + \beta_m, \end{cases} \quad (3.65)$$

де $\beta_i = \frac{b_i}{a_{ii}}$; $\alpha_{ij} = -\frac{a_{ij}}{a_{ii}}$ при $a_{ii} \neq 0$.

Введемо матриці $\alpha = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1m} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2m} \\ \dots & \dots & \dots & \dots \\ \alpha_{m1} & \alpha_{m2} & \dots & \alpha_{mm} \end{pmatrix}$ та $\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_m \end{pmatrix}$

Використовуючи ці матриці, представимо систему (3.66) у матричній формі

$$x = \alpha x + \beta, \quad (3.66)$$

Систему (3.66) розв'яжемо методом послідовних наближень при нульовому наближенні $x^{(0)} = \beta$. Далі, послідовно будуємо матриці-стовпці:

$$x^{(1)} = \beta + \alpha x^{(0)} \quad (\text{перше наближення}),$$

$$x^{(2)} = \beta + \alpha x^{(1)} \quad (\text{друге наближення}).$$

Наближення в загальному вигляді обчислюють за формулою:

$$x^{(n+1)} = \beta + \alpha x^{(n)} \quad (n = 0, 1, 2, \dots) \quad (3.67)$$

Якщо послідовність наближень $x^{(0)}, x^{(1)}, \dots, x^{(n)}, \dots$ має границю $x = \lim_{n \rightarrow \infty} x^{(n)}$.

Переходячи до границі в рівності (3.67), одержимо:

$$\lim_{n \rightarrow \infty} x^{(n+1)} = \beta + \alpha \lim_{n \rightarrow \infty} x^{(n)} \quad \text{або} \quad x = \beta + \alpha x.$$

Таким чином, граничний вектор x є розв'язком системи $x = \alpha x + \beta$, а отже, і системи $Ax = b$.

Крок 3. Ітерації зупиняють за умови виконання нерівності:

$$\|x^{(n+1)} - x^{(n)}\| < \varepsilon$$

Загальна формула наближень за методом простої ітерації має вигляд:

$$\begin{cases} x_i^{(0)} = \beta_i, \\ x_i^{(n+1)} = \beta_i + \sum_{j=1}^m \alpha_{ij} x_j^{(n)}, \\ (i = 1, \dots, m; n = 0, 1, 2, \dots). \end{cases} \quad (3.68)$$

Умови збіжності

Для того, щоб метод простої ітерації збігався, рівняння системи $Ax = b$ переставляють таким чином, щоб виконувалася умова домінування діагональних елементів:

$$|a_{ii}| \geq |a_{i1}| + \dots + |a_{i,i-1}| + |a_{i,i+1}| + \dots + |a_{im}|, \quad i = 1, 2, \dots, m.$$

і хоча б для одного i нерівність була строгою.

Інакше кажучи, модулі діагональних коефіцієнтів у кожному рівнянні системи більші за суму модулів недиагональних коефіцієнтів (вільні члени не враховують) [60].

Для одержання збіжності перетворення системи $Ax = b$ до вигляду $x = \alpha x + \beta$ з матрицею α , що задовольняє умові збіжності, може бути виконано декількома способами.

Способи забезпечення збіжності методу простих ітерацій

Спосіб 1. Рівняння в системі $Ax = b$ переставляють так, щоб виконувалася умова збіжності

$$|a_{ii}| \geq |a_{i1}| + \dots + |a_{i,i-1}| + |a_{i,i+1}| + \dots + |a_{im}|, \quad i = 1, 2, \dots, m$$

Для досягнення даної мети можна використовувати й інші елементарні еквівалентні перетворення.

Приклад 3.40. Нехай дана система:
$$\begin{cases} -2.8x_1 + x_2 + 4x_3 = 60, \\ 10x_1 - x_2 + 8x_3 = 10, \\ -x_1 + 2x_2 - 0.6x_3 = 20. \end{cases}$$

За допомогою перестановки приведемо її до вигляду:

$$\begin{cases} 10x_1 - x_2 + 8x_3 = 10, \\ -x_1 + 2x_2 - 0.6x_3 = 20, \\ -2.8x_1 + x_2 + 4x_3 = 60. \end{cases}$$

Для даної системи умови збіжності мають вигляд :

$$|10| > |-1| + |8|, \quad |2| > |-1| + |-0.6|, \quad |4| > |-2.8| + |1|$$

Діагональні елементи мають домінування.

Виразимо x_1 з першого рівняння, x_2, x_3 – з наступних.

Одержимо систему $x = \alpha x + \beta$:

$$\begin{cases} x_1 = 0x_1 + 0.1x_2 - 0.8x_3 + 1, \\ x_2 = 0.5x_1 + 0x_2 + 0.3x_3 + 10, \\ x_3 = 0.7x_1 - 0.25x_2 + 0x_3 + 15, \end{cases} \quad \alpha = \begin{pmatrix} 0 & 0.1 & -0.8 \\ 0.5 & 0 & 0.3 \\ 0.7 & -0.25 & 0 \end{pmatrix}, \quad \beta = \begin{pmatrix} 1 \\ 10 \\ 15 \end{pmatrix}.$$

Помітимо, що $\|\alpha\|_m = \max\{0.9, 0.8, 0.95\} = 0.95 < 1$, тобто умова збіжності виконана.

Приклад 3.41. Одержання діагонального домінування шляхом додавання рівнянь системи.

Нехай дана система:
$$\begin{cases} 4x_1 + x_2 + 9x_3 = -7, \\ 3x_1 + 8x_2 - 7x_3 = -6, \\ x_1 + x_2 - 8x_3 = 7. \end{cases}$$

Додамо 1-е рівняння до 3-го і віднімемо 2-е від 3-го:

$$\begin{array}{rcl} 4x_1 + x_2 + 9x_3 = -7 & & 3x_1 + 8x_2 - 7x_3 = -6 \\ + & & - \\ x_1 + x_2 - 8x_3 = 7 & & x_1 + x_2 - 8x_3 = 7 \\ = 5x_1 + 2x_1 + x_3 = 0 & & = 2x_1 + 7x_1 + x_3 = -13 \end{array}$$

$$\begin{cases} 5x_1 + 2x_2 + x_3 = 0, \\ 2x_1 + 7x_2 + x_3 = -13, \\ x_1 + x_2 - 8x_3 = 7. \end{cases}$$

Спосіб 2. Рівняння в системі $Ax = b$ переставляють так, щоб виконувалася умова збіжності

$$|a_{ii}| \geq |a_{i1}| + \dots + |a_{i,i-1}| + |a_{i,i+1}| + \dots + |a_{im}|, \quad i = 1, 2, \dots, m$$

Перетворення виконують таким чином, щоб елементи на діагоналі переважали, але при цьому коефіцієнти a_{ii} не обов'язково дорівнювали нулю.

Приклад 3.42. Нехай дана система
$$\begin{cases} 1.02x_1 - 0.15x_2 = 2.7, \\ 0.8x_1 + 1.05x_2 = 4 \end{cases}$$

Запишемо дану систему у формі:

$$\begin{cases} x_1 = -0.02x_1 + 0.15x_2 + 2.7 \\ x_2 = -0.8x_1 - 0.05x_2 + 4 \end{cases}, \text{ де } \alpha = \begin{pmatrix} -0.02 & 0.15 \\ -0.8 & -0.05 \end{pmatrix}$$

При цьому $\|\alpha\|_m = \max\{0.17; 0.85\} = 0.85 < 1$

Спосіб 3. Якщо $\det A \neq 0$, то систему $Ax = b$ можна помножити на матрицю: $D = A^{-1} - \varepsilon$, де $\varepsilon = (\varepsilon_{ij})$ – матриця з малими за модулем елементами. Тоді отримуємо систему $(A^{-1} - \varepsilon)Ax = Db$ або після розкриття дужок: $A^{-1}Ax - \varepsilon Ax = Db$ або $x - \varepsilon Ax = Db$.

Увівши заміни $\alpha = \varepsilon Ax$, $\beta = Db$, одержимо формулу для методу простої ітерації: $x = \alpha x + \beta$.

Якщо елементи $|\varepsilon_{i,j}|$, $i, j = 1, 2, \dots, m$ достатньо малі, то умова збіжності виконується.

Алгоритм методу простих ітерацій

1. Перетворити систему $Ax = b$ до вигляду $x = \alpha x + \beta$ одним з розглянутих способів.

2. Задати початкове наближення розв'язку $x^{(0)}$ довільно або покласти $x^{(0)} = \beta$, а також мале додатне число ε (точність). Покласти $n = 0$

3. Обчислити чергове наближення $x^{(n+1)}$ за формулою

$$x^{(n+1)} = \alpha x^{(n)} + \beta.$$

4. Якщо виконана умова $\|x^{(n+1)} - x^{(n)}\| < \varepsilon$, то процес обчислення ітерацій завершити й установити наближений розв'язок задачі $x_* \cong x^{(n+1)}$

5. Якщо $\|x^{(n+1)} - x^{(n)}\| \geq \varepsilon$, то встановлюємо $n := n + 1$ й переходимо до п. 3.

Приклад 3.43. Методом простих ітерацій з точністю $\varepsilon = 0.01$ розв'язати таку систему алгебраїчних рівнянь:

$$\begin{cases} 2x_1 + 2x_2 + 10x_3 = 14, \\ 10x_1 + x_2 + x_3 = 12, \\ 2x_1 + 10x_2 + x_3 = 13. \end{cases}$$

Розв'язок

Етап 1. Перевіряємо умову збіжності

$$|2| < |2| + |10|, \quad |1| < |10| + |1|, \quad |1| < |2| + |10|$$

Очевидно, що умова не виконується.

Переставимо рівняння так, щоб виконувалася умова домінування діагональних елементів.

$$\begin{cases} 10x_1 + x_2 + x_3 = 12, \\ 2x_1 + 10x_2 + x_3 = 13, \\ 2x_1 + 2x_2 + 10x_3 = 14. \end{cases}$$

Після цього одержуємо:

$$|10| > |1| + |1|, \quad |10| > |2| + |1|, \quad |10| > |2| + |2|.$$

Виразимо з першого рівняння x_1 , а x_2, x_3 – з другого й третього.

$$\begin{cases} x_1 = -0.1x_2 - 0.1x_3 + 1.2, \\ x_2 = -0.2x_1 - 0.1x_3 + 1.3, \\ x_3 = -0.2x_1 - 0.2x_2 + 1.4 \end{cases}, \alpha = \begin{pmatrix} 0 & -0.1 & -0.1 \\ -0.2 & 0 & -0.1 \\ -0.2 & -0.2 & 0 \end{pmatrix}, \beta = \begin{pmatrix} 1.2 \\ 1.3 \\ 1.4 \end{pmatrix}$$

Помітимо, що $\|\alpha\|_m = \max\{0.2, 0.3, 0.4\} = 0.4 < 1$ умова збіжності виконана.

Етап 2. Задамо $x^{(0)} = \beta = \begin{pmatrix} 1.2 \\ 1.3 \\ 1.4 \end{pmatrix}$. У даній задачі $\varepsilon = 0.01$.

Етап 3. Виконаємо розрахунки за формулами обчислення:

$$\begin{cases} x_1^{(n+1)} = -0.1x_2^{(n)} - 0.1x_3^{(n)} + 1.2, \\ x_2^{(n+1)} = -0.2x_1^{(n)} - 0.1x_3^{(n)} + 1.3, \\ x_3^{(n+1)} = -0.2x_1^{(n)} - 0.2x_2^{(n)} + 1.4. \end{cases}$$

Результати обчислень занесемо в таблицю 3.16.

Таблиця 3.16

Проміжні обчислення методом простих ітерацій

n	$x_1^{(n)}$	$x_2^{(n)}$	$x_3^{(n)}$	$\ x^{(n)} - x^{(n-1)}\ $
0	1.2000	1.3000	1.4000	-
1	0.9300	0.9200	0.9000	0.5000
2	1.0180	1.0240	1.0300	0.1300
3	0.9946	0.9934	0.9916	0.0384
4	1.0015	1.0020	1.0024	0.0108
5	0.9996	0.9995	0.9993	0.0031 < ε

3.12.12. Модифікація методу ітерацій

Відзначимо, що іноді буває зручніше обчислювати не наближення, а їх різниці. Введемо позначення:

$$\Delta^{(n)} = x^{(n)} - x^{(n-1)} \quad (n = 0, 1, 2, \dots),$$

Використаємо базовий вираз для однокрокових методів, одержуємо:

$$x^{(n+1)} = \alpha x^{(n)} + \beta \quad \text{та} \quad x^{(n)} = \alpha x^{(n-1)} + \beta$$

Відніmemo першу рівність від другої:

$$x^{(n+1)} - x^{(n)} = \alpha(x^{(n)} - x^{(n-1)}).$$

Позначимо $\Delta^{(n+1)} = x^{(n+1)} - x^{(n)}$, $\Delta^{(n)} = x^{(n)} - x^{(n-1)}$,

Тоді $\Delta^{(n+1)} = \alpha\Delta^{(n)}$ ($n = 1, 2, \dots$).

За нульове наближення приймаємо:

$$\Delta^{(0)} = x^{(0)},$$

$$\Delta^{(0)} + \Delta^{(1)} = x^{(0)} + x^{(1)} - x^{(0)} = x^{(1)}$$

$$\Delta^{(0)} + \Delta^{(1)} + \Delta^{(2)} = x^{(0)} + x^{(1)} - x^{(0)} + x^{(2)} - x^{(1)} = x^{(2)}$$

$$\Delta^{(0)} + \Delta^{(1)} + \Delta^{(2)} + \Delta^{(3)} = x^{(0)} + x^{(1)} - x^{(0)} + x^{(2)} - x^{(1)} + x^{(3)} - x^{(2)} = x^{(3)}$$

У загальному випадку m -е наближення є

$$x^{(m)} = \sum_{n=0}^m \Delta^{(n)}.$$

Нехай $\Delta^{(0)} = x^{(0)} = \beta$. Тоді рівність $\Delta^{(n+1)} = \alpha\Delta^{(n)}$ ($n = 1, 2, \dots$).

буде виконана й при $n = 0$. Звідси випливає відповідна методика обчислення цього варіанта ітерацій.

Алгоритм модифікованого методу простих ітерацій

Якщо

1. $\Delta^{(0)} = x^{(0)} = \beta$, то $\Delta^{(n)} = \alpha\Delta^{(n-1)} = \alpha^n\beta$ ($n = 0, 1, 2, \dots$) і

$$x^{(n)} = \sum_{k=0}^n \Delta^{(k)} = \sum_{k=0}^n \alpha^k \beta.$$

2. Якщо ж $\Delta^{(0)} = x^{(0)} \neq \beta$, то знаходимо

$$\Delta^{(1)} = x^{(1)} - x^{(0)} = \alpha x^{(0)} + \beta - x^{(0)}$$

і вважаємо, що $\Delta^{(n)} = \alpha\Delta^{(n-1)} = \alpha^{n-1}\Delta^{(1)}$ ($n = 1, 2, 3, \dots$).

Отже, $x^{(n)} = \sum_{k=0}^n \Delta^{(k)} = x^{(0)} + \sum_{k=1}^n \alpha^{k-1}\Delta^{(1)}$.

Приклад 3.44. Розв'язати систему
$$\begin{cases} 2x_1 - x_2 + x_3 = -3, \\ 3x_1 + 5x_2 - 2x_3 = 1, \\ x_1 - 4x_2 + 10x_3 = 0. \end{cases}$$

Розв'язок. Вирішуємо рівняння щодо відповідних змінних:

$$\begin{cases} x_1 = -1.5 + 0.5x_2 - 0.5x_3, \\ x_2 = 0.2 - 0.6x_1 + 0.4x_3, \\ x_3 = -0.1x_1 + 0.4x_2 \end{cases}$$

У цьому прикладі $\alpha = \begin{pmatrix} 0 & 0.5 & -0.5 \\ -0.6 & 0 & 0.4 \\ -0.1 & 0.4 & 0 \end{pmatrix}$, $\beta = \begin{pmatrix} -1.5 \\ 0.2 \\ 0 \end{pmatrix}$.

Користуючись формулами:

$$\Delta^{(n+1)} = \alpha\Delta^{(n)}, x^{(n)} = \sum_{k=0}^n \Delta^{(k)} = \sum_{k=0}^n \alpha^k \beta, (n=1, 2, \dots)$$

одержимо:

$$\Delta^{(1)} = \alpha\Delta^{(0)} = \begin{pmatrix} 0 & 0.5 & -0.5 \\ -0.6 & 0 & 0.4 \\ -0.1 & 0.4 & 0 \end{pmatrix} \cdot \begin{pmatrix} -1.5 \\ 0.2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.9 \\ 0.23 \end{pmatrix},$$

$$\Delta^{(2)} = \alpha\Delta^{(1)} = \begin{pmatrix} 0 & 0.5 & -0.5 \\ -0.6 & 0 & 0.4 \\ -0.1 & 0.4 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0.1 \\ 0.9 \\ 0.23 \end{pmatrix} = \begin{pmatrix} 0.335 \\ 0.032 \\ 0.350 \end{pmatrix} \text{ і т. д.}$$

Виконавши таким чином 9 ітерацій, одержимо: $x^{(9)} = \sum_{k=0}^9 \Delta^{(k)} = \begin{pmatrix} -1.235 \\ 1.089 \\ 0.560 \end{pmatrix}$

Переваги модифікованого методу

Простий обчислювальний алгоритм, що дозволяє використовувати стандартні алгоритми прискореного множення матриць.

Недоліки модифікованого методу.

1. Систематичне нагромадження помилок при збільшенні числа доданків, у результаті чого можуть виникнути значні похибки шуканого кореня.

2. Помилка, допущена в обчисленнях, впливає на остаточний результат. Тому надійніше користуватися першим варіантом ітерації.

3.12.13. Достатня умова збіжності процесу ітерації в координатній формі

Теорема 3.18. *Про збіжність методу простих ітерацій*

Якщо для системи $x = \alpha x + \beta$ виконана щонайменше одна з умов

$$\sum_{j=1}^n |\alpha_{ij}| < 1 \text{ або } \sum_{i=1}^n |\alpha_{ij}| < 1 \quad (j = 1, 2, \dots, n),$$

то процес ітерації $x^{(n+1)} = \alpha x^{(n)} + \beta$ сходиться до єдиного розв'язку цієї системи, незалежно від вибору початкового наближення.

Наслідок. Для системи

$$\sum_{j=1}^n a_{ij} x_j = b_i \quad (i = 1, 2, \dots, n)$$

метод ітерації сходиться, якщо виконані нерівності [61]:

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \quad (i = 1, 2, \dots, n).$$

3.12.14. Метод Якобі в координатній формі

Розглянемо систему лінійних алгебраїчних рівнянь: $Ax = b$, де матриця $A = (a_{ij})$ $i, j = 1, 2, \dots, m$ має обернену матрицю,

$$x = (x_1, x_2, \dots, x_m)^T, \quad b = (b_1, b_2, \dots, b_m)^T.$$

Перетворимо систему до вигляду:

$$x_i = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j - \sum_{j=i+1}^m \frac{a_{ij}}{a_{ii}} x_j + \frac{b_i}{a_{ii}}, \quad i = 1, 2, \dots, m$$

При цьому передбачається, що всі $a_{ii} \neq 0$.

Уникнути цієї проблеми іноді вдається перестановкою рівнянь у системі $Ax = b$.

Представимо вектор n -ї ітерації у вигляді: $x^{(n)} = (x_1^{(n)}, x_2^{(n)}, \dots, x_m^{(n)})^T$.

Тоді метод Якобі для системи $Ax = b$ в координатній формі має вигляд

$$x_i^{(n+1)} = -\sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{(n)} - \sum_{j=i+1}^m \frac{a_{ij}}{a_{ii}} x_j^{(n)} + \frac{b_i}{a_{ii}}, \quad i = 1, 2, \dots, m,$$

Виключивши з суми в даному виразі елемент $j = i$, одержимо загальний вираз для методу Якобі в компактному вигляді [62]:

$$x_i^{(n+1)} = -\frac{1}{a_{ii}} \left(\sum_{\substack{j=1 \\ j \neq i}}^m a_{ij} x_j^{(n)} - b_i \right), \quad i = 1, \dots, m$$

Початкові значення

Початкові значення $x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_m^{(0)})^T$ вибирають довільно.

У більшості випадків $x_i^{(0)} = 0, \quad i = 1, 2, \dots, m$.

Умова закінчення алгоритму

Закінчення ітерацій визначають або задаванням максимального числа ітерацій n_{\max} , або умовою $\max_{1 \leq i \leq m} |x_i^{(n+1)} - x_i^{(n)}| < \varepsilon$, де $\varepsilon > 0$.

Умова збіжності алгоритму

Теорема 3.19. *Про збіжність методу Якобі*

Нехай A – матриця з діагональною перевагою, тобто

$$|a_{ii}| > \sum_{i \neq j} |a_{ij}|.$$

Тоді метод Якобі сходиться.

Приклад 3.45. Методом Якобі розв'язати систему лінійних рівнянь:

$$\begin{cases} 2x_1 - 1,8x_2 + 0,4x_3 = 1 & (I) \\ 3x_1 + 2x_2 - 1,1x_3 = 0 & (II) \\ x_1 - x_2 + 7,3x_3 = 0 & (III) \end{cases}$$

попередньо перетворивши матрицю системи до матриці з діагональною перевагою. Ітерації виконувати до досягнення точності $\varepsilon = 0.001$.

Розв'язок. Приведемо систему до такого вигляду, щоб елементи головної діагоналі були би більшими за інші елементи рядків:

$$\begin{cases} 25x_1 + x_2 - 3,5x_3 = 5 & (5I + 5II) \\ -9,4x_2 + 3,4x_3 = 3 & (3I - 2II) \\ x_1 - x_2 + 7,3x_3 = 0 & (III) \end{cases}$$

Нульове наближення $x^{(0)} = 0$.

Обчислимо перше наближення: $x_i^{(1)} = -\frac{1}{a_{ii}} \left(\sum_{\substack{j=1 \\ j \neq i}}^m a_{ij} x_j^{(0)} - b_i \right)$, $i = 1, 2, 3$,

де a_{ij} – елементи матриці

$$A = \begin{pmatrix} 25 & 1 & -3,5 \\ 0 & -9,4 & 3,4 \\ 1 & -1 & 7,3 \end{pmatrix},$$

а b_i – елементи вектора $b = (5 \ 3 \ 0)^T$.

$$x_1^{(1)} = -\frac{1}{25} (x_2^{(0)} - 3,5x_3^{(0)} - 5) = \frac{0 - 3,5 \cdot 0 - 5}{25} = -0,2,$$

$$x_2^{(1)} = \frac{1}{9,4} (3,4x_3^{(0)} - 3) = \frac{3,4 \cdot 0 - 3}{9,4} = -0,3191,$$

$$x_3^{(1)} = -\frac{1}{7,3} (x_1^{(0)} - x_2^{(0)} - 0) = 0$$

Виконаємо другу ітерацію:

$$x_1^{(2)} = -\frac{1}{25} (x_2^{(1)} - 3,5x_3^{(1)} - 5) = \frac{-0,3191 - 3,5 \cdot 0 - 5}{25} = -\frac{5,319}{25} = 0,213$$

$$x_2^{(2)} = \frac{1}{9,4} (3,4x_3^{(1)} - 3) = \frac{3,4 \cdot 0 - 3}{9,4} = -\frac{3}{9,4} = -0,3191,$$

$$x_3^{(2)} = -\frac{1}{7,3} (x_1^{(1)} - x_2^{(1)} - 0) = \frac{0,2 + 0,3191}{-7,3} = \frac{0,519}{-7,3} = -0,0071$$

Повторюємо ітерації, поки для вектора $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, x_3^{(i)})$

буде виконана умова $\max_{1 \leq i \leq 3} |x_i^{(n+1)} - x_i^{(n)}| < \varepsilon$.

3.12.15. Метод Гауса-Зейделя в координатній формі

Метод Гауса-Зейделя – модифікація методу ітерації [63]. Основна його ідея полягає в тому, що при обчисленні $(n+1)$ наближення невідомого x_i враховують вже обчислені раніше $(n+1)$ наближення невідомих x_1, x_2, \dots, x_{j-1} .

Нехай дана лінійна система:

$$x_i = \sum_{j=1}^m \alpha_{ij} x_j + \beta_i \quad (i = 1, 2, \dots, m).$$

Виберемо довільно початкові наближення коренів

$$x_1^{(0)}, x_2^{(0)}, \dots, x_m^{(0)},$$

намагаючись, звичайно, щоб вони деякою мірою відповідали шуканим невідомим x_1, x_2, \dots, x_n .

Далі, припускаючи, що n наближення $x_i^{(n)}$ коренів відомі, згідно Зейделю будемо будувати $(n+1)$ наближення коренів за наступними формулами:

$$x_1^{(n+1)} = \sum_{j=1}^m \alpha_{1j} x_j^{(n)} + \beta_1;$$

$$x_2^{(n+1)} = \alpha_{21} x_1^{(n+1)} + \sum_{j=2}^m \alpha_{2j} x_j^{(n)} + \beta_2;$$

.....

$$x_i^{(n+1)} = \sum_{j=1}^{i-1} \alpha_{ij} x_j^{(n+1)} + \sum_{j=i}^m \alpha_{ij} x_j^{(n)} + \beta_i;$$

.....

$$x_n^{(n+1)} = \sum_{j=1}^{m-1} a_{mj} x_j^{(n+1)} + a_{mm} x_m^{(n)} + \beta_m \quad (m = 0, 1, 2, \dots)$$

Алгоритм методу релаксації

1. Перетворити систему до виду, зручного для релаксації.
2. Обчислити нев'язки для кожного рівняння на ітераційному кроці (n) .
3. Вибрати максимальну з отриманих нев'язок.
4. Обчислити нев'язки на ітераційному кроці $(n + 1)$, виконавши наступні дії:
 - максимальну нев'язку покласти рівною нулю;
 - інші нев'язки збільшити на величину $b_{ij}R_j^{(n)}$ при $j \neq i$
5. Якщо величини всіх нев'язок не дорівнюють нулю із заданою точністю, то перейти до п. 3.
6. Розв'язок для кожного x_i дорівнює сумі приростів для кожної з відповідних нев'язок.

Приклад 3.47. Методом релаксації розв'язати систему:

$$\begin{cases} 10x_1 - 2x_2 - 2x_3 = 6, \\ -x_1 + 10x_2 - 2x_3 = 7, \\ -x_1 - x_2 + 10x_3 = 8. \end{cases}$$

виконуючи обчислення з двома десятковими знаками.

Розв'язок. Приводимо дану систему до виду, зручного для релаксації:

$$\begin{cases} -x_1 + 0.2x_2 + 0.2x_3 + 0.6 = 0, \\ -x_2 + 0.1x_1 + 0.2x_3 + 0.7 = 0, \\ -x_3 + 0.1x_1 + 0.1x_2 + 0.8 = 0. \end{cases}$$

Вибираючи як початкові наближення коренів нульові значення

$$x_1^{(0)} = x_2^{(0)} = x_3^{(0)} = 0,$$

знаходимо відповідні їм нев'язки

$$R_1^{(0)} = 0.60; R_2^{(0)} = 0.70; R_3^{(0)} = 0.80.$$

Згідно із загальною теорією вважаємо: $\delta x_3^{(0)} = 0.80$.

Звідси одержуємо нев'язки:

$$R_1^{(1)} = R_1^{(0)} + b_{13}R_3^{(0)} = R_1^{(0)} + 0.2 \cdot 0.8 = 0.60 + 0.16 = 0.76;$$

$$R_2^{(1)} = R_2^{(0)} + b_{23}R_3^{(0)} = R_2^{(0)} + 0.2 \cdot 0.8 = 0.70 + 0.16 = 0.86;$$

$$R_3^{(1)} = 0.$$

$$R_1^{(1)} = 0.76; R_2^{(1)} = 0.86; R_3^{(1)} = 0.$$

$$R_1^{(1)} = 0.76; R_2^{(1)} = 0.86; R_3^{(1)} = 0.$$

Далі вважаємо: $\delta x_2^{(1)} = 0.86$

$$R_1^{(2)} = R_1^{(1)} + b_{12}R_2^{(1)} = 0.76 + 0.2 \cdot 0.86 = 0.932;$$

$$R_2^{(2)} = 0;$$

$$R_3^{(2)} = R_3^{(1)} + b_{32}R_2^{(1)} = 0 + 0.1 \cdot 0.86 = 0.086$$

$$R_1^{(2)} = 0.932; R_2^{(2)} = 0; R_3^{(2)} = 0.086$$

Далі вважаємо: $\delta x_1^{(2)} = 0.932$.

$$R_1^{(3)} = 0;$$

$$R_2^{(3)} = R_2^{(2)} + b_{21}R_1^{(2)} = 0 + 0.1 \cdot 0.932 = 0.093$$

$$R_3^{(3)} = R_3^{(2)} + b_{31}R_1^{(2)} = 0.086 + 0.1 \cdot 0.932 = 0.179$$

$$R_1^{(3)} = 0; R_2^{(3)} = 0.093; R_3^{(3)} = 0.179$$

Далі вважаємо $\delta x_3^{(3)} = 0.179$ і т. д. до одержання заданої точності

нев'язок.

Фінальний етап. Додавши усі прирости $\delta x_i^{(n)}$ ($i = 1, 2, 3; n = 0, 1, \dots$),

одержимо значення коренів:

$$x_1 = 0 + 0.93 + 0.07 = 1.0,$$

$$x_2 = 0 + 0.86 + 0.13 + 0.01 = 1.00,$$

$$x_3 = 0 + 0.80 + 0.18 + 0.02 = 1.00.$$

Контрольні запитання

1. Розв'язати систему алгебраїчних рівнянь методом Якобі з точністю $\varepsilon = 0.01$.

$$\begin{cases} x_1 + x_2 + x_3 + 11x_4 = -20, \\ 15x_1 + 2x_2 - x_3 - x_4 = 22, \\ 2x_1 + x_2 + 12x_3 + x_4 = -10, \\ x_1 - 10x_2 - x_3 - 2x_4 = -14. \end{cases} \quad x(0) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

2. Методом простих ітерацій з точністю $\varepsilon = 0.01$ розв'язати таку систему алгебраїчних рівнянь:

$$\begin{cases} 2x_1 + 2x_2 + 10x_3 = 14, \\ 12x_1 + 11x_2 + 2x_3 = 25, \\ 2x_1 + 10x_2 + x_3 = 13. \end{cases}$$

3. Виконати 3 ітерації розв'язку лінійних алгебраїчних рівнянь модифікованим методом простих ітерацій.

$$\begin{cases} x_1 - 4x_2 + 10x_3 = 0, \\ 5x_1 + 4x_2 - x_3 = -2, \\ 3x_1 + 5x_2 - 2x_3 = 1. \end{cases}$$

4. Методом Гауса-Зейделя знайти розв'язок СЛАР з точністю $\varepsilon = 0.01$

$$\begin{cases} x_1 + 6x_2 + 2x_3 = 9, \\ 4x_1 - x_2 + x_3 = 4, \\ -x_1 - 2x_2 + 5x_3 = 2. \end{cases}$$

5. Розв'язати систему алгебраїчних рівнянь методом Гауса-Зейделя

$$\begin{cases} 2x_1 + 2x_2 + 10x_3 = 14, \\ 10x_1 + x_2 + x_3 = 12, \\ 2x_1 + 10x_2 + x_3 = 13, \end{cases} \quad \text{з точністю } \varepsilon = 0.01$$

3.13. Розв'язування систем нелінійних рівнянь

3.13.1. Постановка задачі

Розглянуті раніше системи лінійних алгебраїчних рівнянь – це лише окремий випадок систем рівнянь. На практиці частіше зустрічаються системи нелінійних рівнянь [65].

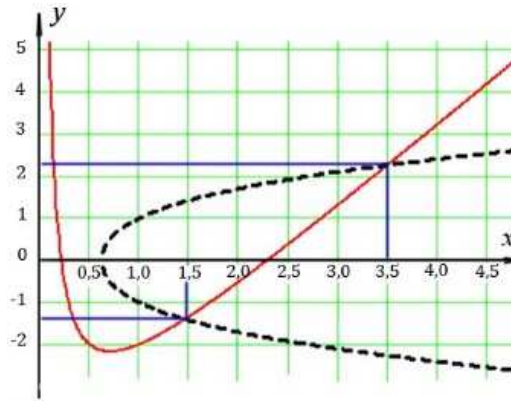


Рис.3.26. Графічний розв’язок системи нелінійних рівнянь

Приведемо систему (3.70) до загального вигляду:

$$\begin{cases} 2x_1^2 - x_1x_2 - 5x_1 + 1 = 0, \\ x_1 + 3\lg x_1 - x_2^2 = 0. \end{cases} \quad (3.71)$$

На відміну від систем лінійних рівнянь, для систем нелінійних рівнянь немає прямих методів розв’язування. Лише в окремих випадках систему можна розв’язати безпосередньо. Наприклад, для системи з двох рівнянь іноді вдається подати одне невідоме через інше й у такий спосіб звести задачу до розв’язання одного нелінійного рівняння. Тому ітераційні методи для нелінійних систем набувають особливої актуальності. Розглянемо кілька найпростіших ітераційних методів розв’язування систем нелінійних рівнянь, а саме, метод простої ітерації, метод Зейделя і метод Ньютона.

3.13.2. Метод простої ітерації

Для реалізації цього методу стосовно заданої нелінійної системи рівнянь треба виконати наступну послідовність кроків.

1. Шляхом алгебраїчних перетворень виокремити з кожного рівняння по одній змінній і в такий спосіб привести систему (3.70) до вигляду:

$$\begin{cases} x_1 = \varphi_1(x_1, x_2, \dots, x_n), \\ x_2 = \varphi_2(x_1, x_2, \dots, x_n), \\ \dots \\ x_n = \varphi_n(x_1, x_2, \dots, x_n), \end{cases} \quad (3.72)$$

2. Обрати вектор початкового наближення $x^{(0)}$:

$$x^{(0)} = (x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}) \quad (3.73)$$

3. Підставити вектор початкового наближення в систему (3.73).

4. З першого рівняння обчислити нове наближення до першої змінної $x_1^{(1)}$, з другого – до другої змінної $x_2^{(1)}$, тощо.

5. Обчислені уточнені значення змінних знову підставляти у рівняння (3.72).

6. Отже, на $(k + 1)$ -му кроці ітераційної процедури формула методу ітерацій для розв'язання системи нелінійних рівнянь має вигляд:

$$\begin{cases} x_1^{(k+1)} = \varphi_1(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}), \\ x_2^{(k+1)} = \varphi_2(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}), \\ \dots \\ x_n^{(k+1)} = \varphi_n(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}), \end{cases} \quad (3.74)$$

7. Важливо забезпечити збіжність ітераційного процесу. Процес ітерації для системи нелінійних рівнянь (3.72) збігається до єдиного її розв'язку, якщо кожна норма матриці $\Phi'(x)$ в заданому околі є меншою за одиницю, тобто для збіжності розв'язків системи достатньою є умова:

$$\|\Phi'(x)\| < 1 \quad (3.75)$$

де $\Phi'(x)$ – матриця частинних похідних, яку називають матрицею Якобі:

$$\Phi'(x) = \begin{pmatrix} \frac{\partial \varphi_1}{\partial x_1} & \frac{\partial \varphi_1}{\partial x_2} & \dots & \frac{\partial \varphi_1}{\partial x_n} \\ \frac{\partial \varphi_2}{\partial x_1} & \frac{\partial \varphi_2}{\partial x_2} & \dots & \frac{\partial \varphi_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial \varphi_n}{\partial x_1} & \frac{\partial \varphi_n}{\partial x_2} & \dots & \frac{\partial \varphi_n}{\partial x_n} \end{pmatrix} \quad (3.76)$$

8. Перевірити умову завершення ітераційного процесу.

$$\frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)}\|} \leq \varepsilon \quad (3.77)$$

де ε – похибка розв’язку системи рівнянь; $\|x^{(k+1)} - x^{(k)}\|, \|x^{(k)}\|$ – норми векторів різниці останнього та передостаннього наближень відповідно.

Приклад 3.49. Розв’яжемо методом простих ітерацій систему нелінійних рівнянь (3.71) з похибкою 0.0001.

1. Перетворимо цю систему до вигляду (3.72), за якого можна використовувати метод простих ітерацій:

$$\begin{cases} x_1 = \sqrt{\frac{x_1(x_2 + 5) - 1}{2}}, \\ x_2 = \sqrt{x_1 + 3 \lg x_1}, \end{cases} \quad (3.78)$$

2. Оберемо вектор початкового наближення:

$$x^{(0)} = \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} = \begin{pmatrix} 3.5 \\ 2.2 \end{pmatrix}.$$

Ці значення одержано в результаті побудови графіка рівнянь, показаного на рис. 3.26.

3. Обчислимо перше наближення:

$$\begin{aligned} x_1^{(1)} &= \sqrt{\frac{x_1^{(0)}(x_2^{(0)} + 5) - 1}{2}} = \sqrt{\frac{3.5(2.2 + 5) - 1}{2}} = 3.478505, \\ x_2^{(1)} &= \sqrt{x_1^{(0)} + 3 \lg x_1^{(0)}} = \sqrt{3.5 + 3 \lg 3.5} = 2.265436. \end{aligned}$$

4. Формули для обчислення $(k+1)$ -го наближення мають вигляд:

$$\begin{cases} x_1^{(k+1)} = \sqrt{\frac{x_1^{(k)}(x_2^{(k)} + 5) - 1}{2}}, \\ x_2^{(k+1)} = \sqrt{x_1^{(k)} + 3 \lg x_1^{(k)}}. \end{cases}$$

Результати ітераційного процесу наведені в таблиці 3.17.

Таблиця 3.17

Проміжні обчислення методом простої ітерації

k – номер ітерації	$x_1^{(k)}$	$x_2^{(k)}$
0	3.5	2.2
1	3.478505	2.265436
2	3.483738	2.258912
3	3.484834	2.260503
4	3.485804	2.260836
5	3.486391	2.261131
6	3.486771	2.261309
7	3.487013	2.261424

Обчислимо похибки вектора наближень рівнянь за формулою (3.77)

після 3-ї ітерації:
$$\frac{\|x^{(3)} - x^{(2)}\|}{\|x^{(2)}\|} \leq \varepsilon .$$

Обчислимо відповідні норми за формулами:

$$\|x^{(2)}\| = \sqrt{|x_1^{(2)}|^2 + |x_2^{(2)}|^2} = \sqrt{12.136430 + 5.102683} = 4.1558503$$

$$\begin{aligned} \|x^{(3)} - x^{(2)}\| &= \sqrt{|x_1^{(3)} - x_1^{(2)}|^2 + |x_2^{(3)} - x_2^{(2)}|^2} = \\ &= \sqrt{|0.001096|^2 + |0.001591|^2} = \sqrt{0.0000012013 + 0.000002531} = \\ &= \sqrt{0.00000373222} = 0.001932 \end{aligned}$$

$$\frac{\|x^{(3)} - x^{(2)}\|}{\|x^{(2)}\|} = \frac{0.001932}{4.1520012} = 0.000465 < 0.001.$$

Обчислимо похибки вектора наближень рівнянь після 7-ї ітерації:

$$\frac{\|x^{(7)} - x^{(6)}\|}{\|x^{(6)}\|} \leq \varepsilon.$$

Одержимо після аналогічних обчислень:

$$\frac{\|x^{(7)} - x^{(6)}\|}{\|x^{(6)}\|} = \frac{0.0002682}{4.1558503} = 0.0000645 < 0.0001.$$

Відповідь: $x_1 = 3.487013$; $x_2 = 2.261424$.

Процес ітерації для системи нелінійних рівнянь (3.72) збігається до єдиного її розв'язку, якщо кожна норма матриці $\Phi'(x)$ в заданому околі є меншою за одиницю, тобто для збіжності розв'язків системи достатньою є умова [66]: $\|\Phi'(x)\| < 1$.

Перевіримо збіжність системи нелінійних рівнянь (3.78), одержаних з системи (3.70), для методу ітерацій. Обчислимо матрицю частинних похідних:

$$\Phi'(x) = \begin{pmatrix} \frac{\partial \Phi_1}{\partial x_1} & \frac{\partial \Phi_1}{\partial x_2} \\ \frac{\partial \Phi_2}{\partial x_1} & \frac{\partial \Phi_2}{\partial x_2} \end{pmatrix}$$

$$\Phi'(x) = \begin{pmatrix} \frac{x_2 + 5}{4\sqrt{\frac{x_1(x_2 + 5) - 1}{2}}} & \frac{x_1}{4\sqrt{\frac{x_1(x_2 + 5) - 1}{2}}} \\ 1 + \frac{3 \cdot 0.4343}{x_1} & 0 \\ \frac{x_1}{2\sqrt{x_1 + 3 \lg x_1}} & 0 \end{pmatrix}$$

У околі точки ($x_1 = 3.5 \pm 0.1$, $x_2 = 2.2 \pm 0.1$) значення частинних похідних задовольняють умовам:

$$\left| \frac{\partial \varphi_1}{\partial x_1} \right| = 0.52, \quad \left| \frac{\partial \varphi_1}{\partial x_2} \right| = 0.25, \quad \left| \frac{\partial \varphi_2}{\partial x_1} \right| = 0.30, \quad \left| \frac{\partial \varphi_2}{\partial x_2} \right| = 0.$$

Підставимо ці значення в матрицю $\Phi'(x)$ та обчислимо її норми:

1. m - норма матриці:

$$\|\Phi'(x)\|_m = \left\| \left\| \frac{\partial \varphi_1}{\partial x_1} \right\| + \left\| \frac{\partial \varphi_1}{\partial x_2} \right\|, \left\| \frac{\partial \varphi_2}{\partial x_1} \right\| + \left\| \frac{\partial \varphi_2}{\partial x_2} \right\| \right\|_m = \|0,77,0,3\|_m = 0.77$$

2. l – норма матриці:

$$\|\Phi'(x)\|_l = \max_j \sum_{i=1}^n |a_{ij}| = \left\| \left| \frac{\partial \varphi_1}{\partial x_1} \right| + \left| \frac{\partial \varphi_2}{\partial x_1} \right|, \left| \frac{\partial \varphi_1}{\partial x_2} \right| + \left| \frac{\partial \varphi_2}{\partial x_2} \right| \right\|_l = \|0.82, 0.25\|_l = 0.82$$

3. E – норма матриці:

$$\begin{aligned} \|\Phi'(x)\|_E &= \sqrt{\sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2} = \sqrt{\left| \frac{\partial \varphi_1}{\partial x_1} \right|^2 + \left| \frac{\partial \varphi_1}{\partial x_2} \right|^2 + \left| \frac{\partial \varphi_2}{\partial x_1} \right|^2 + \left| \frac{\partial \varphi_2}{\partial x_2} \right|^2} = \\ &= \sqrt{0.27 + 0.06 + 0.09} = \sqrt{0.42} = 0.65 \end{aligned}$$

Отже, ми одержали значення усіх норм матриці $\Phi'(x)$ меншими за одиницю. Тому система нелінійних рівнянь (3.78) задовольняє умові збіжності методу простих ітерацій: $\|\Phi'(x)\| < 1$.

3.13.3. Метод Зейделя

Метод Зейделя для систем нелінійних рівнянь, як і для системи лінійних рівнянь, полягає у використуванні уточнених значень змінних уже на поточному ітераційному кроці [67]. Так, для уточнення на $(k+1)$ -му кроці значення першої змінної $x_1^{(k+1)}$ використовуємо усі значення попереднього k -го кроку, для другої змінної $x_2^{(k+1)}$ – значення $x_1^{(k+1)}$ $(k+1)$ -го кроку та значення решти змінних – з попереднього k -го кроку:

$$\begin{cases} x_1^{(k+1)} = \varphi_1(x_1^{(k)}, x_2^{(k)}, x_3^{(k)}, \dots, x_{n-1}^{(k)}, x_n^{(k)}), \\ x_2^{(k+1)} = \varphi_2(x_1^{(k+1)}, x_2^{(k)}, x_3^{(k)}, \dots, x_{n-1}^{(k)}, x_n^{(k)}), \\ x_3^{(k+1)} = \varphi_3(x_1^{(k+1)}, x_2^{(k+1)}, x_3^{(k)}, \dots, x_{n-1}^{(k)}, x_n^{(k)}), \\ \dots \\ x_n^{(k+1)} = \varphi_n(x_1^{(k+1)}, x_2^{(k+1)}, x_3^{(k+1)}, \dots, x_{n-1}^{(k+1)}, x_n^{(k)}). \end{cases} \quad (3.79)$$

де k – номер кроку ітерації.

Умова завершення процесу розв'язання системи нелінійних рівнянь за методом Зейделя збігається з умовою (3.77) для методу простих ітерацій (3.74).

Приклад 3.49. Розв'яжемо методом Зейделя систему нелінійних рівнянь (3.71) з похибкою 0.0001.

1. Використовуватимемо цю систему, як і для методу ітерацій, у перетвореному вигляді (3.78):

$$\begin{cases} x_1 = \sqrt{\frac{x_1(x_2 + 5) - 1}{2}}, \\ x_2 = \sqrt{x_1 + 3 \lg x_1}, \end{cases}$$

2. Оберемо вектор початкового наближення: $x^{(0)} = \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} = \begin{pmatrix} 3.5 \\ 2.2 \end{pmatrix}$.

Ці значення візьмемо, як і в попередньому випадку, з графіка рівнянь, який показаний на рис. 3.26.

3. Обчислимо перше наближення:

$$\begin{aligned} x_1^{(1)} &= \sqrt{\frac{x_1^{(0)}(x_2^{(0)} + 5) - 1}{2}} = \sqrt{\frac{3.5(2.2 + 5) - 1}{2}} = 3.478505, \\ x_2^{(1)} &= \sqrt{x_1^{(1)} + 3 \lg x_1^{(1)}} = \sqrt{3.478505 + 3 \lg 3.478505} = 2.258912. \end{aligned}$$

Подальші наближення обчислимо за поступового використання формули (3.79) розв'язання системи на $(k+1)$ -му кроці ітераційної процедури:

$$\begin{cases} x_1^{(k+1)} = \sqrt{\frac{x_1^{(k)}(x_2^{(k)} + 5) - 1}{2}}, \\ x_2^{(k+1)} = \sqrt{x_1^{(k+1)} + 3 \lg x_1^{(k+1)}}, \end{cases} \quad (3.80)$$

Відповідні значення поступових наближень розв'язків системи (3.80) наведено в таблиці 3.18.

Таблиця 3.18

Проміжні обчислення методом Зейделя

k – номер ітерації	$x_1^{(k)}$	$x_2^{(k)}$
0	3.5	2.2
1	3.478505	2.258912
2	3.482109	2.260008
3	3.484260	2.260662
4	3.485544	2.261052
5	3.486310	2.261284
6	3.486767	2.261423
7	3.487039	2.261506
8	3.487169	2.261528
9	3.487299	2.261585

Обчислимо похибки вектора наближень рівнянь за формулою (3.77):
після 2-го кроку ітерації:

$$\frac{\|x^{(2)} - x^{(1)}\|_2}{\|x^{(1)}\|_2} = \frac{\sqrt{(3.482109 - 3.478505)^2 + (2.260008 - 2.258912)^2}}{\sqrt{3.478505^2 + 2.258912^2}} = 0.0762933 > 0.0001$$

після 9-го кроку ітерації

$$\frac{\|x^{(9)} - x^{(8)}\|}{\|x^{(8)}\|} = \frac{0.0001014}{4.156346} = 0.0000244 < 0.0001;$$

Отже, наближення, обчислені на 2-му кроці ітерації, можна вважати за розв'язки системи (3.70) з похибкою 0.001; наближення, обчислені на 9-му кроці ітерації, за розв'язки системи з похибкою 0.0001, тобто

$$x_1 = 3.487299 \quad x_2 = 2.261585$$

Зауважимо, що умови збіжності розв'язків системи нелінійних рівнянь для методу Зейделя є такі самі, як і для методу простих ітерацій, а саме:

$$\|\Phi'(x)\| \leq 1 .$$

3.13.4. Метод Ньютона

Математичним підґрунтям методу є лінеаризація функцій f_1, f_2, \dots, f_n шляхом розкладання в ряд Тейлора в околі точки початкового наближення до розв'язку системи рівнянь й нехтування всіма членами ряду, окрім лінійних щодо приростів змінних.

Для однієї змінної ряд Тейлора в околі певної точки $x=x_0$ виглядає так:

$$f(x) = f(x_0) + \frac{1}{1!}(x-x_0)f'(x_0) + \frac{1}{2!}(x-x_0)^2 f''(x_0) + \dots + \frac{1}{n!}(x-x_0)^n f^{(n)}(x_0)$$

Для функцій f_1, f_2, \dots, f_n системи рівнянь (1) візьмемо лише лінійну частину (до другої похідної) розкладання в ряд Тейлора в околі точки $x^{(0)} = \{x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}\}$:

$$f_i(x_1, x_2, \dots, x_n) = f_i(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}) + (x_1 - x_1^{(0)}) \frac{\partial}{\partial x_1} f_i(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}) + \\ + (x_2 - x_2^{(0)}) \frac{\partial}{\partial x_2} f_i(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}) + \dots + (x_n - x_n^{(0)}) \frac{\partial}{\partial x_n} f_i(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}),$$

де $i = 1, 2, \dots, n$.

Введемо позначення для змінних:

$$\Delta x_i^{(0)} = (x_i - x_i^{(0)}) - \text{приріст } i - \text{ї змінної,}$$

f_i – значення i – ї функції.

$$F_{ij} = \frac{\partial}{\partial x_j} f_i(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}) - \text{значення першої частинної похідної функції}$$

f_i по змінній x_j

Після перетворення дістанемо систему лінійних рівнянь n -го порядку щодо приросту змінних Δx_j :

$$\begin{cases} F_{11}\Delta x_1 + F_{12}\Delta x_2 + \dots + F_{1n}\Delta x_n = -f_1, \\ F_{21}\Delta x_1 + F_{22}\Delta x_2 + \dots + F_{2n}\Delta x_n = -f_2, \\ \dots \\ F_{n1}\Delta x_1 + F_{n2}\Delta x_2 + \dots + F_{nn}\Delta x_n = -f_n, \end{cases} \quad (3.81)$$

або у матричній формі:

$$\begin{pmatrix} F_{11} & F_{12} & \dots & F_{1n} \\ F_{21} & F_{22} & \dots & F_{2n} \\ \dots & \dots & \dots & \dots \\ F_{n1} & F_{n2} & \dots & F_{nn} \end{pmatrix} \cdot \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \\ \dots \\ \Delta x_n \end{pmatrix} = \begin{pmatrix} -f_1 \\ -f_2 \\ \dots \\ -f_n \end{pmatrix} \quad (3.82)$$

У скороченому вигляді можна записати:

$$F(\Delta x) = -f,$$

де матрицю значень частинних похідних (F') називають матрицею Якобі, чи якобіаном системи рівнянь.

Розв'язок цієї системи (за умови $\det(F') \neq 0$) надає вектор відхилень до початкового наближення $\Delta x = -(F')^{-1}(f)$. Додавання його до вектора початкового наближення надає нові, уточнені значення змінних:

$$x^{(1)} = x^{(0)} + \Delta x^{(0)}$$

Продовжуючи ітераційний процес, дістанемо нові наближення розв'язків системи лінійних рівнянь за скороченою формулою:

$$x^{(k+1)} = x^{(k)} + \Delta x^{(k)} \quad (3.83)$$

або у загальному вигляді:

$$x^{(k+1)} = x^{(k)} - F^{-1}(x^{(k)}) \cdot f(x^{(k)}) \quad (3.84)$$

де $F^{-1}(x^{(k)})$ – обернена матриця Якобі F' для наближення

$$x^{(k)} = \{x_1^{\{k\}}, x_2^{\{k\}}, \dots, x_n^{\{k\}}\}, \quad k = 1, 2, \dots$$

Умова розв'язування системи нелінійних рівнянь за методом Ньютона збігається з методом простих ітерацій:

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

Отже, алгоритм розв'язування системи нелінійних рівнянь за методом Ньютона виглядає є таким:

1. Обираємо початкове наближення $x^{(0)} = \{x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}\}$;

2. Обчислюємо матрицю Якобі (F) – значення частинних похідних F'_{ij} для обраного наближення $x^{(k)}$ (k – номер кроку ітерації);

3. Розв'язуємо систему лінійних рівнянь (13) щодо приростів змінних

$$x^{(k+1)} = x^{(k)} - F^{-1}(x^{(k)}) \cdot f(x^{(k)})$$

4. До вектора наближення $x^{(k)}$ додаємо вектор приростів змінних (3.83) та дістаємо нове наближення $x^{(k+1)} = x^{(k)} + \Delta x^{(k)}$;

5. Перевіряємо умову завершення процесу розв'язування системи рівнянь:

$$\frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)}\|} \leq \varepsilon, \text{ якщо умови не досягнуто, то значення } k$$

збільшуємо на одиницю і повторюємо процедуру з п. 2, інакше процес ітерації зупиняємо.

Частинні похідні, потрібні для розрахунку матриці Якобі, можна обчислити аналітично або ж, якщо це неможливо чи то важко, обчислити за формулами чисельного диференціювання [67].

Приклад 3.50. Розв'язати за методом Ньютона систему нелінійних рівнянь:

$$\begin{cases} f_1(x_1, x_2) \equiv 2x_1^2 - x_1x_2 - 5x_1 + 1 = 0, \\ f_2(x_1, x_2) \equiv x_1 + 3 \lg x_1 - x_2^2 = 0. \end{cases} \text{ з похибкою } 0.0001:$$

Розв'язок. Оберемо вектор початкового наближення $x^{(0)}$ – точку перетину функцій графіка (рис. 3.26) з додатними значеннями:

$$x_1^{(0)} = 3.5 \text{ і } x_2^{(0)} = 2.2$$

Підставимо значення у функції f_1 та f_2

$$f(x) = \begin{pmatrix} 2x_1^2 - x_1x_2 - 5x_1 + 1 \\ x_1 + 3\lg x_1 - x_2^2 \end{pmatrix} = \begin{pmatrix} 0.300000 \\ 0.292204 \end{pmatrix}$$

Складемо матрицю Якобі

$$F = \begin{pmatrix} 4x_1 - x_2 - 5 & -x_1 \\ 1 + \frac{3 \cdot 0.4343}{x_1} & -2x_2 \end{pmatrix}$$

Підставимо значення початкового наближення в цю матрицю і обчислимо її визначник:

$$F(x_0) = \begin{pmatrix} 6.8 & -3.5 \\ 1.37 & -4.4 \end{pmatrix}; \quad \det(F(x^{(0)})) = -25.12 \neq 0$$

Отже, матриця $F'(x^{(0)})$ – невироджена. Обчислимо її обернену матрицю

$$F^{-1}(x^{(0)}) = \begin{pmatrix} 0.1751 & -0.1393 \\ 0.0546 & -0.2707 \end{pmatrix}$$

За формулою (3.84) обчислимо перше наближення

$$\begin{aligned} x^{(1)} &= x^{(0)} - F^{-1}(x^{(0)}) \cdot f(x^{(0)}) = \\ &= \begin{pmatrix} 3.5 \\ 2.2 \end{pmatrix} - \begin{pmatrix} 0.1751 & -0.1393 \\ 0.0546 & -0.2707 \end{pmatrix} \cdot \begin{pmatrix} 0.300000 \\ 0.292204 \end{pmatrix} = \begin{pmatrix} 0.3488164 \\ 0.2627187 \end{pmatrix} \end{aligned}$$

Аналогічно обчислюють наступні наближення. Результати обчислень наведено в таблиці 3.19.

Зупинимось на наближенні $x^{(3)}$, за якого значення функцій системи рівнянь є меншими за 10^{-12} , тобто розв'язок системи рівнянь (3.71) є

$$x_1 = 3.4874428; \quad x_2 = 2.2616286 \text{ (див. табл. 3.19).}$$

Таблиця 3.19

Результати проміжних обчислень системи рівнянь (3.72) методом Ньютона

k – номер кроку ітерації	$x_1^{(k)}$	$x_2^{(k)}$	$f_1(x^{(k)})$	$f_2(x^{(k)})$
0	3.5	2.2	0.3	0.2922041
1	3.488164	2.262718	0.001022	- 0.003941
2	3.487443	2.261629	$2.5404 \cdot 10^{-7}$	$-1.211 \cdot 10^{-6}$
3	3.4874428	2.2616286	$1.7763 \cdot 10^{-14}$	$8.899 \cdot 10^{-13}$
4	3.4874428	2.2616286	0	0

Обчислимо похибки вектора наближень рівнянь після 3-го кроку ітерації за формулою (3.77):

$$\frac{\|x^{(3)} - x^{(2)}\|}{\|x^{(2)}\|} = \frac{0.0000004}{4.156588} = 9.46603 \cdot 10^{-8} < 0.0001$$

Умови збіжності методу Ньютона для систем нелінійних рівнянь досліджували відомі вчені: Канторович, Островський, Віллерс, Стенін. Узагальнюючи їх дослідження, можна вважати за достатні умови збіжності розв’язків систем нелінійних рівнянь методу Ньютона такі:

1. Матриця Якобі для початкового наближення $F(x^{(0)})$ повинна мати обернену матрицю F^{-1} з нормою, меншою за певну величину A , тобто

$$\|F^{-1}(x^{(0)})\| \leq A;$$

2. Норма добутку оберненої матриці Якобі на вектор заданих функцій $f(x)$ повинна мати значення, менше за певну величину B :

$$\|F^{-1}(x^{(0)})f(x^{(0)})\| \leq B.$$

3. Сталі величини A , B та C мають задовольняти умову: $2nABC \leq 1$

4. Значення матриці Якобі для частинних похідних другого порядку мають задовольняти умові:

$$\sum_{k=1}^n \left| \frac{\partial^2 f_i(\bar{x})}{\partial x_j \partial x_k} \right| \leq C,$$

де $i, j = 1, 2, \dots, n$; \bar{x} – певні значення наближень розв’язків системи рівнянь в околі точки $x^{(0)}$.

Перевіримо виконання наведених умов для системи нелінійних рівнянь (3.72):

$$1. \text{ Матриця Якобі цієї системи: } F = \begin{pmatrix} 4x_1 - x_2 - 5 & -x_1 \\ 1 + \frac{3 \cdot 0.4343}{x_1} & -2x_2 \end{pmatrix}$$

має обернену матрицю для початкового наближення $x^{(0)}$:

$$F^{-1}(x^{(0)}) = \begin{pmatrix} 0.1751 & -0.1393 \\ 0.0546 & -0.2707 \end{pmatrix}$$

Норма цієї матриці $\|F^{-1}(x^{(0)})\| = 0.325366 < 0.33$, тобто величина $A = 0.33$.

2. Обчислимо добуток оберненої матриці Якобі на вектор заданих функцій $f(x)$ для початкового наближення $x^{(0)}$:

$$F^{-1}(x^{(0)})f(x^{(0)}) = \begin{pmatrix} 0.1751 & -0.1393 \\ 0.0546 & -0.2707 \end{pmatrix} \cdot \begin{pmatrix} 0.300000 \\ 0.292204 \end{pmatrix} = \begin{pmatrix} 0.011836 \\ -0.062718 \end{pmatrix}$$

Норма вектора добутку $\|F^{-1}(x^{(0)})f(x^{(0)})\| = 0.063826 < 0.1$, тобто величина $B = 0.1$.

3. Обчислимо значення матриці Якобі для частинних похідних другого порядку для початкового наближення $x^{(0)}$:

$$H = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -\frac{1.3029}{x_1^2} & 0 & 0 & -2 \end{pmatrix}, \quad H(x^{(0)}) = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -0.10636 & 0 & 0 & -2 \end{pmatrix}$$

З матриці $H(x^{(0)})$ обчислимо матрицю значень:

$$\sum_{k=1}^n \left| \frac{\partial^2 f_i(\bar{x})}{\partial x_j \partial x_k} \right|, \text{ де } i, j = 1, 2.$$

\bar{x} – значення наближень розв’язків системи рівнянь в точці $x^{(0)}$:

$$\begin{pmatrix} 5 & 1 \\ 0.10636 & 2 \end{pmatrix}$$

Усі елементи цієї матриці є менше чи дорівнюють 5, тобто величина $C = 5$;

4. Підставимо значення величин A, B, C та $n=2$ й перевіримо умову:

$$2nABC = 2 \cdot 2 \cdot 0.33 \cdot 0.1 \cdot 5 = 0.66 \leq 1$$

Отже, система нелінійних рівнянь (3.72) задовольняє умові збіжності методу Ньютона.

Контрольні завдання

1. Які методи розв’язування систем нелінійних рівнянь вам відомі?
 2. Запишіть формулу для визначення збіжності ітераційних методів розв’язання систем нелінійних рівнянь.

3. Знайти матрицю Якобі для даної системи нелінійних рівнянь:

$$\begin{cases} \sin x_1 + 2x_2 + x_3^3 = 18, \\ 2x_1 + x_2^2 + x_3 = 5. \end{cases}$$

4. Розв’язати систему нелінійних рівнянь:

$$\begin{cases} \sin(x - 0.75) - y = 1.5, \\ 2x - \cos(y) = 0.85. \end{cases}$$

5. Розв’язати систему нелінійних рівнянь:

$$\begin{cases} \sin(\pi x_1) - x_1 - x_2 \\ \cos(\pi x_2) - x_1 + x_2 \end{cases}$$

3.14. Методи чисельного інтегрування

3.14.1. Однокрокові методи

Прямий спосіб обчислення визначеного інтеграла

Прямий спосіб обчислення визначеного інтеграла полягає у використанні основної формули інтегрального числення

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

де $f(x)$ – неперервна на $[a, b]$ функція, $F(x)$ – її первісна.

Однак такі обчислення утруднені тим, що фактичне знаходження значень $F(x)$ можливо лише в незначній кількості випадків.

Із цієї причини велике значення мають формули для наближеного обчислення інтегралів [68].

3.14.2. Методи наближеного обчислення інтегралів

Методи Ньютона-Котеса. Методи засновані на апроксимації функції $f(x)$ поліномом степеня n . Алгоритми цього класу відрізняються тільки степенем полінома. Як правило, вузли апроксимуючого полінома – рівновіддалені.

Методи сплайн-інтегрування. Методи сплайн-інтегрування базуються на апроксимації функції $f(x)$ сплайном.

Методи Монте-Карло. Методи Монте-Карло використовують найчастіше при обчисленні кратних інтегралів, вузли вибирають випадковим чином, відповідь носить імовірнісний характер.

3.14.3. Методи Ньютона-Котеса. Постановка задачі

Нехай $y = f(x)$ – неперервна функція на відрізку $[a; b]$. Потрібно обчислити $\int_a^b f(x) dx$.

Розіб'ємо відрізок інтегрування на n рівних частин точками:
 $a = x_0 < x_1 < x_2 < \dots < x_n = b$, так, що $x_{i+1} - x_i = \frac{b-a}{n} = h$, $i = 0, 1, 2, \dots, n-1$ і
 нехай $y_i = f(x_i)$ – значення $f(x)$ в точках розбиття.

3.14.4. Метод прямокутників

На відрізку $[x_i; x_{i+1}]$ беремо довільну точку c . Інтерполяційним многочленом у випадку одного вузла є $f(c)$ [69].

$$\int_{x_i}^{x_{i+1}} f(x) dx = h \cdot f(c), \quad h = x_{i+1} - x_i. \quad (3.85)$$

Геометрична інтерпретація формули (3.85) представлена трьома можливими варіантами, показаними на рис. 3.27.

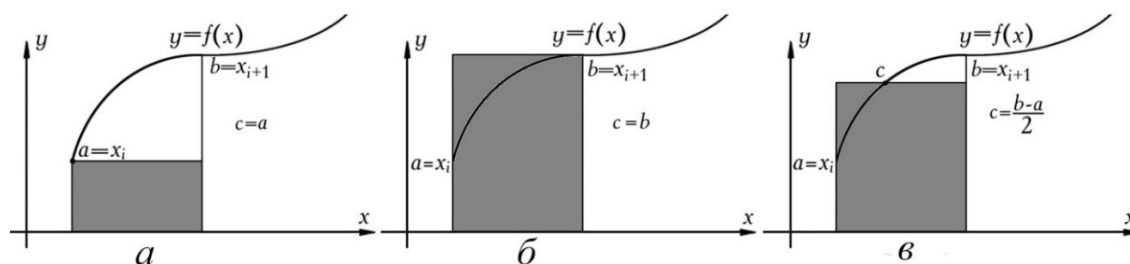


Рис. 3.27. Наближення прямокутниками: a – наближення лівими прямокутниками; $б$ – наближення правими прямокутниками; $в$ – наближення середніми прямокутниками.

Загальний вигляд формули прямокутників

$$\int_a^b f(x) dx = h \sum_{k=0}^{n-1} f(c + kh). \quad (3.86)$$

1) Формула лівих прямокутників: $c = x_0$.

2) Формула правих прямокутників: $c = x_1$.

3) Формула середніх прямокутників: $c = x_0 + \frac{h}{2}$.

Чисельне інтегрування методом прямокутників

Обчислення наближеного інтеграла від функції $y = f(x)$ на відрізку $[a; b]$.

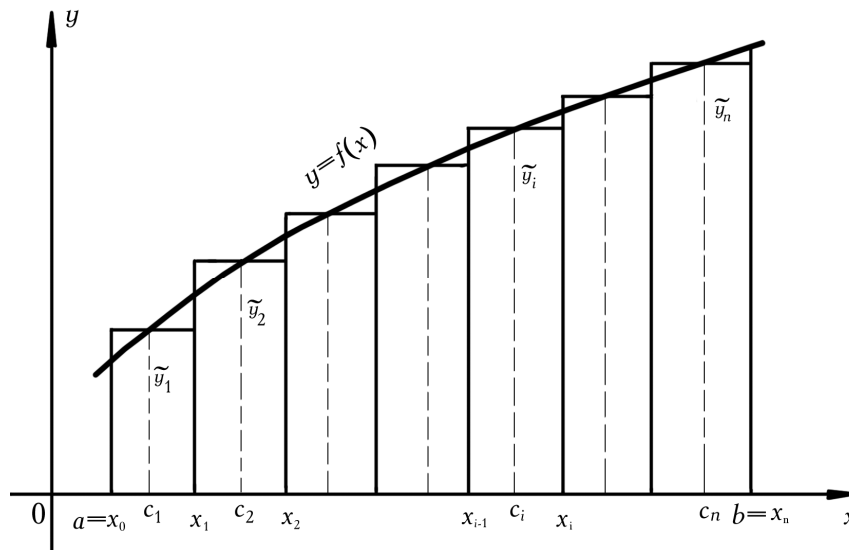


Рис. 3.28. Графічне визначення інтеграла через суму площ прямокутників при інтегруванні методом прямокутників

Алгоритм обчислень методом прямокутників

1. Поділимо відрізок $[a, b]$ на n частин

$a = x_0 < x_1 < x_2 < \dots < x_n = b$. Тоді крок дискретизації

$$h = \frac{b - a}{n} = x_{i+1} - x_i.$$

2. На кожному відрізку $[x_i, x_{i+1}]$ вибираємо точку $c_i = \frac{x_i + x_{i+1}}{2}$.

3. Приймавши ординату $\tilde{y}_i = f(c_i)$ за висоту, будемо прямокутники площею $s_i = h\tilde{y}_i$.

4. Тоді $\sum_{i=1}^n s_i$ дає площу східчастої фігури, що є наближеним

значенням шуканого визначеного інтеграла.

Загальна формула обчислень методом прямокутників:

$$\int_a^b f(x) dx \approx h(\tilde{y}_1 + \tilde{y}_2 + \dots + \tilde{y}_{i-1} + \tilde{y}_i + \dots + \tilde{y}_n) = \frac{b-a}{n} \sum_{i=1}^n f\left(\frac{x_{i-1} + x_i}{2}\right)$$

Цю формулу називають формулою середніх прямокутників.

Похибки методів прямокутників

1. Ліві прямокутники: $R_n(f, a) = \frac{(b-a)^2}{2n} f'(x^*)$, $x^* \in [a; b]$,
 2. Праві прямокутники: $R_n(f, a+h) = -\frac{(b-a)^2}{2n} f'(x^*)$, $x^* \in [a; b]$,
 3. Середні прямокутники: $R_n\left(f, a + \frac{h}{2}\right) = -\frac{(b-a)^2}{24n^2} f''(x^*)$, $x^* \in [a; b]$,
- $$f'(x^*) = \max f'(x), f''(x^*) = \max f''(x), x, x^* \in [a; b].$$

3.14.5. Метод трапецій. Постановка задачі

Нехай $y = f(x)$ – неперервна функція на відрізку $[a; b]$. Потрібно обчислити $\int_a^b f(x) dx$. Розіб'ємо відрізок інтегрування на n рівних частин

точками $a = x_0 < x_1 < x_2 < \dots < x_n = b$ так, що $x_{i+1} - x_i = \frac{b-a}{n} = h$, $i = 0, 1, 2, \dots, n-1$ і нехай $y_i = f(x_i)$ – значення $f(x)$ в точках ділення.

Чисельне інтегрування методом трапецій [65]

На рисунку показаний спосіб обчислення наближеного інтеграла від функції $y = f(x)$ на відрізку $[a; b]$ методом трапецій.

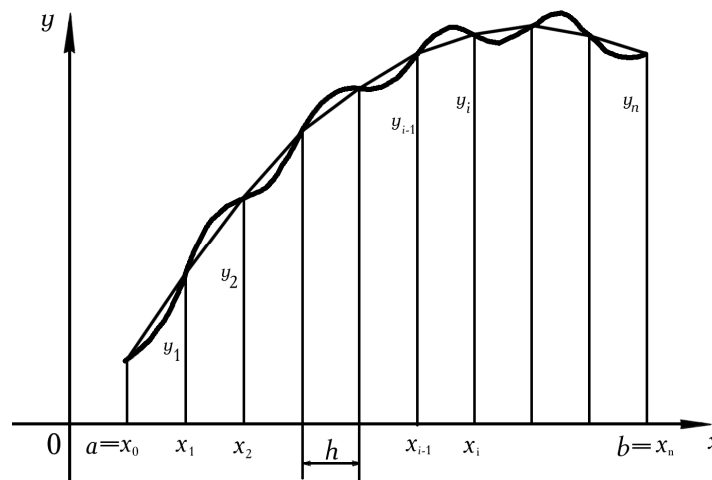


Рис. 3.29. Графічне визначення інтеграла через суму площ трапецій при інтегруванні методом трапецій

$$\begin{aligned}
h &= x_{i+1} - x_i, \\
h &= \frac{b-a}{n}, \\
i &= 0, 1, 2, \dots, n-1 \\
y_0 &= f(x_0), \\
y_1 &= f(x_1), \\
&\dots\dots\dots \\
y_i &= f(x_i), \\
&\dots\dots\dots \\
y_n &= f(x_n)
\end{aligned}$$

Формула методу трапецій

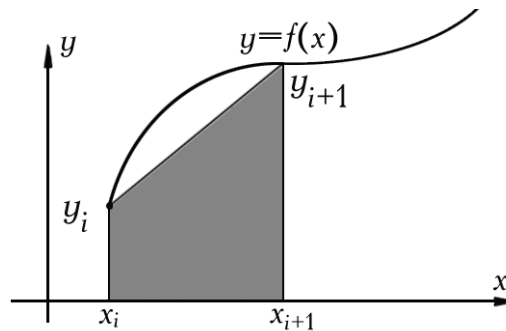


Рис. 3.29. Наближення трапецією

На відрізку $[x_i; x_{i+1}]$ утворюємо трапецію, обмежену віссю абсцис, ординатами y_i та y_{i+1} й відрізком, що з'єднує точки (x_i, y_i) і (x_{i+1}, y_{i+1}) .

Визначимо площу цієї трапеції як наближене значення інтеграла від функції $f(x)$ на відрізку $[x_i; x_{i+1}]$.

$$\int_{x_i}^{x_{i+1}} f(x) dx \approx h \frac{y_i + y_{i+1}}{2}, \quad h = x_{i+1} - x_i.$$

У випадку рівновіддалених вузлів: $h = \frac{b-a}{n}$, $x_i = a + h \cdot i$, $y_i = f(x_i)$,

$i = 0, 1, 2, \dots, n-1$.

Замінімо криву $y = f(x)$ ламаною лінією, ланки якої з'єднують кінці ординат y_i і y_{i+1} ($i = 0, 1, 2, \dots, n-1$).

Тоді одержимо множину трапецій з основами y_i, y_{i+1} і висотою $h = \frac{b-a}{n}$. Загальна формула трапецій:

$$\int_a^b f(x) dx \approx \frac{y_0 + y_1}{2} h + \frac{y_1 + y_2}{2} h + \dots + \frac{y_{n-1} + y_n}{2} h =$$

$$= \frac{b-a}{n} \left(\frac{y_0 + y_n}{2} + y_1 + y_2 + \dots + y_{n-1} \right).$$

Абсолютну похибку R_n наближення, отриманого за формулою трапецій, оцінюють за допомогою формули

$$|R_n| \leq \frac{(b-a)^3}{12n^2} M_2, \text{ де } M_2 = \max_{a \leq x \leq b} |f''(x)|$$

3.14.6. Формула парабол (Симпсона)

Замінімо графік функції $y = f(x)$ на кожному відрізку $[x_{i-1}; x_i]$

дугами парабол: $\int_a^b f(x) dx$.

Знайдемо площу S криволінійної трапеції, обмеженої зверху графіком параболи $y = ax^2 + bx + c$, збоку – ординатами y_0, y_1 і знизу – відрізком $[-h, h]$.

Нехай парабола проходить через три точки:

$m_0(-h, y_0), m_1(0, y_1), m_2(h, y_2)$, де $y_0 = ah^2 - bh + c$ – ордината параболи в точці $x = -h$;

$y_1 = c$ – ордината параболи в точці $x = 0$;

$y_2 = ah^2 + bh + c$ – ордината параболи в точці $x = h$.

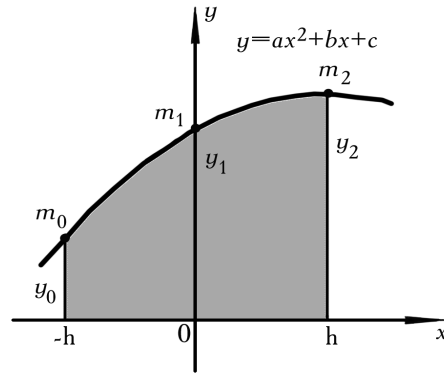


Рис. 3.30. Наближення параболою

Площа S дорівнює

$$S = \int_{-h}^h (ax^2 + bx + c) dx = \left(a \frac{x^3}{3} + b \frac{x^2}{2} + cx \right) \Big|_{-h}^h =$$

$$= \left(\frac{ah^3}{3} + \frac{bh^2}{2} + ch \right) - \left(-\frac{ah^3}{3} + \frac{bh^2}{2} - ch \right) = 2\frac{ah^3}{3} + 2ch.$$

Виразимо цю площу через h, y_0, y_1, y_2 :

$$1) y_0 = ah^2 - bh + c, \quad 2) y_1 = c, \quad 3) y_2 = ah^2 + bh + c.$$

Складемо перший та третій вираз й зведемо подібні члени:

$$y_0 + y_2 = 2ah^2 + bh - bh + 2c = 2ah^2 + 2c$$

$$\text{Підставимо 2) в отриманий вираз: } y_0 + y_2 = 2ah^2 + 2y_1.$$

З отриманого співвідношення виразимо значення a :

$$a = \frac{1}{2h^2}(y_0 - 2y_1 + y_2).$$

Підставляючи значення c й a в отриманий вираз для площі:

$$S = \frac{2ah^3}{3} + 2ch = \frac{2h^3}{3} \cdot \frac{1}{2h^2}(y_0 - 2y_1 + y_2) + 2hy_1 =$$

$$= \frac{h}{3}(y_0 - 2y_1 + y_2) + 2hy_1 = \frac{h}{3}(y_0 + 4y_1 + y_2).$$

Розглянемо відрізок $[a; b]$. Розіб'ємо його на $2n$ рівних частин (відрізків)

довжиною $h = \frac{a-b}{2n}$. Заміняємо кожен парусусідніх елементарних

криволінійних трапецій з основами, рівними h , однією елементарною параболічною трапецією з основою, що дорівнює $2h$. На відрізку $[x_0; x_2]$ парабола проходить через *три точки* $m_0(x_0, y_0), m_1(x_1, y_1), m_2(x_2, y_2)$.

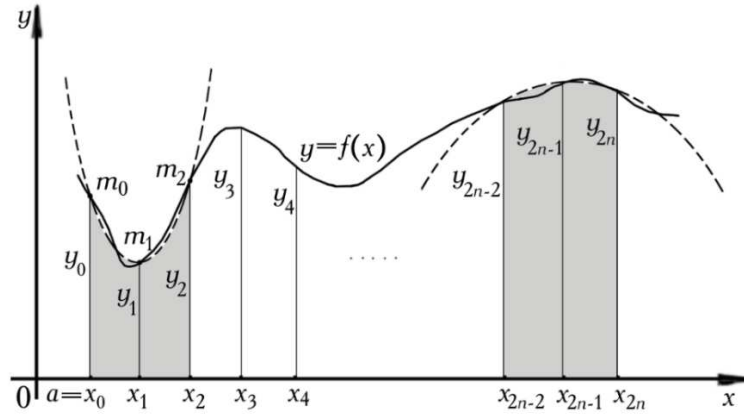


Рис. 3.31. Графічне визначення інтеграла через суму площ під параболою при інтегруванні методом парабол

Використовуючи формулу $S = \frac{h}{3}(y_0 + 4y_1 + y_2)$, знаходимо

$$S_1 = \int_{x_0}^{x_2} f(x) dx = \frac{h}{3}(y_0 + 4y_1 + y_2).$$

Аналогічно знаходимо:

$$S_2 = \int_{x_2}^{x_4} f(x) dx = \frac{h}{3}(y_2 + 4y_3 + y_4),$$

...

$$S_n = \int_{x_{2n-2}}^{x_{2n}} f(x) dx = \frac{h}{3}(y_{2n-2} + 4y_{2n-1} + y_{2n}).$$

Склавши отримані рівності, одержимо:

$$\int_a^b f(x) dx = \frac{h}{3}(y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + \dots + 2y_{2n-2} + 4y_{2n-1} + y_{2n})$$

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

$$= \frac{b-a}{6n}((y_0 + y_{2n}) + 4(y_1 + y_3 + \dots + y_{2n-1}) + 2(y_2 + y_4 + \dots + y_{2n-2}))$$

Формулу називають формулою парабол або формулою Симпсона [70].

Абсолютну похибку обчислень за формулою Симпсона оцінюють співвідношенням:

$$|R_n| \leq \frac{(b-a)^5}{180 \cdot (2n)^4} \cdot M_4, \text{ де } M_4 = \max_{a \leq x \leq b} |f^{IV}(x)|.$$

Слід зазначити, що формула дає точне значення інтеграла $\int_a^b f(x) dx$

у всіх випадках, коли $f(x)$ – многочлен, степінь якого менше або дорівнює трьом [70].

3.14.7. Сплайн-квадратура

Нехай потрібно обчислити $I = \int_a^b f(x) dx$.

Розіб'ємо $[a; b]$ на відрізки довжини

$$h_j = x_j - x_{j-1}, j = 1, 2, \dots, n, x_0 = a, x_n = b.$$

На кожному з відрізків замінимо підінтегральну функцію кубічним сплайном $\varphi_j(x)$.

$$\varphi_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3, x \in [x_{j-1}, x_j].$$

Тоді початкова задача набуде вигляду:

$$I = \int_a^b f(x) dx = \sum_{j=1}^n \int_{x_{j-1}}^{x_j} \varphi_j(x) dx.$$

Обчислимо інтеграл від функції φ_j на довільному відрізку $[x_{j-1}; x_j]$:

$$\int_{x_{j-1}}^{x_j} \varphi_j(x) dx = a_j h_j + \frac{b_j}{2} h_j^2 + \frac{c_j}{3} h_j^3 + \frac{d_j}{4} h_j^4.$$

Початкова задача набуде вигляду:

$$I = \sum_{j=1}^n \int_{x_{j-1}}^{x_j} \varphi_j(x) dx = \sum_{j=1}^n \left(a_j h_j + \frac{b_j}{2} h_j^2 + \frac{c_j}{3} h_j^3 + \frac{d_j}{4} h_j^4 \right).$$

Коефіцієнти сплайнів a_j, b_j, d_j виражають через коефіцієнти c_j та значення підінтегральної функції $f_i(x)$:

$$\begin{aligned} a_j &= f_{j-1}, \\ b_j &= \frac{f_i - f_{j-1}}{h_j} - \frac{(c_{j+1} + 2c_j)h_j}{3}, \\ d_j &= \frac{c_{j+1} - c_j}{3h_j} \end{aligned} \quad (3.87)$$

Значення коефіцієнтів c_i можна одержати, розв'язавши систему лінійних алгебраїчних рівнянь [65].

$$h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} = 3 \left(\frac{f_i - f_{i-1}}{h_i} - \frac{f_{i-1} - f_{i-2}}{h_{i-1}} \right), \quad 2 \leq i \leq n.$$

Матриця для розв'язування системи – це трьохдіагональна матриця. Тому для одержання розв'язку доцільно використовувати метод прогонки.

Коефіцієнти c_1 та c_{n+1} отримані з умов вільних кінців сплайна. Зазвичай вимагають нульової кривизни на кінцях сплайна. Тому беруть

$$c_1 = c_{n+1} = 0. \text{ Підставимо у вираз } I = \sum_{j=1}^n \left(a_j h_j + \frac{b_j}{2} h_j^2 + \frac{c_j}{3} h_j^3 + \frac{d_j}{4} h_j^4 \right)$$

значення коефіцієнтів:

$$\begin{aligned} a_j &= f_{j-1}, \quad b_j = \frac{f_j - f_{j-1}}{h_j} - \frac{(c_{j+1} + 2c_j)h_j}{3}, \quad d_j = \frac{c_{j+1} - c_j}{3h_j} \\ I &= \sum_{j=1}^n \left(f_{j-1} h_j + \frac{f_j - f_{j-1}}{2} h_j - \frac{c_{j+1} + 2c_j}{6} h_j^3 + \frac{c_j}{3} h_j^3 + \frac{c_{j+1} - c_j}{12} h_j^3 \right) \\ I &= \sum_{j=1}^n \frac{f_j + f_{j-1}}{2} h_j - \sum_{j=1}^n \frac{h_j^3 (c_{j+1} + c_j)}{12}. \end{aligned}$$

Контрольні завдання

1. Обчислити приблизно інтеграл $\int_0^1 (3x^2 + x + 2) dx$ використовуючи середні прямокутники в методі Ньютона-Котеса за умови, що $h = 0.2$.
2. Обчислити приблизно інтеграл $\int_0^1 (x^2 + 2x + 1) dx$ використовуючи метод трапецій при $h = 0.2$.
3. Обчислити приблизно інтеграл $\int_0^1 (x^2 - 2x + 3) dx$ використовуючи ліві прямокутники в методі Ньютона-Котеса за умови, що $h = 0.2$.
4. Обчислити інтеграл $\int_1^2 \left(\frac{x^4}{6} + \frac{x}{3} - \frac{1}{40} \right) dx$ методом трапецій з точністю 0.01.
5. Обчислити інтеграл $\int_0^5 \frac{x}{x^4 + 8} dx$ методом парабол за умови поділу відрізка на 5 частин.

3.15. Чисельний розв'язок диференціальних рівнянь

3.15.1. Загальні положення

Будемо розглядати звичайні диференціальні рівняння порядку n , що мають загальний вигляд:

$$F(x, y, y', \dots, y^{(n-1)}, y^{(n)}) = 0. \quad (3.88)$$

Визначення. Порядком диференціального рівняння називають порядок його старшої похідної.

Розв'язати диференціальне рівняння – означає знайти невідому функцію $y(x)$, яка перетворює це рівняння у вірну тотожність.

3.15.2. Частковий розв'язок диференціального рівняння

Визначення. Частковим розв'язком рівняння

$$F(x, y, y', \dots, y^{(n-1)}, y^{(n)}) = 0.$$

на інтервалі (a, b) називають будь-яку n раз диференційовну функцію $y = \varphi(x)$, що задовольняє цьому рівнянню, тобто таку, що перетворює рівняння на цьому інтервалі в тотожність [71].

Процедуру розв'язування диференціального рівняння часто називають інтегруванням рівняння, при цьому інтегрувати доводиться в загальному випадку рівно n раз, і при кожному інтегруванні в розв'язок входить чергова довільна постійна.

3.15.3. Загальний розв'язок диференціального рівняння

Визначення. Загальним розв'язком (загальним інтегралом) рівняння $F(x, y, y', \dots, y^{(n-1)}, y^{(n)}) = 0$ називають таке співвідношення

$$\Phi(x, y, C_1, \dots, C_n) = 0$$

з якого може бути отриманий частковий розв'язок

$$y = \varphi(x, y, C_1, C_2, \dots, C_n), \text{ де } C_1 = \text{const}, \dots, C_n = \text{const}.$$

Будемо розглядати диференціальні рівняння, виражені відносно старшої похідної:

$$y^{(n)} = f(x, y, y', y'', \dots, y^{(n-1)})$$

і одержувати загальний розв'язок, виражений відносно невідомої функції

$$y = \varphi(x, y, C_1, C_2, \dots, C_n).$$

Існують два роди задач, пов'язаних з визначенням розв'язків рівняння:

$$y^{(n)} = f(x, y, y', \dots, y^{(n-1)}). \quad (3.89)$$

1. Задача Коші, або задача з початковими даними.
2. Крайова задача або задача з граничними умовами.

Задача Коші

Знайти той розв'язок $y(x)$ диференціального рівняння, який при $x = x_0$ задовольняє умовам:

$$y(x_0) = y_0, y'(x_0) = y'_0, \dots, y^{(n-1)}(x_0) = y_0^{(n-1)}. \quad (3.90)$$

Характерним для задачі Коші є те, що умови (3.90) задають в одній точці $x = x_0$.

3.15.4. Геометрична інтерпретація розв'язків диференціальних рівнянь першого порядку

Нехай дано рівняння першого порядку: $F(x, y, y') = 0$.

Один з розв'язків даного рівняння $S = \varphi(x, c)$ називають інтегральною кривою рівняння. Кожна її точка відповідає значенням x , y і $\frac{dy}{dx}$ у вигляді кута нахилу дотичної, які перетворюють $F(x, y, y') = 0$ у тотожність. Множина дотичних утворює *поле напрямків*.

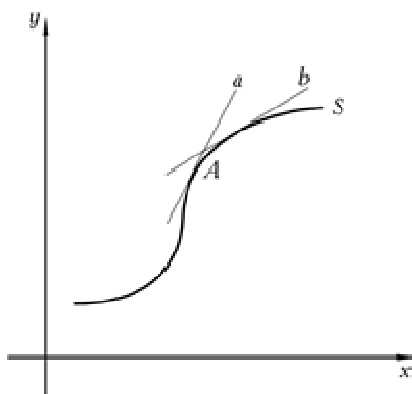


Рис. 3.31. Інтегральна крива S

Розглянемо сімейство ліній Γ_c , задане рівнянням:

$$F(x, y, c) = 0, \text{ де } c - \text{ параметр.}$$

Крива сімейства Γ_c – це геометричне місце точок (ГМТ), де всі інтегральні криві мають однаковий кут y' нахилу дотичної до осі x .

Визначення. Криві $\Gamma_{c_1}, \Gamma_{c_2}, \dots, \Gamma_{c_n}$, які відповідають послідовності параметрів c_1, c_2, \dots, c_n й характеризуються однаковим кутом $\frac{dy}{dx}$ нахилу дотичної для кожного параметра, називають ізоклінами.

Нехай задана початкова умова $y_1 = \varphi(x_1)$.

Тоді

$$F(x_1, y_1, c_1) = 0,$$

$$F(x_2, y_2, c_2) = 0, \dots,$$

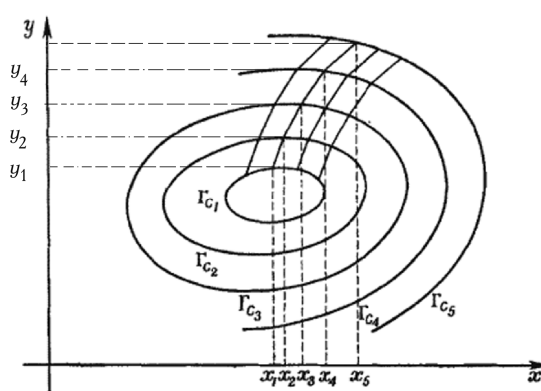
$$F(x_n, y_n, c_n) = 0$$


Рис. 3.32. Графік ізоклін

Інтегральна крива проходить через точки $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ і представлена кусочно-лінійними відрізками.

3.15.5. Чисельні методи розв'язування диференціальних рівнянь

Методи розв'язування диференціальних рівнянь:

1. Методи точного інтегрування.

Дозволяють знайти розв'язок у вигляді аналітичної функції.

Недолік. Застосовні для дуже обмеженого класу рівнянь.

2. Методи чисельного інтегрування.

Представляють розв'язок диференціального рівняння у вигляді таблиць значень шуканої функції залежно від значення змінної.

3.15.6. Однокрокові методи. Метод Ейлера

Нехай дана задача Коші для рівняння першого порядку:

$$\frac{dy}{dx} = f(x, y) \text{ при } y|_{x=x_0} = y_0.$$

Функція f задана в деякій області $D \subset R^2$

Розв'язок будемо шукати на відрізку $[a; b]$.

На відрізку $[a; b]$ введемо вузли: $a = x_0 < x_1 < \dots < x_n = b$

Метод полягає у виконанні таких дій:

1. Підставимо початкову умову (x_0, y_0) у диференціальне рівняння

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

2. Замінімо на відрізку $[x_0, x_1]$ інтегральну криву на дотичну:

$$y_1 = y_0 + f(x_0, y_0)(x_1 - x_0)$$

3. Замінімо на відрізку $[x_1, x_2]$ інтегральну криву на дотичну:

$$y_2 = y_1 + f(x_1, y_1)(x_2 - x_1).$$

.....

n . Замінімо на відрізку $[x_{n-1}, x_n]$ інтегральну криву на дотичну:

$$y_n = y_{n-1} + f(x_{n-1}, y_{n-1})(x_n - x_{n-1})$$

Продовжуючи подібні дії далі, одержуємо ламану криву, яку називають ламаною Ейлера (рис. 3.32).

При $x_{i+1} - x_i = h$ одержуємо:

$$y_n = y_{n-1} + f(x_{n-1}, y_{n-1})h$$

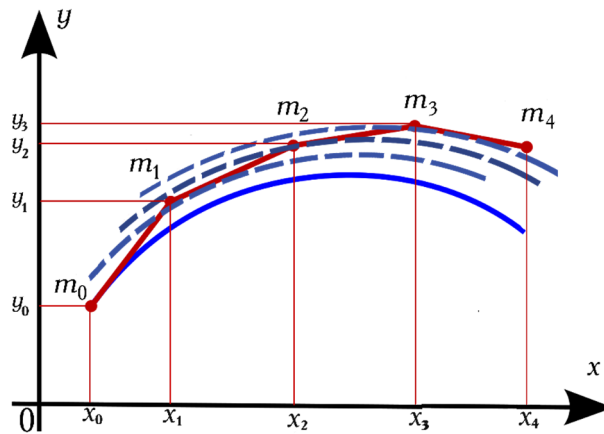


Рис. 3.32. Побудова ламаної Ейлера

На рисунку видно джерело похибки методу Ейлера. У випадку опуклої функції ця похибка накопичується.

Похибка методу Ейлера

Нехай точно задане початкове значення інтегральної кривої $y_0 = \varphi(x_0)$. Тоді в точці $x_0 + h$ метод Ейлера дає розв'язок:

$$y(x_0 + h) = y(x_0) + hf(x_0, y(x_0)).$$

Враховуючи, що $y' = f(x, y)$, $y(x_0 + h) = y(x_0) + hy'(x_0)$.

Розкладання y в ряд Тейлора в околі h буде мати вигляд:

$$y(x_0 + h) = y(x_0) + hy'(x_0) + \frac{1}{2}h^2y''(x_0) + O(h^3). \quad (3.91)$$

Похибку визначають як різницю між (3.90) і (3.91).

$$\frac{1}{2}h^2y''(x_0) + O(h^3)$$

Слід зазначити, що точність методу Ейлера відносно невисока. Підвищити точність, звичайно, можна, зменшивши крок обчислень, однак це призведе до ускладнення розрахунків [72].

Тому на практиці застосовують так званий уточнений метод Ейлера або формулу перерахування.

3.15.7. Уточнений метод Ейлера

Суть методу полягає в тому, що після обчислення нульового наближення $y_1^{(0)} = y_0 + f(x_0, y_0)h$ виконують уточнення шляхом обчислення середніх арифметичних значень $f(x_0, y_0)$ і $f(x_1, y_1^{(0)})$ [73]. Тоді уточнене значення:

$$y_1^{(1)} = y_0 + \frac{f(x_0, y_0) + f(x_1, y_1^{(0)})}{2} h.$$

Тоді уточнене значення: $y_1^{(1)} = y_0 + \frac{f(x_0, y_0) + f(x_1, y_1)}{2} h.$

Потім знаходимо значення похідної в точці $(x_1, y_1^{(1)})$. Заміняючи $f(x_0, y_0)$ середнім арифметичним значень $f(x_0, y_0)$ та $f(x_1, y_1^{(1)})$, знаходимо друге уточнене значення y_1 .

$$y_1^{(2)} = y_0 + \frac{f(x_0, y_0) + f(x_1, y_1^{(1)})}{2} h.$$

Потім третє: $y_1^{(3)} = y_0 + \frac{f(x_0, y_0) + f(x_1, y_1^{(2)})}{2} h$

і т. д., поки два послідовні уточнені значення не співпадуть у межах заданого ступеня точності. Тоді це значення беруть за ординату точки m_1 ламаної Ейлера. Аналогічну операцію проводять для інших значень y .

3.15.8. Метод Рунге-Кутта

Метод Рунге-Кутта є більш точним у порівнянні з методом Ейлера.

Суть уточнення полягає в тому, що шуканий розв'язок представляють у вигляді розкладання в ряд Тейлора.

$$y_{i+1} = y_i + y_i' h + y_i'' \frac{h^2}{2!} + y_i''' \frac{h^3}{3!} + y_i^{IV} \frac{h^4}{4!} + \dots$$

Якщо в цій формулі обмежитися двома першими доданками, то одержимо формулу методу Ейлера. Метод Рунге-Кутта враховує чотири перші члени розкладання [74].

$$y_{i+1} = y_i + y_i' h + y_i'' \frac{h^2}{2!} + y_i''' \frac{h^3}{3!} = y_i + \Delta y_i$$

У методі Рунге-Кутта прирости Δy_i обчислюють за формулою:

$$\Delta y_i = \frac{1}{6} \left(k_1^{(i)} + 2k_2^{(i)} + 2k_3^{(i)} + k_4^{(i)} \right),$$

де коефіцієнти k_i обчислюють за формулами:

$$k_1^{(i)} = hf(x_i, y_i),$$

$$k_2^{(i)} = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_1^{(i)}}{2}\right),$$

$$k_3^{(i)} = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_2^{(i)}}{2}\right),$$

$$k_4^{(i)} = hf(x_i + h, y_i + k_3^{(i)}).$$

Приклад 3.51. Розв'язати методом Рунге-Кутта диференціальне рівняння $y' = x + y$ при початковій умові $y(0) = 1$ на відрізку $[0; 0.5]$ із кроком 0.1.

Розв'язок

Для $i = 0$ обчислимо коефіцієнти k_i .

$$k_1^{(0)} = hf(x_0, y_0) = 0.1(x_0 + y_0) = 0.1(0 + 1) = 0.1;$$

$$k_2^{(0)} = hf\left(x_0 + \frac{h}{2}; y_0 + \frac{k_1^{(0)}}{2}\right) = 0.1(0.05 + 1.05) = 0.11;$$

$$k_3^{(0)} = hf\left(x_0 + \frac{h}{2}; y_0 + \frac{k_2^{(0)}}{2}\right) = 0.1(0.05 + 1.055) = 0.1105;$$

$$k_4^{(0)} = hf(x_0 + h; y_0 + k_3^{(0)}) = 0.1(0.1 + 1.1105) = 0.1211.$$

$$\begin{aligned} \Delta y_0 &= \frac{1}{6}(k_1^{(0)} + 2k_2^{(0)} + 2k_3^{(0)} + k_4^{(0)}) = \\ &= \frac{1}{6}(0.1 + 0.22 + 0.221 + 0.1211) = 0.1104 \end{aligned}$$

$$x_1 = x_0 + h, \quad y_1 = y_0 + \Delta y_0 = 1 + 0.1104 = 1.1104.$$

Наступні обчислення приводити не будемо, а результати представимо у вигляді таблиці.

Таблиця 3.20

Результати проміжних обчислень методом Рунге-Кутта

i	x_i	k		Δy_i	y_i
0	0	1	0.1000	0.1104	1
		2	0.1100		
		3	0.1105		
		4	0.1155		
1	0.1	1	0.1210	0.1325	1.1104
		2	0.1321		
		3	0.1326		
		4	0.1443		
2 2	0.2	1	0.1443	0.1569	1.2429
		2	0.1565		
		3	0.1571		
		4	0.1700		
3	0.3	1	0.1700	0.1840	1.3998
		2	0.1835		
		3	0.1842		
		4	0.1984		
4	0.4	1	0.1984	0.2138	1.5838
		2	0.2133		
		3	0.2140		
		4	0.2298		
5	0.5				1.7976

Розв'яжемо це ж рівняння методом Ейлера

Застосовуємо формулу $y_n = y_{n-1} + hf(x_{n-1}, y_{n-1})$.

$$1) x_0 = 0, y_0 = 1.$$

$$f(x_0, y_0) = x_0 + y_0 = 1,$$

$$hf(x_0, y_0) = h(x_0 + y_0) = 0.1,$$

$$y_1 = y_0 + hf(x_0, y_0) = 1 + 0.1 = 1.1.$$

$$2) x_1 = 0.1, y_1 = 1.1.$$

$$f(x_1, y_1) = x_1 + y_1 = 1.2,$$

$$hf(x_1, y_1) = h(x_1 + y_1) = 0.12,$$

$$y_2 = y_1 + hf(x_1, y_1) = 1.1 + 0.12 = 1.22.$$

Виконуючи аналогічні обчислення далі, одержуємо таблицю значень.

Таблиця 3.21

Результати проміжних обчислень методом Ейлера

i	0	1	2	3	4	5
x_i	0.0	0.1	0.2	0.3	0.4	0.5
y_i	1	1.1	1.22	1.362	1.528	1.721

Застосуємо тепер уточнений метод Ейлера.

Таблиця 3.22

Результати проміжних обчислень уточненим методом Ейлера

i	0	1	2	3	4	5
x_i	0.0	0.1	0.2	0.3	0.4	0.5
y_i	1	0.1	0.243	1.400	1.585	1.799

3.15.9. Визначення однокрокових методів

Однокрокові методи – це методи, що дозволяють одержати наближення y_{n+1} до значення точного розв'язку $y(x_{n+1})$ для кожного вузла дискретизації x_{n+1} на основі відомого наближення y_n до точного значення $y(x_n)$ у вузлі x_n . Загальний вигляд явного однокрокового методу:

$$y_{n+1} = F(f, x_n, x_{n+1}, y_n)$$

Загальний вигляд неявного однокрокового методу:

$$y_{n+1} = F(f, x_n, x_{n+1}, y_n, y_{n+1})$$

Однокроковий метод має порядок точності s , якщо для достатньо гładких задач виконується нерівність

$$\|y(x_{n+1}) - y_{n+1}\| \leq Ch^{s+1}, \text{ де } C - \text{const.}$$

Явний метод Ейлера

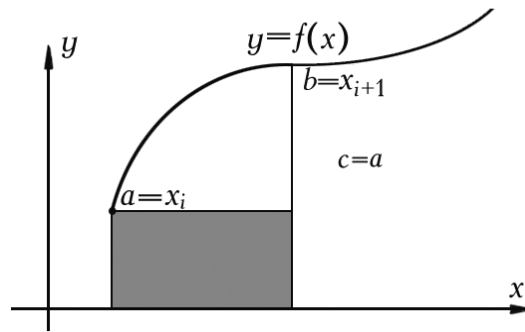


Рис. 3.33. Графічне представлення явного методу Ейлера

Нехай дана задача Коші для рівняння першого порядку:

$$\frac{dy}{dx} = f(x, y) \text{ при } y|_{x=x_0} = y_0 \rightarrow y(x) = y_0 + \int_{x_0}^x f(x, y) dx$$

На відрізку $[a; b]$ введемо вузли $a = x_0 < x_1 < \dots < x_n = b$ й застосуємо метод лівих прямокутників Ньютона-Котеса:

$$y_1 = y_0 + f(x_0, y_0)(x_1 - x_0)$$

$$y_2 = y_1 + f(x_1, y_1)(x_2 - x_1)$$

.....

$$y_n = y_{n-1} + f(x_{n-1}, y_{n-1})(x_n - x_{n-1})$$

На кожному кроці явно можна одержати значення y_i .

Неявний метод Ейлера

Нехай дана задача Коші для рівняння першого порядку:

$$\frac{dy}{dx} = f(x, y) \text{ при } y|_{x=x_0} = y_0 \rightarrow y(x) = y_0 + \int_{x_0}^x f(x, y) dx$$

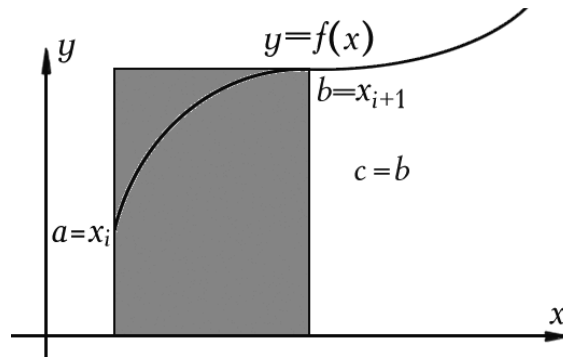


Рис. 3.34. Графічне представлення неявного методу Ейлера

На відрізку $[a; b]$ введемо вузли $a = x_0 < x_1 < \dots < x_n = b$ й застосуємо метод правих прямокутників Ньютона-Котеса

$$\begin{cases} y_1 = y_0 + f(x_1, y_1)(x_1 - x_0) \\ y_2 = y_1 + f(x_2, y_2)(x_2 - x_1) \\ \dots \\ y_n = y_{n-1} + f(x_n, y_n)(x_n - x_{n-1}) \end{cases}$$

Для одержання значень y_1, y_2, \dots, y_n необхідно розв'язати систему з n рівнянь із n невідомими.

3.15.10. Визначення багатокрокових методів

Багатокрокові методи – це методи, що дозволяють одержати наближення y_{n+k} до значення точного розв'язку $y(x_{n+k})$ для кожного вузла дискретизації x_{n+k} у загальному випадку на основі відомих наближень $y_{n+k-1}, y_{n+k-2}, \dots, y_n$ у вузлах $x_{n+k-1}, x_{n+k-2}, \dots, x_n$.

Загальний вигляд неявного k – крокового методу:

$$y_{n+k} = F(f, x_{n+k}, x_{n+k-1}, \dots, x_n, y_{n+k}, y_{n+k-1}, \dots, y_n)$$

Загальний вигляд явного k – крокового методу:

$$y_{n+k} = F(f, x_{n+k-1}, \dots, x_n, y_{n+k-1}, \dots, y_n)$$

Постановка задачі

Потрібно знайти наближений розв'язок диференціального рівняння $y' = f(x, y)$ з початковою умовою $y(x_0) = y_0$. Розв'язування задачі полягає в побудові наближених значень y_i розв'язку $y(x)$ у точках $x_i = x_0 + ih$.

Для обчислення значення y_{i+1} використовуються результати не одного, а k попередніх кроків, тобто значення $y_{i-k+1}, y_{i-k+2}, \dots, y_i$. У цьому випадку отримуємо k – кроковий метод.

Явний метод Адамса (метод Адамса-Башифорта)

Нехай відомі розв'язки у вузлах $x_i, i = 0, 1, \dots, n$. Тоді на відрізку $[n, n+1]$ можна записати розв'язок рівняння $y' = f(x, y)$ у вигляді

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(x, y) dx$$

Для чисельного обчислення даного інтеграла використовуємо представлення функції $f(x, y)$ інтерполяційним поліномом Ньютона з інтерполяцією назад:

$$N_{n-1}(x) = f(x_n) + f(x_{n-1}; x_n) \cdot (x - x_n) + \dots + f(x_1; \dots; x_n) \cdot (x - x_n) \cdot \dots \cdot (x - x_1)$$

Цей поліном припускає інтерполяцію функції при нерівновіддалених вузлах.

Представимо усічену формулу для m рівновіддалених вузлів:

$$f(x) = a_0 + a_1 \cdot (x - x_n) + \dots + a_k \cdot (x - x_n) \cdot \dots \cdot (x - x_{n-k+1}) + \dots + a_m \cdot (x - x_n) \cdot \dots \cdot (x - x_{n-m+1}) + R_m(x),$$

де $a_k = \frac{\Delta^k f_n}{k! h^k}$ – коефіцієнти полінома Ньютона,

$\Delta^k f_n$ – скінченні ліві різниці k – го порядку функції $f(x, y)$ у точці x_n .

У випадку постійного кроку скінченні різниці для правої частини у вузлі мають вигляд:

$$\Delta f_i = f_i - f_{i-1},$$

$$\Delta^2 f_i = f_i - 2f_{i-1} + f_{i-2},$$

$$\Delta^3 f_i = f_i - 3f_{i-1} + 3f_{i-2} - f_{i-3}$$

$R_m(x)$ – похибка інтерполяції.

Підставимо даний поліном замість підінтегральної функції у виразі

$$y_{n+1} = y_n + \int_{x_n}^{x_{n+1}} f(x, y) dx$$

Нехтуючи помилкою інтерполяції, одержимо:

$$y_{n+1} \approx y_n + \sum_{k=0}^m \frac{\Delta^k f_n}{k! h^k} \int_{x_n}^{x_{n+1}} (x - x_n) \cdot \dots \cdot (x - x_{n-k+1}) dx$$

Обчислимо значення інтеграла в правій частині даного виразу для декількох k

При $k = 0$ одержимо: $\int_{x_n}^{x_{n+1}} dx = h$.

При $k = 1$ одержимо: $\int_{x_n}^{x_{n+1}} (x - x_n) dx = \frac{h^2}{2}$.

При $k = 2$ одержимо: $\int_{x_n}^{x_{n+1}} (x - x_n)(x - x_{n-1}) dx = \frac{5h^3}{6}$.

При $k = 3$ одержимо: $\int_{x_n}^{x_{n+1}} (x - x_n)(x - x_{n-1})(x - x_{n-2}) dx = \frac{18h^4}{8}$.

Тоді різницеву схему четвертого порядку Адамса можна записати після необхідних перетворень у вигляді

$$y_{i+1} = y_i + h \left[f_i + \frac{1}{2} \Delta f_i + \frac{5}{12} \Delta^2 f_i + \frac{3}{8} \Delta^3 f_i \right].$$

Кінцеві формули для методу Адамса

Якщо в отриману формулу для методу Адамса

$$y_{i+1} = y_i + h \left[f_i + \frac{1}{2} \Delta f_i + \frac{5}{12} \Delta^2 f_i + \frac{3}{8} \Delta^3 f_i \right].$$

підставити значення скінченних різниць Δf_i , то одержимо остаточні вирази для обчислень за методом Адамса:

$$y_{i+1} = y_i + \frac{h}{24} (55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3})$$

Цей метод називають явним методом Адамса четвертого порядку або методом Адамса-Башфорта [72].

Проблема обчислення початкових точок

Неможливо почати обчислення по одному лише відомому значенню y_0 . Розрахунки можна почати тільки з вузла x_3 , а не x_0 . Значення y_1, y_2, y_3 , необхідні для обчислення y_4 , потрібно одержати у будь-який інший спосіб.

Для цього можна використовувати цей же метод Адамса, але більш низьких порядків.

Розглянемо початкове наближення поліномом Ньютона:

$$y_{n+1} \approx y_n + \int_{x_n}^{x_{n+1}} N_0(x) dx.$$

1. При $m = 0$ поліном $N_0(x) = f(x_n, y_n) = f_n - \text{const}$.

Тому одержуємо метод Ейлера:

$$y_{n+1} = y_n + f(x_n, y_n)(x_{n+1} - x_n)$$

2. При $m = 1$ поліном $N_1(x) = \frac{f_n}{h}(x - x_{n-1}) - \frac{f_{n-1}}{h}(x - x_n)$

Обчисливши $y_{n+1} \approx y_n + \int_{x_n}^{x_{n+1}} N_1(x) dx$

одержимо двокроковий метод Адамса: $y_{n+1} = y_n + \frac{h}{2}(3f_n - f_{n-1})$.

3. При $m = 2$ поліном $N_2(x)$ – квадратичний поліном, що інтерполіює функцію по вузлах:

$$(x_{n-2}, f_{n-2}), (x_{n-1}, f_{n-1}), (x_n, f_n).$$

Обчисливши $y_{n+1} \approx y_n + \int_{x_n}^{x_{n+1}} N_2(x) dx$, одержимо трикроковий метод Адамса:

$$y_{n+1} = y_n + \frac{h}{12}(23f_n - 16f_{n-1} + 5f_{n-2})$$

Неявний метод Адамса (метод Адамса-Мултона)

Розглянемо початкове наближення поліномом Ньютона

$$y_{n+1} \approx y_n + \int_{x_n}^{x_{n+1}} N_{n+1}(x) dx,$$

що забезпечує інтерполяцію по $x_{n+1}, x_n, x_{n-1}, \dots, x_{n-m}$ точках.

При цьому виникає клас неявних методів, названих методами Адамса-Мултона.

1. Якщо $m = 0$, то $N_{n+1}(x)$ – лінійна функція, що проходить через точки (x_n, f_n) й (x_{n+1}, f_{n+1}) . Після інтегрування одержуємо двокроковий метод Адамса-Мултона:

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

2. Якщо $m = 1$, то $N_{n+1}(x)$ – квадратичний поліном, побудований по точках $(x_{n+1}, f_{n+1}), (x_n, f_n), (x_{n-1}, f_{n-1})$. Після інтегрування одержуємо трикроковий метод Адамса-Мултона:

$$y_{n+1} = y_n + \frac{h}{12}(5f_{n+1} + 8f_n - f_{n-1}).$$

3. Якщо $m = 2$ то $N_{n+1}(x)$ – кубічний поліном, побудований по точках $(x_{n+1}, f_{n+1}), (x_n, f_n), (x_{n-1}, f_{n-1}), (x_{n-2}, f_{n-2})$. Після інтегрування одержуємо чотирікроковий метод Адамса-Мултона:

$$y_{n+1} = y_n + \frac{h}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}).$$

3.15.11. Порівняння методів Адамса та Рунге-Кутта

Перевага методу Адамса

При однаковій точності метод Адамса більш економічний, оскільки він вимагає обчислення лише одного значення правої частини на кожному кроці (у методі Рунге-Кутта – чотирьох).

Недоліки методу Адамса

1. Неможливо почати обчислення по одному лише відомому значенню y_0 . Розрахунки можна почати тільки з вузла x_3 , а не x_0 . Значення y_1, y_2, y_3 , необхідні для обчислення y_4 , потрібно одержати у будь-який інший спосіб (наприклад, методом Рунге-Кутта), що суттєво ускладнює алгоритм.

2. Метод Адамса не дозволяє (без ускладнення формул) змінити крок h у процесі обчислень; цього недоліку позбавлені однокрокові методи.

3.15.12. Методи прогнозу й корекції або методи предиктор-коректор

Загальний алгоритм методів

На кожному кроці вводяться два етапи, що використовують багатокрокові методи:

1. За допомогою явного методу (*предиктора*) за відомими значеннями функції в попередніх вузлах обчислюють початкове наближення $y_{i+1} = y_{i+1}^{(0)}$ у новому вузлі.

2. Використовуючи неявний метод (*коректор*), у результаті ітерацій обчислюють наближення $y_{i+1}^{(1)}, y_{i+1}^{(2)}, \dots$

Метод прогнозу й корекції на основі методу Адамса четвертого порядку

Наведемо остаточний вигляд різницевих співвідношень: на етапі предиктора:

$$y_{i+1} = y_i + \frac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}). \quad (3.92)$$

На етапі коректора

$$y_{i+1} = y_i + \frac{h}{24}(9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}). \quad (3.93)$$

Явну схему (3.92) використовують на кожному кроці один раз, а за допомогою неявної схеми (3.93) будують ітераційний процес обчислення y_{i+1} , оскільки це значення входить у праву частину виразу $f_{i+1} = f(x_{i+1}, y_{i+1})$.

Локальна похибка методів Адамса k -го порядку – $O(h^k)$.

Методи Адамса мають кращу, у порівнянні з методами Рунге-Кутта, стійкість.

Формула із другими різницями:

$$y_{i+1} = y_i + q_i + \frac{1}{2}\Delta q_{i-1} + \frac{5}{12}\Delta^2 q_{i-2}, \text{ де } q_i = hf(x_i, y_i).$$

Приклад 3.51. Використовуючи метод Адамса із другими різницями, скласти таблицю наближених значень інтеграла диференціального рівняння $y' = 1 + 0.2 \cdot y \cdot \sin x - 1.5 \cdot y^2$, що задовольняє початковим умовам $y(0) = 0$ на відрізку $x \in [0, 1]$; крок $h = 0.1$. Усі обчислення вести із чотирма десятковими знаками. Початковий відрізок визначити методом Рунге-Кутта.

Розв'язок

1. Визначимо значення $y_1 = y(0.1), y_2 = y(0.2)$ (початковий відрізок) методом Рунге-Кутта. При цьому значення $y_{i+1} = y(x_{i+1})$, де $x_{i+1} = x_i + h$, обчислюємо за формулами:

$$y_{i+1} = y_i + \Delta y_i, \Delta y_i = \frac{1}{6}(k_1^i + 2k_2^i + 2k_3^i + k_4^i),$$

$$\text{де } k_1^i = hf(x_i, y_i), k_2^i = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_1^i}{2}\right),$$

$$k_3^i = hf\left(x_i + \frac{h}{2}, y_i + \frac{k_2^i}{2}\right), k_4^i = hf(x_i + h, y_i + k_3^i).$$

Результати усіх обчислень будемо розміщувати в таблиці.

Таблиця 3.23

Проміжні обчислення при розв'язуванні диференціального рівняння
методом Адамса

x	$y(x)$	$\sin x$	$0.2y\sin x$	$-1.5y^2$	$f(x, y)$	$hf(x, y)$	Δy
0	0	0	0	0	1	0.1	0.1000
0.05	0.05	0.0500	0.0005	-0.0038	0.9967	0.9967	0.1994
0.05	0.0498	0.0500	0.0005	-0.0037	0.9968	0.9968	0.1994
0.10	0.0997	0.0998	0.0020	-0.0149	0.9871	0.0987	0.0987
0.5979-1/6=0.0996							
0.10	0.0996	0.0998	0.0020	-0.0149	0.9871	0.0987	0.0987
0.15	0.1490	0.1494	0.0045	-0.0333	0.9712	0.0971	0.1942
0.15	0.1482	0.1494	0.0044	-0.0329	0.9715	0.0972	0.1944
0.20	0.1968	0.1987	0.0078	-0.0581	0.9497	0.0950	0.0950
0.5823*1/6=0.0970							
0.20	0.1966	0.1987	0.0078	-0.0580	0.94982		

2. Обчислення наступних значень $y_i = y(x_i)$, де $x_i = x_0 + ih$ ($i = 3, 4, \dots$), здійснюємо за формулою Адамса із другими різницями

$$y_{i+1} = y_i + q_i + \frac{1}{2} \Delta q_{i-1} + \frac{5}{12} \Delta^2 q_{i-2}, \text{ де } q_i = hf(x_i, y_i).$$

Зведемо в таблицю остаточні значення $y(x_i)$ й значення скінченних різниць, наявних у обчислювальній формулі.

Таблиця 3.24

Проміжні обчислення скінченних різниць

i	x_i	y_i	$f(x_i, y_i)$	$q_i = hf_i$	Δq_i	$\Delta^2 q_i$
0	0	0	0.1000	0.1000	-0.00129	-0.00244
1	0.1	0.0996	0.9871	0.09871	-0.00373	-0.00204
2	0.2	0.1966	0.9498	0.09498	-0.00577	-0.00154
3	0.3	0.2887	0.8921	0.08921	-0.00731	-0.00088
4	0.4	0.3742	0.8190	0.08190	-0.00819	-0.00035
5	0.5	0.4518	0.7371	0.07371	-0.00854	0.00008
6	0.6	0.5210	0.6517	0.06517	-0.00846	0.00049
7	0.7	0.5818	0.5671	0.05671	-0.00797	0.00067
8	0.8	0.6343	0.4874	0.04874	-0.00730	-
9	0.9	0.6792	0.4144	0.04144	-	-
10	0.0	0.7173	-	-	-	-

У наступній таблиці представлені результати обчислень за формулами Адамса із другими різницями.

Таблиця 3.25

Результати обчислень за формулами Адамса

i	2	3	4	5	6	7	8	9
y_i	0.1966	0.28870	0.37418	0.37418	0.54102	0.58177	0.63428	0.67924
q_i	0.09498	-0.08921	-0.08190	-0.08190	0.6517	0.05671	0.04874	0.04144
$\frac{1}{2}\Delta q_{i-1}$	-0.00186	-0.00288	-0.00366	-0.00366	-0.00427	-0.00423	-0.00398	-0.00365
$\frac{5}{12}\Delta^2 q_{i-2}$	-0.0102	-0.0085	-0.00064	-0.00064	-0.00015	0.00003	0.00020	0.00028
y_{i+1}	0.28870	0.37418	0.45178	0.45178	0.58177	0.63428	0.67924	0.71731

У наступній таблиці представлені результати обчислення значень функції.

Таблиця 3.26

Результати обчислення значень функції

x_i	y_i	$0.2 \sin x_i$	$0.2 y_i \sin x_i$	$-1.5 y_i^2$	$f(x_i, y_i)$
0.3	0.2887	0.0591	0.0171	-0.1250	0.8921
0.4	0.3742	0.0779	0.0292	0.2102	0.8190
0.5	0.4518	0.0959	0.0433	0.3062	0.7371
0.6	0.5210	0.1129	0.0588	0.4071	0.6517
0.7	0.5818	0.1288	0.0749	0.5078	0.5671
0.8	0.6343	0.1435	0.0910	0.6036	0.4874
0.9	0.6792	0.1567	0.1064	0.6920	0.4144

Метод Мілна

Метод Мілна – це метод типу предиктор-коректор четвертого порядку точності. Прогнозно-коригуючий метод Мілна використовує пару скінченно-різницевих формул.

Формула-предиктор

$$y_{n+1} = y_{n-3} + \frac{4h}{3}(2f_n - f_{n-1} + 2f_{n-2}) + O(h^5),$$

де $O(h^5) = \frac{28}{90}h^5 f^{(5)}$ – похибка формули прогнозу.

Формула-коректор (формула Симпсона)

$$y_{n+1} = y_{n-1} + \frac{h}{3}(f_{n+1} + 4f_n + f_{n-1}) + O(h^5),$$

де $O(h^5) = -\frac{1}{90}h^5 f^{(5)}$ – похибка формули корекції.

Метод Хеммінга

Це стійкий метод четвертого порядку, в основі якого лежать наступні формули прогнозу й корекції.

$$\text{Формула-предиктор: } y_{n+1} = y_{n-3} + \frac{4h}{3}(2f_n - f_{n-1} + 2f_{n-2}) + O(h^5),$$

де $O(h^5) = \frac{28}{90}h^5 f^{(5)}$ – похибка формули прогнозу.

$$\text{Формула-коректор: } y_{n+1} = \frac{1}{8}[9f_n - f_{n-2} + 3h(f_{n+1} + 2f_n - f_{n-1})] + O(h^5)$$

де $O(h^5) = -\frac{1}{40}h^5 f^{(5)}$ – похибка формули корекції.

Особливістю методу Хеммінга є те, що він дозволяє оцінювати похибки, внесені на стадіях прогнозу та корекції, і усувати їх.

Контрольні запитання

1. Розв'язати методом Ейлера диференціальне рівняння $y' = 2x + y$ при початковій умові $y(0) = 1$ на відрізку $[0; 0.5]$ із кроком 0.1.
2. Використовуючи метод Ейлера, побудувати наближений розв'язок для наступної задачі Коші: $\frac{dy}{dx} = 2x - y$, $x_0 = y_0 = 0$ на сітці із кроком 0.2 на відрізку $[0; 1]$.
3. Розв'язати задачу Коші $y' = 10y^2 - x$, $y(1) = 0$ на відрізку $[1, 2]$ методом Ейлера із кроком $h = 0.2$.
4. Методом Ейлера знайти розв'язок диференціального рівняння, $y' = x + y$, що задовольняє початковій умові $y(0) = 1$ на відрізку $[0, 0.5]$ з кроком $h = 0.1$.
5. Розв'язати методом Ейлера диференціальне рівняння $y' = \cos y + 3x$ з початковою умовою $y(0) = 1.3$ на відрізку $[0, 0.6]$, з кроком $h = 0.2$.

3.16. Чисельні методи розв'язування крайової задачі для звичайного диференціального рівняння (ЗДР)

3.16.1. Постановка задачі

Відмінність крайової задачі від задачі Коші (задачі з початковими умовами) полягає в тому, що розв'язок диференціального рівняння повинен задовольняти граничним умовам, що пов'язують значення шуканої функції

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

більш ніж в одній точці із граничними умовами, заданими на кінцях відрізка $[a, b]$: $y(a) = y_0, y(b) = y_1$ – граничні умови першого роду.

Слід знайти такий розв'язок $y(x)$ на цьому відрізку, який набуває на кінцях відрізка значення y_0, y_1 . Якщо функція $f(x, y, y')$ лінійна за аргументами y, y' , то дана задача – лінійна крайова задача, а якщо ні, то – нелінійна [75].

3.16.2. Скінченно-різницевий метод розв'язування крайової задачі з граничними умовами першого роду

Розглянемо двохточкову крайову задачу для лінійного диференціального рівняння другого порядку на відрізку $[a, b]$:

$$y'' + p(x)y' + q(x)y = f(x),$$

$$y(a) = y_0, y(b) = y_1$$

Введемо різницеву сітку на відрізку $[a, b]$ $\Omega^{(k)} = \{x_k = x_0 + hk\}$,
 $k = 0, 1, \dots, N, h = \frac{|b-a|}{N}$.

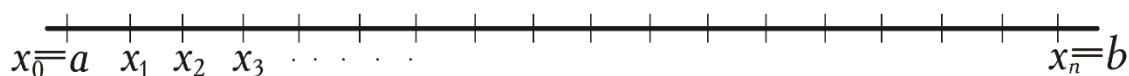


Рис. 3.36. Одновимірна сітка на відрізку $[a, b]$

Введемо різницеву апроксимацію похідних у такий спосіб:

$$y'_k = \frac{y_{k+1} - y_{k-1}}{2h} + O(h^2),$$

$$y''_k = \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} + O(h^2).$$

Підставляючи апроксимації похідних у рівняння

$$y'' + p(x)y' + q(x)y = f(x), y(a) = y_0, y(b) = y_1,$$

одержимо систему рівнянь для знаходження y_k :

$$\begin{cases} y_0 = y_a, \\ \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2} + p(x_k) \frac{y_{k+1} - y_{k-1}}{2h} + q(x_k)y_k = f(x_k), \\ y_N = y_b \end{cases}$$

де $k = 1, \dots, N-1$.

Приклад 3.52. Розв'язати крайову задачу:

$$\begin{cases} y'' - xy' - y = 0, \\ y(0) = 1, \\ y'(1) + 2y(1) = 0. \end{cases} \quad \text{з кроком } h = 0.2.$$

Розв'язок

У цій задачі $p(x) = x, q(x) = 1, f(x) = 0, N = 5,$

$x_0 = 0, x_1 = 0.2, x_3 = 0.6, x_4 = 0.8, x_5 = 1.0$

У всіх внутрішніх вузлах відрізка $[0,1]$ після заміни похідних їхніми різницевиими аналогами одержимо

$$(1 - 0.1x_k)y_{k-1} + (-2.04)y_k + (1 + 0.1x_k)y_{k+1} = 0, k = 1, \dots, 4.$$

На лівій границі $y_0 = 1$, на правій границі апроксимуємо похідну однобічною різницею 1-го порядку:

$$\frac{y_5 - y_4}{0.2} + 2y_5 = 0.$$

За допомогою групування доданків, зведення подібних членів і підстановки значень x_k та з урахуванням, що $y_0 = 1$, одержимо систему лінійних алгебраїчних рівнянь.

$$\begin{cases} -2.04y_1 + 1.02y_2 = -0.98, \\ 0.96y_1 - 2.04y_2 + 1.04y_3 = 0, \\ 0.94y_2 - 2.04y_3 + 1.06y_4 = 0, \\ 0.92y_3 - 2.04y_4 + 1.08y_5 = 0, \\ y_4 - 1.4y_5 = 0. \end{cases}$$

У даній трьохдіагональній системі виконана умова переважання діагональних елементів, і можна використовувати метод прогонки.

У результаті розв'язування системи методом прогонки одержимо наступні значення:

$$y_5 = 0.2233205, y_4 = 0.31265, y_3 = 0.43111, y_2 = 0.58303, y_1 = 0.77191$$

3.16.3. Скінченно-різницевий метод розв'язування крайової задачі з граничними умовами третього роду

Розглянемо диференціальне рівняння:

$$Ly = f(x), \text{ де } Ly = y'' + p(x)y' + q(x)y$$

з двохточковими лінійними крайовими умовами

$$\begin{cases} \alpha_0 y(a) + \alpha_1 y'(a) = \alpha \\ \beta_0 y(b) + \beta_1 y'(b) = \beta \end{cases}$$

$$(|\alpha_0| + |\alpha_1| \neq 0, |\beta_0| + |\beta_1| \neq 0),$$

де $p(x)$, $q(x)$ і $f(x)$ неперервні на відрізку $[a, b]$.

Зведемо крайову задачу до системи скінченно-різницевих рівнянь. Для цього розіб'ємо основний відрізок $[a, b]$ на n рівних частин довжини

$$h, \text{ де } h = \frac{b-a}{n}.$$

Точки розбивки мають абсциси:

$$x_i = x_0 + ih \quad (i = 0, 1, 2, \dots, n), \quad x_0 = a, \quad x_n = b.$$

Значення в точках x_i шуканої функції $y = y(x)$ і її похідних $y' = y'(x)$, $y'' = y''(x)$ позначимо відповідно через

$$y_i = y(x_i), y'_i = y'(x_i), y''_i = y''(x_i).$$

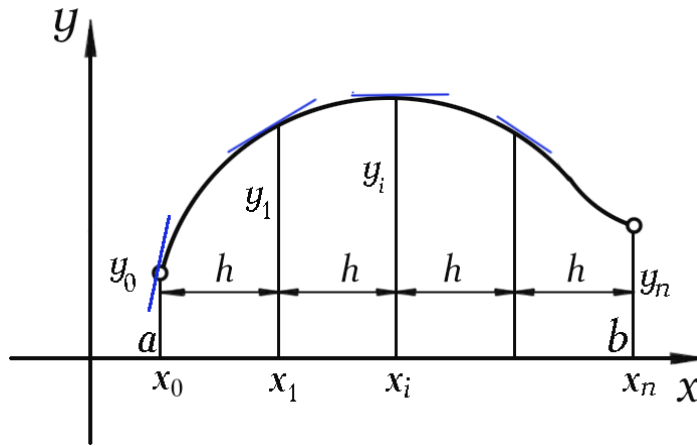


Рис. 3.35. Графічне представлення значень функції та її похідних у точках

1. Замінімо похідну y' скінченно-різницевиими відношеннями для внутрішніх точок x_i відрізка $[a, b]$

$$y'_i = \frac{y_{i+1} - y_{i-1}}{2h},$$

2. Замінімо похідну y'' скінченно-різницевиими відношеннями

$$y''_i = \frac{\frac{y_{i+1} - y_i}{h} - \frac{y_i - y_{i-1}}{h}}{h} = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}.$$

3. Для кінцевих точок $x_0 = a$ і $x_n = b$, щоб не виходити за межі відрізка $[a, b]$, задамо:

$$y'_0 = \frac{y_1 - y_0}{h}, y'_n = \frac{y_n - y_{n-1}}{h}$$

4. Більш точне визначення похідних на границі.

Виходячи зі способу дискретизації, одержуємо:

$$y_1 = y(x_0 + h) \text{ і } y_2 = y(x_0 + 2h)$$

Використовуємо формулу Тейлора:

$$y_1 = y_0 + hy'_0 + \frac{h^2}{2!} y''_0 + \frac{h^3}{3!} y'''_0 + \dots,$$

$$y_2 = y_0 + 2hy'_0 + \frac{(2h)^2}{2!} y''_0 + \frac{(2h)^3}{3!} y'''_0 + \dots$$

Складемо систему:
$$\begin{cases} y_1 = y_0 + hy'_0, \\ y_2 = y_0 + 2hy'_0. \end{cases}$$

Домножимо перше рівняння на 4 і віднімемо від другого:

$$\begin{cases} 4y_1 = 4y_0 + 4hy'_0, \\ y_2 = y_0 + 2hy'_0 \end{cases} \Rightarrow 4y_1 - y_2 = 3y_0 + 2hy'_0 \Rightarrow y'_0 = \frac{-y_2 + 4y_1 - 3y_0}{2h}.$$

Використовуємо формулу Тейлора з кінця:

$$y_{n-1} = y_n - hy'_n - \frac{h^2}{2!} y''_n + \frac{h^3}{3!} y'''_n + \dots,$$

$$y_{n-2} = y_n + 2hy'_n + \frac{(2h)^2}{2!} y''_n + \frac{(2h)^3}{3!} y'''_n + \dots$$

Аналогічно можна показати, що

$$\begin{cases} y_{n-1} = y_n - hy'_n, \\ y_{n-2} = y_n - 2hy'_n. \end{cases} \Rightarrow \begin{cases} 4y_{n-1} = 4y_n - 4hy'_n, \\ y_{n-2} = y_n - 2hy'_n. \end{cases} \Rightarrow$$

$$\Rightarrow 4y_{n-1} - y_{n-2} = 3y_n - 2hy'_n \Rightarrow y'_n = \frac{3y_n - 4y_{n-1} + y_{n-2}}{2h}$$

5. Проводимо дискретизацію граничних умов.

У вирази для граничних умов підставимо

$$\begin{cases} \alpha_0 y(a) + \alpha_1 y'(a) = \alpha \\ \beta_0 y(b) + \beta_1 y'(b) = \beta \end{cases}$$

Підставимо отримані значення похідних.

Одержимо систему з двох рівнянь:

$$\begin{cases} \alpha_0 y_0 + \alpha_1 \frac{-y_2 + 4y_1 - 3y_0}{2h} = \alpha, \\ \beta_0 y_n + \beta_1 \frac{3y_n - 4y_{n-1} + y_{n-2}}{2h} = \beta. \end{cases}$$

6. Проводимо дискретизацію початкового рівняння

$$y'' + p(x)y' + q(x)y = f(x),$$

використовуючи отримані скінченно-різницеві відношення.

Одержимо систему з $(n-1)$ алгебраїчних рівнянь:

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

$$x = x_i \quad (i = 1, 2, \dots, n-1)$$

У такий спосіб отримана лінійна система $n+1$ рівнянь із $n+1$ невідомими $y_0, y_1, y_2, \dots, y_n$, що є значеннями шуканої функції $y = y(x)$ у точках $x_0, x_1, x_2, \dots, x_n$

$$\begin{cases} \alpha_0 y_0 + \alpha_1 \frac{-y_2 + 4y_1 - 3y_0}{2h} = \alpha, \\ \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p_i \frac{y_{i+1} - y_{i-1}}{2h} + q_i y_i = f_i \\ \beta_0 y_n + \beta_1 \frac{3y_n - 4y_{n-1} + y_{n-2}}{2h} = \beta. \end{cases}$$

Розв'язавши цю систему, якщо це можливо, одержимо таблицю значень шуканої функції y .

3.16.4. Апроксимація диференціального рівняння скінченними різницями в загальному виді

Дано рівняння $Ly = f(x)$, де Ly має вигляд:

$$Ly = -p(x)y'' + q(x)y' + r(x)y$$

Розіб'ємо проміжок $[a, b]$ на n рівних частин. Нехай $h = \frac{b-a}{n}$.

Побудуємо сітку вузлів із кроком h : $x_i = x_0 + ih, i = 0, 1, \dots, n$.

Цю сітку вузлів називають основною (головною):

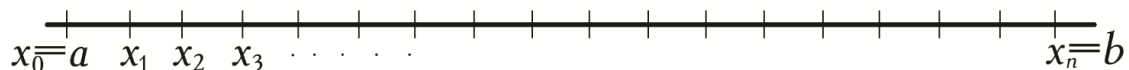


Рис. 3.37. Основна одновимірна сітка вузлів

Розв'язок задачі будемо шукати у вигляді таблиці значень у точках сітки $y_i = y(x_i)$ $i = 0, 1, \dots, n$.

Позначимо через p_i, q_i, r_i, f_i значення функцій $p(x), q(x), r(x), f(x)$ у точках сітки x_i .

Замінімо похідні в рівнянні $Ly = f(x)$ скінченно-різницевиими відношеннями з похибкою $O(h^2)$.

Тоді:

$$-p_i \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + q_i \frac{y_{i+1} - y_{i-1}}{2h} + r_i y_i = f_i, \text{ де } i = 1, 2, \dots, n-1.$$

Перегрупуємо члени:

$$\begin{aligned} -\frac{p_i}{h^2} y_{i+1} + \frac{2p_i}{h^2} y_i - \frac{p_i}{h^2} y_{i-1} + \frac{q_i}{2h} y_{i+1} - \frac{q_i}{2h} y_{i-1} + r_i y_i &= f_i \\ \left(\frac{p_i}{h^2} + \frac{q_i}{2h} \right) y_{i-1} - \left(r_i + \frac{2p_i}{h^2} \right) y_i + \left(\frac{p_i}{h^2} - \frac{q_i}{2h} \right) y_{i+1} &= -f_i. \end{aligned}$$

Увівши заміни

$$A_i = \left(\frac{p_i}{h^2} + \frac{q_i}{2h} \right), \quad B_i = \left(r_i + \frac{2p_i}{h^2} \right), \quad C_i = \left(\frac{p_i}{h^2} - \frac{q_i}{2h} \right), \quad G_i = -f_i,$$

одержимо систему рівнянь: $A_i y_{i-1} - B_i y_i + C_i y_{i+1} = G_i$, $i = 1, 2, \dots, n-1$

3.16.5. Апроксимація граничних умов. Граничні умови другого роду й змішані

Крім розглянутих граничних умов, названих граничними умовами першого роду, використовуються ще умови на похідні від розв'язку на кінцях – граничні умови другого роду: $y'(a) = \alpha, y'(b) = \beta$, або лінійна комбінація розв'язків і похідних – граничні умови третього роду:

$$\begin{aligned} \alpha_1 y(a) - \alpha_2 y'(a) &= \alpha, \quad |\alpha_1| + |\alpha_2| \neq 0, \quad \alpha_1 \alpha_2 \geq 0, \\ \beta_1 y(b) + \beta_2 y'(b) &= \beta, \quad |\beta_1| + |\beta_2| \neq 0, \quad \beta_1 \beta_2 \geq 0 \end{aligned}$$

На різних кінцях відрізка можна використовувати умови різних типів. Відсутні два рівняння для досягнення замкнутої системи відносно y_0, y_1, \dots, y_n одержимо, апроксимуючи граничні умови.

При $\alpha_2 = 0, \beta_2 = 0$ – граничні умови (I) роду.

При $\alpha_1 = 0, \beta_1 = 0$ – граничні умови (II) роду.

У інших випадках – граничні умови (III) роду.

Апроксимуючи похідні у виразах

$$\alpha_1 y(a) - \alpha_2 y'(a) = \alpha,$$

$$\beta_1 y(b) + \beta_2 y'(b) = \beta,$$

відповідно різницями «уперед» і «назад», одержимо

$$\alpha_1 y_0 - \alpha_2 \frac{(y_1 - y_0)}{h} = \alpha, \quad \beta_1 y_n + \beta_2 \frac{(y_n - y_{n-1})}{h} = \beta.$$

Перегрупуємо члени:

$$\alpha_1 y_0 - \frac{\alpha_2}{h} y_1 + \frac{\alpha_2}{h} y_0 = \alpha, \quad -\left(\alpha_1 + \frac{\alpha_2}{h}\right) y_0 + \frac{\alpha_2}{h} y_1 = -\alpha,$$

$$\beta_1 y_n + \frac{\beta_2}{h} y_n - \frac{\beta_2}{h} y_{n-1} = \beta, \quad \frac{\beta_2}{h} y_{n-1} - \left(\beta_1 + \frac{\beta_2}{h}\right) y_n = -\beta$$

Введемо заміни:

$$B_0 = \left(\alpha_1 + \frac{\alpha_2}{h}\right), \quad C_0 = \frac{\alpha_2}{h}, \quad G_0 = -\alpha, \quad A_n = \frac{\beta_2}{h}, \quad B_n = \left(\beta_1 + \frac{\beta_2}{h}\right), \quad G_n = -\beta$$

Одержимо два додаткових рівняння:

$$-B_0 y_0 + C_0 y_1 = G_0,$$

$$A_n y_{n-1} - B_n y_n = G_n$$

У підсумку одержуємо систему лінійних алгебраїчних рівнянь трьохдіагонального вигляду.

$$\begin{cases} -B_0 y_0 + C_0 y_1 = G_0, \\ A_i y_{i-1} - B_i y_i + C_i y_{i+1} = G_i, \\ \dots \\ A_n y_{n-1} - B_n y_n = G_n \end{cases}$$

3.16.6. Метод прогонки для розв'язування систем з трьохдіагональною матрицею

Будемо шукати розв'язок системи у вигляді:

$$y_i = s_i y_{i+1} + t_i, \quad i = 0, 1, \dots, n-1.$$

Прогоночні коефіцієнти s_i, t_i підлягають визначенню.

З нульового рівняння системи знаходимо: $s_0 = \frac{C_0}{B_0}, t_0 = \frac{-G_0}{B_0}$.

Підставимо $y_{i-1} = s_{i-1} y_i + t_{i-1}$, в i -те рівняння системи:

$$A_i (s_{i-1} y_i + t_{i-1}) - B_i y_i + C_i y_{i+1} = G_i,$$

Одержимо рекурентні формули:

$$s_i = \frac{C_i}{B_i - A_i s_{i-1}}, \quad t_i = \frac{A_i t_{i-1} - G_i}{B_i - A_i s_{i-1}}, \quad i = 1, 2, \dots, n$$

У n -ому рівнянні $C_n = 0$, отже $s_n = 0$.

З $y_i = s_i y_{i+1} + t_i, i = n$ одержуємо $y_n = t_n$.

Знову, використовуючи вираз $y_i = s_i y_{i+1} + t_i, i = 0, 1, \dots, n-1$, знаходимо $y_{n-1}, y_{n-2}, \dots, y_0$. Послідовність обчислень наведена в таблиці.

Таблиця 3.27

Послідовність обчислень методом прогонки

i	x_i	A_i	B_i	C_i	G_i	s_i	t_i	y_i
0	x_0	A_0	B_0	C_0	G_0	s_0	t_0	y_0
1	x_1	A_1	B_1	C_1	G_1	s_1	t_1	y_1
...
$n-1$	x_{n-1}	A_{n-1}	B_{n-1}	C_{n-1}	G_{n-1}	s_{n-1}	t_{n-1}	y_{n-1}
n	x_n	A_n	B_n	C_n	G_n	s_n	t_n	$y_n = t_n$

Пряма прогонка – заповнювати стовпці зверху вниз, крім останнього стовпця. Зворотна прогонка – знизу вгору, останній стовпець із розв'язком.

Контрольні завдання

1. Які умови називають граничними умовами першого роду?
2. Розв'язати крайову задачу $2x^2y'' - 6xy' + 8y = x^3$ на відрізку $[1, 4]$ з граничними умовами $y(1) = 0.5, y(4) = 0$
3. Розв'язати крайову задачу $xy'' - y' = x^2$ на відрізку $[1, 2]$ з граничними умовами $\begin{cases} y(1) + y'(1) = 5, \\ y(2) - y'(2) = 2. \end{cases}$
4. Розв'язати крайову задачу $xy'' - y' = 2x^2$ на відрізку $[1, 3]$ з граничними умовами: $\begin{cases} 6y(1) + 4y'(1) = 0, \\ 3y(3) - y'(3) = 1. \end{cases}$
5. Розв'язати крайову задачу $2x^2y'' - 2xy' - 6y = 10x^2$ на відрізку $[1, 3]$ з граничними умовами: $\begin{cases} 3y(1) + 3y'(1) = -9, \\ 2y(3) - y'(3) = 7. \end{cases}$

3.17. Розв'язування рівнянь з частинними похідними

Розглянемо наближені методи розв'язування деяких задач для диференціальних рівнянь із частинними похідними другого порядку з двома незалежними змінними [76].

У загальному випадку таке рівняння має вигляд:

$$F(x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}) = 0, \quad (3.94)$$

де x, y – невідомі змінні, u – шукана функція, $u_x, u_y, u_{xx}, u_{xy}, u_{yy}$ – її перші й другі частинні похідні по аргументах x і y (для зручності запису похідних «штрихи» не пишуть).

Визначення. Розв'язком рівняння $F(x, y, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}) = 0$, називають функцію $u = u(x, y)$, що перетворює це рівняння на тотожність. Графік розв'язку є поверхнею у просторі R^3 .

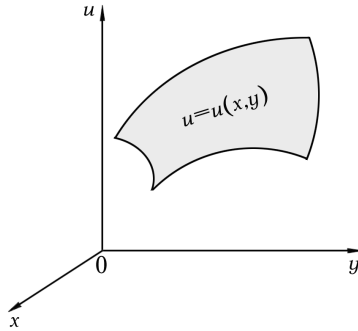


Рис. 3.38. Приклад розв'язку в R^3

Геометричне місце точок розв'язку $u = u(x, y)$ утворює *інтегральну поверхню* (інтегральна крива в одновимірному випадку).

3.17.1. Типи рівнянь із частинними похідними

Рівняння називають *лінійним*, якщо воно

1. першого степеня відносно шуканої функції й усіх її похідних,
2. не містить добутків похідних, тобто якщо це рівняння може бути записане у вигляді [77] :

$$A \frac{\partial^2 u}{\partial x^2} + 2B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + a \frac{\partial u}{\partial x} + b \frac{\partial u}{\partial y} + cu = F(x, y).$$

Рівняння зі *змінними коефіцієнтами*:

коефіцієнти $A(x, y), B(x, y), C(x, y), a(x, y), b(x, y), c(x, y)$ залежать тільки від x і y .

Рівняння з *постійними коефіцієнтами*:

коефіцієнти A, B, C, a, b, c не залежать від x, y .

Нехай $D = AC - B^2$ – дискримінант рівняння

$$A \frac{\partial^2 u}{\partial x^2} + 2B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + a \frac{\partial u}{\partial x} + b \frac{\partial u}{\partial y} + cu = F(x, y).$$

Залежно від значення функції D лінійне диференціальне рівняння відносять у даній області до одного з наступних типів:

$D > 0$ – еліптичний тип,

$D = 0$ – параболічний тип,

$D < 0$ – гіперболічний тип,

D – не зберігає постійного знака – змішаний тип.

Приклад 3.53. Утворення рівняння Лапласа

У рівняння:

$$A \frac{\partial^2 u}{\partial x^2} + 2B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + a \frac{\partial u}{\partial x} + b \frac{\partial u}{\partial y} + cu = F(x, y)$$

підставимо значення $A=1, B=0, C=1$ і $D = AC - B^2 > 0$.

Крім того, для спрощення задачі прийемо $a=0, b=0, c=0$. При стаціонарному процесі також $F(x, y) = 0$, одержуємо рівняння Лапласа:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \text{ – еліптичний тип.}$$

Таке рівняння, наприклад, описує розподіл температури $u = u(x, y)$ точки (x, y) для пластинки при стаціонарному розподіленні (тобто при розподіленні, що не залежить від часу) і відсутності джерел тепла.

Приклад 3.54. Утворення рівняння теплопровідності

У рівняння:

$$A \frac{\partial^2 u}{\partial x^2} + 2B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + a \frac{\partial u}{\partial x} + b \frac{\partial u}{\partial y} + cu = F(x, y)$$

підставимо значення $A=1, B=0, C=0$. $D = AC - B^2 = 0$. Крім того, $a=0, c=0$. Одержане таким чином рівняння називають рівнянням теплопровідності:

$$\frac{\partial u}{\partial t} - a^2 \frac{\partial^2 u}{\partial x^2} = F(x, t) \text{ – параболічний тип,}$$

де a – постійна, що залежить від фізичних властивостей стрижня,

$F(x, t)$ – функція, пов'язана з густиною джерел розподілення тепла.

Рівняння описує температуру $u = u(x, t)$ точки однорідного тонкого стрижня з абсцисою x для кожного моменту часу t .

Приклад 3.55. Рівняння теплопровідності без джерела тепла

У рівняння:

$$A \frac{\partial^2 u}{\partial x^2} + 2B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + a \frac{\partial u}{\partial x} + b \frac{\partial u}{\partial y} + cu = F(x, y)$$

підставимо значення $A=1, B=0, C=0, D=AC-B^2=0$.

Крім того, $a=0, c=0, F(x, y)=0$.

За таких умов одержуємо рівняння теплопровідності без джерела тепла:

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2} \text{ – параболічний тип.}$$

Увівши новий час $a^2 t = \tau$, одержуємо таке рівняння теплопровідності:

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2}.$$

Рівняння теплопровідності також є рівнянням параболічного типу.

Приклад 3.56. Утворення хвильового рівняння

У рівняння:

$$A \frac{\partial^2 u}{\partial x^2} + 2B \frac{\partial^2 u}{\partial x \partial y} + C \frac{\partial^2 u}{\partial y^2} + a \frac{\partial u}{\partial x} + b \frac{\partial u}{\partial y} + cu = F(x, y)$$

підставимо значення $A=1, B=0, C=-1$.

$D=AC-B^2 < 0$. Крім того, $a=0, b=0, c=0, F(x, y)=0$.

Після даних спрощень отримуємо хвильове рівняння:

$$\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2} \text{ – гіперболічний тип.}$$

Рівняння описує процес малих поперечних коливань струни. У цьому випадку $u(x, t)$ – поперечні переміщення (коливання) струни, a – швидкість поширення малих збурювань у матеріалі, з якого виготовлена струна.

3.17.2. Перша початково-крайова задача для рівняння теплопровідності

1. Задано рівняння теплопровідності (дифузії) [78].

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2}, \text{ де } 0 < x < l, t > 0$$

2. На границях $x=0$ і $x=l$ задані значення шуканої функції $u(x,t)$

у вигляді:

$$\begin{aligned} u(0,t) &= \varphi_0(t), \quad x=0, \quad t > 0; \\ u(l,t) &= \varphi_l(t), \quad x=l, \quad t > 0, \end{aligned} \quad \text{– граничні умови першого роду.}$$

3. Задані початкові умови: $u(x,0) = \psi(x)$, $0 \leq x \leq l$, $t = 0$,

$u(x,t)$ – розподіл температури в просторово-часовій області,

$\Omega \times T = \{0 \leq x \leq l, 0 \leq t \leq T\}$, a^2 – коефіцієнт теплопровідності,

а $\varphi_0(t)$, $\varphi_l(t)$ задають температуру на границях $x=0$ і $x=l$.

3.17.3. Друга початково-крайова задача для рівняння теплопровідності

1. Задано рівняння теплопровідності (дифузії).

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2}, \text{ де } 0 < x < l, t > 0$$

2. На границях $x=0$ і $x=l$ задані значення шуканої функції $u(x,t)$

у вигляді:

$$\begin{aligned} \frac{\partial u(0,t)}{\partial x} &= \varphi_0(t), \quad x=0, \quad t > 0; \\ \frac{\partial u(l,t)}{\partial x} &= \varphi_l(t), \quad x=l, \quad t > 0, \end{aligned} \quad \text{– граничні умови другого роду.}$$

3. Задані початкові умови: $u(x,0) = \psi(x)$, $0 \leq x \leq l$, $t = 0$

У термінах теорії теплообміну на границях області у цьому випадку задані теплові потоки.

3.17.4. Третя початково-крайова задача для рівняння теплопровідності

1. Задане рівняння теплопровідності (дифузії) [79].

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2}, \text{ де } 0 < x < l, t > 0$$

2. На границях $x=0$ і $x=l$ задані значення шуканої функції $u(x,t)$ у вигляді:

$$\alpha \frac{\partial u(0,t)}{\partial x} + \beta u(0,t) = \varphi_0(t), \quad x=0, t > 0; \quad \text{– граничні умови третього роду}$$

$$\gamma \frac{\partial u(l,t)}{\partial x} + \delta u(l,t) = \varphi_l(t), \quad x=l, t > 0, \quad \text{(змішані)}$$

3. Задані початкові умови: $u(x,0) = \psi(x), 0 \leq x \leq l, t = 0$

У термінах теорії теплообміну граничні умови задають теплообмін між газоподібним або рідким середовищем і границями розрахункової (обчислювальної) області обчислень з невідомими температурами $u(0,t), u(l,t)$.

Для розв'язування рівнянь параболічного типу широко використовують метод скінченних різниць.

На просторово-часову область $\Omega \times T$, де $0 \leq x \leq l, 0 \leq t \leq T$ нанесемо скінченно-різницеву сітку $\omega_{h\tau}$. $\omega_{h\tau} = \{x_j = jh, j = \overline{0, N}, t_k = k\tau, k = \overline{0, K}\}$

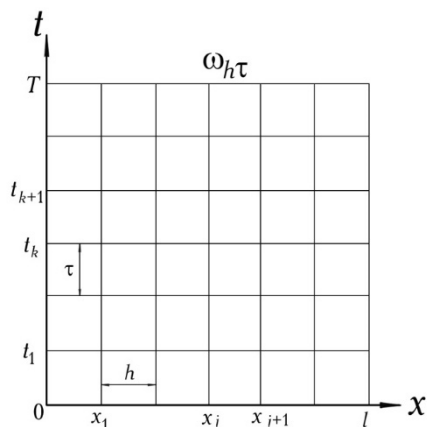


Рис. 3.39. Графічне представлення просторово-часової сітки ω_{ht}

Просторовий крок: $h = \frac{l}{N}$. Крок за часом: $\tau = \frac{T}{K}$.

Будемо шукати значення функції $u_j(k)$ у вузлах сітки ω_{ht} , кожен з вузлів визначають координатами (j, k) .

Розв'язок будемо одержувати послідовно по часових шарах.

У початковий момент часу відомий розподіл шуканої функції $u_j(0)$ у просторі $(0 \leq j \leq N)$ на початковому часовому шарі (кроці), одержуваний з початкових умов при $x_j = jh$: $u(x, 0) = \psi(x)$, $0 \leq x \leq l, t = 0$

Визначення. Сітковою функцією $u_j(k)$ називають відображення цілих аргументів j, k у значення функції: $u_j(k) = u(x_j, t_k)$.

Нехай відома сіткова функція $u_k(k)$ на шарі k й потрібно знайти значення сіткової функції $u_k(k+1)$ на шарі (кроці) $k+1$.

3.17.5. Явна схема для першої початково-крайової задачі

Виконаємо дискретизацію частинних похідних у рівнянні:

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2}$$

шляхом заміни їх відношенням скінченних різниць:

1. По часовий змінній: $\frac{\partial u}{\partial t} = \frac{u_j(k+1) - u_j(k)}{\tau} + O(\tau)$,

2. По просторовій змінній на k -му часовому кроці:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{j+1}(k) - 2u_j(k) + u_{j-1}(k)}{h^2} + O(h^2).$$

Підставимо їх у рівняння теплопровідності і одержимо явну скінченно-різницеву схему:

$$\frac{u_j(k+1) - u_j(k)}{\tau} = a^2 \frac{u_{j+1}(k) - 2u_j(k) + u_{j-1}(k)}{h^2} + O(\tau + h^2),$$

$$j = \overline{1, N-1}, \quad k = \overline{0, K-1},$$

$$u_0(k) = \varphi_0(t_k), \quad u_N(k) = \varphi_l(t_k), \quad k = \overline{0, K}, \quad u_j(0) = \psi(x_j), \quad j = \overline{0, N}$$

Явна скінченно-різницева схема

Значення сіткової функції $u_j(k+1)$ у вузлі сіткової області $(j, (k+1))$ може бути отримане з виразу:

$$u_j(k+1) = \frac{\tau a^2}{h^2} \left[u_{j+1}(k) + (h^2 - 2)u_j(k) + u_{j-1}(k) \right]$$

Введемо позначення $\sigma = a^2 \tau / h^2$. Тоді одержимо:

$$u_j(k+1) = \sigma u_{j+1}(k) + (1 - 2\sigma)u_j(k) + \sigma u_{j-1}(k), \quad j = \overline{1, N-1}, \quad k = 1, 2, \dots$$

При цьому використані значення сіткової функції

$u_{j-1}(k)$, $u_j(k)$ і $u_{j+1}(k)$ у вузлах $(j-1, k)$, (j, k) і $(j+1, k)$.

Ці значення відомі, оскільки були отримані на попередньому кроці обчислень.

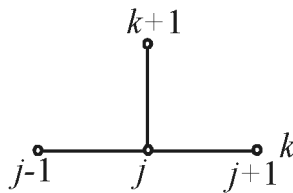


Рис. 3.40. Форма шаблону для обчислення в точці j

Геометрична інтерпретація явної скінченно-різницевої схеми, *шаблон явної схеми*, показана на рисунку.

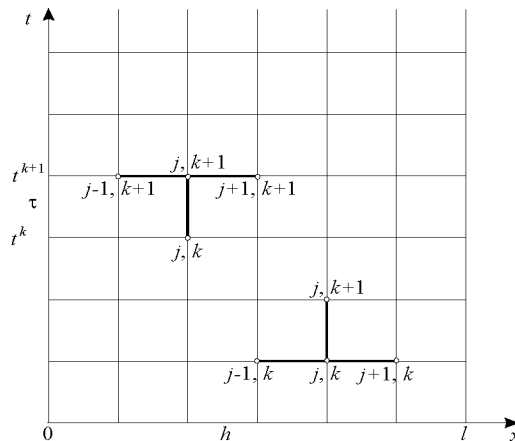


Рис. 3.41. Можливі положення шаблонів на сітці

3.17.6. Неявна схема для першої початково-крайової задачі

Виконаємо дискретизацію частинних похідних у рівнянні $\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2}$

шляхом заміни їх відношенням скінченних різниць:

$$1. \text{ По часовій змінній: } \frac{\partial u}{\partial t} = \frac{u_j(k+1) - u_j(k)}{\tau} + O(\tau),$$

2. По просторовій змінній на $(k+1)$ -му кроці:

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{j+1}(k+1) - 2u_j(k+1) + u_{j-1}(k+1)}{h^2} + O(h^2).$$

Підставимо їх у рівняння теплопровідності й одержимо *неявну скінченно-різницеву схему*:

$$\frac{u_j(k+1) - u_j(k)}{\tau} = a^2 \frac{u_{j+1}(k+1) - 2u_j(k+1) + u_{j-1}(k+1)}{h^2} + O(\tau + h^2),$$

$$j = \overline{1, N-1}, \quad k = \overline{0, K-1},$$

$$u_0(k+1) = \varphi_0(t_{k+1}), \quad u_N(k+1) = \varphi_l(t_{k+1}), \quad k = \overline{0, K}, \quad u_j(0) = \psi(x_j), \quad j = \overline{0, N}$$

Неявна скінченно-різницева схема

У рівнянні

$$\frac{u_j(k+1) - u_j(k)}{\tau} = a^2 \frac{u_{j+1}(k+1) - 2u_j(k+1) + u_{j-1}(k+1)}{h^2}$$

зведемо подібні члени

$$h^2 u_j(k+1) - h^2 u_j(k) = \tau a^2 u_{j+1}(k+1) - 2\tau a^2 u_j(k+1) + \tau a^2 u_{j-1}(k+1),$$

$$-h^2 u_j(k) = \tau a^2 u_{j-1}(k+1) - (h^2 + 2\tau a^2) u_j(k+1) + \tau a^2 u_{j+1}(k+1),$$

1. Розглянемо випадок $j=1$.

$$-h^2 u_1(k) = \tau a^2 u_0(k+1) - (h^2 + 2\tau a^2) u_1(k+1) + \tau a^2 u_2(k+1),$$

$$-u_1(k) - \frac{\tau a^2}{h^2} \varphi_0(t_{k+1}) = -\frac{(h^2 + 2\tau a^2)}{h^2} u_1(k+1) + \frac{\tau a^2}{h^2} u_2(k+1)$$

Введемо позначення:

$$\sigma = \frac{a^2\tau}{h^2}, b_1 = -(1 + 2\sigma), c_1 = \sigma, d_1 = -(u_1(k) + \sigma\varphi_0(t_{k+1}))$$

Одержимо перше рівняння системи: $b_1u_1(k+1) + c_1(k+1) = d_1$

2. Розглянемо випадок $j = \overline{2, N-2}$.

$$-h^2u_j(k) = \tau a^2u_{j-1}(k+1) - (h^2 + 2\tau a^2)u_j(k+1) + \tau a^2u_{j+1}(k+1)$$

$$-u_j(k) = \frac{\tau a^2}{h^2}u_{j-1}(k+1) - \left(1 + 2\frac{\tau a^2}{h^2}\right)u_j(k+1) + \frac{\tau a^2}{h^2}u_{j+1}(k+1)$$

Введемо позначення: $\sigma = \frac{a^2\tau}{h^2}, b_j = -(1 + 2\sigma), c_j = \sigma, d_j = -u_1(k)$

Одержимо $N-3$ рівняння системи для $j = \overline{2, N-2}$:

$$a_ju_{j-1}(k+1) + b_ju_j(k+1) + c_ju_{j+1}(k+1) = d_j$$

3. Розглянемо випадок $j = N-1$.

$$-u_j(k) - \frac{\tau a^2}{h^2}\varphi_l(t_{k+1}) = \frac{\tau a^2}{h^2}u_{j-1}(k+1) - \left(1 + 2\frac{\tau a^2}{h^2}\right)u_j(k+1)$$

Введемо позначення:

$$\sigma = \frac{a^2\tau}{h^2}, a_{N-2} = \sigma, b_{N-1} = -(1 + 2\sigma), d_{N-1} = -(u_1(k) + \sigma\varphi_l(t_{k+1}))$$

$N-1$ -е рівняння системи: $a_{N-1}u_{N-2}(k+1) + b_{N-1}u_{N-1}(k+1) = d_{N-1}$

3.17.7. Підсумкова СЛАР для розв'язування методом прогонки першої початково-крайової задачі

$$c_{N-1} = 0 \begin{cases} a_1 = 0 & \begin{cases} b_1u_1(k+1) + c_1u_2(k+1) = d_1, & j=1 \\ a_ju_{j-1}(k+1) + b_ju_j(k+1) + c_ju_{j+1}(k+1) = d_j, & j = \overline{2, N-2}, \text{ де} \\ a_{N-1}u_{N-2}(k+1) + b_{N-1}u_{N-1}(k+1) = d_{N-1}. & j = N-1 \end{cases} \end{cases}$$

$$\begin{aligned}
 a_j &= \sigma \text{ при } j = \overline{2, N-1}, & b_j &= -(1+2\sigma), \text{ при } j = \overline{1, N-1}, \\
 c_j &= \sigma, \text{ при } j = \overline{1, N-2}, & d_j &= -u_j(k) \text{ при } j = \overline{2, N-2}, \\
 d_1 &= -(u_1(k) + \sigma\varphi_0(t_{k+1})), & d_{N-1} &= -(u_{N-1}(k) + \sigma\varphi_l(t_{k+1})), \quad \sigma = a^2\tau/h^2
 \end{aligned}$$

Геометрична інтерпретація неявної скінченно-різницевої схеми, шаблон неявної схеми, показана на рисунку.

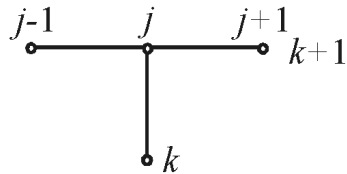


Рис. 3.42. T-шаблон для обчислення в точці j

Порівняння явної й неявної скінченно-різницевих схем

Перевага явної схеми. Важлива перевага явної схеми полягає в тому, що розв'язок на верхньому часовому шарі t_{k+1} одержуємо відразу (без розв'язування СЛАР).

Недолік явної схеми. Розв'язок може бути отриманий тільки при виконанні умов стійкості $\sigma = \frac{a^2\tau}{h^2} \leq \frac{1}{2}$, які накладають обмеження на співвідношення кроків τ і h .

Перевага неявної схеми. Така схема абсолютно стійка. Розв'язок можна одержати при будь-яких співвідношеннях кроків τ і h .

Недолік неявної схеми. При реалізації неявної схеми зростає обчислювальна складність алгоритму, оскільки потрібне розв'язування СЛАР.

3.17.8. Явна схема для третьої початково-крайової задачі

Граничні умови 2-го й 3-го роду містять похідні першого порядку від шуканої функції по x .

Тому потрібна їх апроксимація різницями 1-го порядку.

$$\frac{\partial u}{\partial x} = \frac{u_1(k+1) - u_0(k+1)}{h} + O(h), \quad \text{при } j = 0,$$

$$\frac{\partial u}{\partial x} = \frac{u_N(k+1) - u_{N-1}(k+1)}{h} + O(h), \quad \text{при } j = N$$

Підставимо отримані відношення різниць у вирази для граничних умов третього роду.

$$\begin{aligned} \alpha \frac{\partial u(0,t)}{\partial x} + \beta u(0,t) &= \varphi_0(t), \quad x=0, \quad t > 0; \\ \gamma \frac{\partial u(l,t)}{\partial x} + \delta u(l,t) &= \varphi_l(t), \quad x=l, \quad t > 0, \end{aligned}$$

Одержимо граничні умови після дискретизації:

$$\begin{aligned} \alpha \frac{u_1(k+1) - u_0(k+1)}{h} + \beta u_0(k+1) &= \varphi_0(t_{k+1}) + O(h); \\ \gamma \frac{u_N(k+1) - u_{N-1}(k+1)}{h} + \delta u_N(k+1) &= \varphi_l(t_{k+1}) + O(h). \end{aligned}$$

Доповнюючи отриманими рівняннями явну скінченно-різницеву апроксимацію у внутрішніх вузлах, одержимо явну різницеву схему для третьої початково-крайової задачі.

$$\left\{ \begin{aligned} \alpha \frac{u_1(k+1) - u_0(k+1)}{h} + \beta u_0(k+1) &= \varphi_0(t_{k+1}), \\ \frac{u_j(k+1) - u_j(k)}{\tau} &= a^2 \frac{u_{j+1}(k+1) - 2u_j(k+1) + u_{j-1}(k+1)}{h^2}, \quad j = \overline{1, N-1}, k = \overline{0, K-1} \\ \gamma \frac{u_N(k+1) - u_{N-1}(k+1)}{h} + \delta u_N(k+1) &= \varphi_l(t_{k+1}) \end{aligned} \right.$$

Виразимо явно значення шуканої функції й одержимо алгоритм переходу на $k+1$ -й часовий шар (крок) з використанням явної схеми:

$$u_j(k+1) = \sigma u_{j+1}(k) + (1 - 2\sigma)u_j(k) + (1 - \sigma)u_{j-1}(k), \quad \sigma = a^2\tau/h^2, \quad j = \overline{1, N-1},$$

$$u_0(k+1) = -\frac{\alpha/h}{\beta - \alpha/h} u_1(k+1) + \frac{\varphi_0(t_{k+1})}{\beta - \alpha/h},$$

$$u_N(k+1) = -\frac{\gamma/h}{\delta + \gamma/h} u_{N-1}(k+1) + \frac{\varphi_l(t_{k+1})}{\delta + \gamma/h},$$

Спочатку обчислюють значення шуканої функції у всіх внутрішніх вузлах на новому часовому шарі, а потім визначають значення на границях.

3.17.9. Неявна схема для третьої початково-крайової задачі

При використанні неявної скінченно-різницевої схеми одержуємо наступний різницевий аналог диференціальної задачі:

$$\begin{cases} b_0 u_0(k+1) + c_0 u(k+1) = d_0, \\ a_j u_{j-1}(k+1) + b_j u_j(k+1) + c_j u_{j+1}(k+1) = d_j, \quad j = \overline{1, N-1} \\ a_N u_{N-1}(k+1) + b_N u_N(k+1) = d_N. \end{cases}$$

$$b_0 = \beta - a/h, \quad c_0 = \alpha/h, \quad d_0 = \frac{\varphi_0(t_{k+1})}{\beta - \alpha/h},$$

$$a_N = -\gamma/h, \quad b_N = \delta + \gamma/h, \quad d_N = \frac{\varphi_l(t_{k+1})}{\delta + \gamma/h},$$

$$a_j = \sigma, \quad b_j = -(1 + 2\sigma), \quad c_j = \sigma, \quad d_j = -u_j(k), \quad j = \overline{1, N-1}, \quad \sigma = a^2 \tau / h^2$$

Для одержання розв'язку на новому часовому шарі розв'язують систему лінійних алгебраїчних рівнянь з трьохдіагональною матрицею.

3.17.10. Перша початково-крайова задача для рівняння гіперболічного типу

Якщо кінці струни рухаються за заданими законами, тобто на кінцях задані переміщення (або значення шуканої функції), то перша початково-крайова задача для хвильового рівняння має вигляд:

$$\begin{cases} \frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < l, \quad t > 0; \\ u(0, t) = \varphi_0(t), \quad x = 0, \quad t > 0; \\ u(l, t) = \varphi_l(t), \quad x = l, \quad t > 0; \\ u(x, 0) = \psi_1(x), \quad 0 \leq x \leq l, \quad t = 0; \\ \frac{\partial u(x, 0)}{\partial t} = \psi_2(x), \quad 0 \leq x \leq l, \quad t = 0. \end{cases}$$

Якщо кінці струни жорстко закріплені, то $\varphi_0(t) = 0$, $\varphi_l(t) = 0$.

3.17.11. Друга початково-крайова задача для рівняння гіперболічного типу

Якщо на кінцях струни задані значення сили, яка за законом Гука пропорційна значенням похідної переміщення по просторовій змінній (тобто на кінцях задані значення перших похідних по змінній x), то маємо *другу початково-крайову задачу для хвильового рівняння*:

$$\left\{ \begin{array}{l} \frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < l, \quad t > 0; \\ \frac{\partial u(0,t)}{\partial x} = \varphi_0(t), \quad x = 0, \quad t > 0; \\ \frac{\partial u(l,t)}{\partial x} = \varphi_l(t), \quad x = l, \quad t > 0; \\ u(x,0) = \psi_1(x), \quad 0 \leq x \leq l, \quad t = 0; \\ \frac{\partial u(x,0)}{\partial t} = \psi_2(x), \quad 0 \leq x \leq l, \quad t = 0. \end{array} \right.$$

Якщо кінці струни вільні, то $\varphi_0(t) = 0$, $\varphi_l(t) = 0$.

3.17.12. Третя початково-крайова задача для рівняння гіперболічного типу

Нарешті, в умовах, коли кінці закріплені *пружно*, тобто на кінцях діють сили, пропорційні переміщенням, маємо *третю початково-крайову задачу для хвильового рівняння*

$$\left\{ \begin{array}{l} \frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < l, \quad t > 0; \\ \alpha \frac{\partial u(0,t)}{\partial x} + \beta u(0,t) = \varphi_0(t), \quad x = 0, \quad t > 0; \\ \gamma \frac{\partial u(l,t)}{\partial x} + \delta u(l,t) = \varphi_l(t), \quad x = l, \quad t > 0; \\ u(x,0) = \psi_1(x), \quad 0 \leq x \leq l, \quad t = 0; \\ \frac{\partial u(x,0)}{\partial t} = \psi_2(x), \quad 0 \leq x \leq l, \quad t = 0. \end{array} \right.$$

3.17.13. Скінченно-різницева апроксимація рівнянь гіперболічного типу

Розглянемо першу початково-крайову задачу для хвильового рівняння. На просторово-часовій сітці можна апроксимувати

диференціальне рівняння $\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}$ явною скінченно-різницевою

схемою:
$$\frac{u_j(k+1) - 2u_j(k) + u_j(k-1)}{\tau^2} = a^2 \frac{u_{j+1}(k) - 2u_j(k) + u_{j-1}(k)}{h^2},$$

де $j = \overline{1, N-1}$, $k = 1, 2, \dots$ – явна схема.

Геометрична інтерпретація явної схеми, шаблон явної схеми показаний на рисунку.

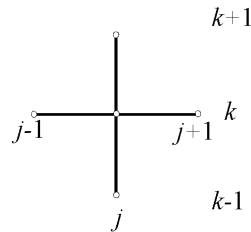


Рис. 3.43. Хрестовидний шаблон

Хвильове рівняння $\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}$ може бути апроксимовано за

допомогою неявної скінченно-різницевої схеми:

$$\frac{u_j(k+1) - 2u_j(k) + u_j(k-1)}{\tau^2} = a^2 \frac{u_{j+1}(k+1) - 2u_j(k+1) + u_{j-1}(k+1)}{h^2}$$

де $j = \overline{1, N-1}$, $k = 1, 2, \dots$ – неявна схема.

Геометрична інтерпретація неявної схеми, шаблон неявної схеми показаний на рисунку 3.44.

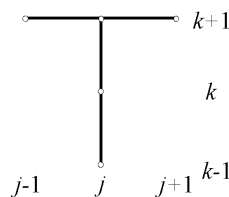


Рис. 3.44. Шаблон для хвильового рівняння

3.17.14. Перша початково-крайова задача для рівняння еліптичного типу

Розглянемо першу початково-крайову задачу для рівняння Лапласа

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

у прямокутнику $0 \leq x \leq l_1, 0 \leq y \leq l_2$, на який накладемо сітку

$$\omega_{h_1 h_2} = \{x_i = ih, i = \overline{0, N}; y_j = jh_2, j = \overline{0, N_2}\}$$

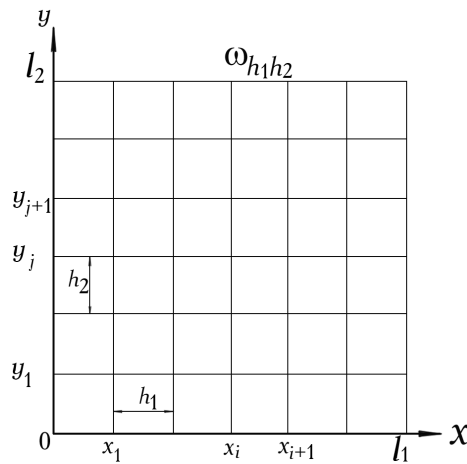


Рис. 3.45. Сітка дискретизації з кроками h_1 і h_2 .

На цій сітці апроксимуємо диференціальну задачу у внутрішніх вузлах за допомогою відношення скінченних різниць.

Введемо сіткову функцію $u_{ij}, i = \overline{0, N_1}, j = \overline{0, N_2}$.

Замінімо похідні $\frac{\partial^2 u}{\partial x^2}$ й $\frac{\partial^2 u}{\partial y^2}$ відношеннями скінченних різниць за

формулами

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2}, \quad \frac{\partial^2 u}{\partial y^2} \approx \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2}.$$

Тоді будемо мати $\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} = 0,$

і, отже, $u_{ij} = \frac{1}{4} [u_{i+1j} + u_{i-1j} + u_{ij+1} + u_{ij-1}]$, $i = \overline{1, N_1 - 1}$, $j = \overline{1, N_2 - 1}$

СЛАР має діагональний вигляд. Розв'язувати її можна різними методами лінійної алгебри, наприклад, ітераційними методами, методом прогонки і т. ін.

3.17.15. Третя початково-крайова задача для рівняння еліптичного типу

Третя крайова задача для рівняння Пуассона має вигляд:

$$\left\{ \begin{array}{l} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y), \quad 0 \leq x \leq l_1, \quad 0 \leq y \leq l_2; \\ \alpha_1 \frac{\partial u(0, y)}{\partial x} + \beta_1 u(0, y) = \varphi_1(y); \\ \alpha_2 \frac{\partial u(l_1, y)}{\partial x} + \beta_2 u(l_1, y) = \varphi_2(y); \\ \alpha_3 \frac{\partial u(x, 0)}{\partial x} + \beta_3 u(x, 0) = \varphi_3(x); \\ \alpha_4 \frac{\partial u(x, l_2)}{\partial x} + \beta_4 u(x, l_2) = \varphi_4(x). \end{array} \right.$$

Побудуємо різницеву схему, що апроксимує третю крайову задачу для рівняння Пуассона в прямокутнику.

1. Як і раніше в прямокутнику $0 \leq x \leq l_1$, $0 \leq y \leq l_2$ побудуємо сітку

$$\omega_{h_1 h_2} = \{x_i = ih_1, i = \overline{0, N_1}; y_j = jh_2, j = \overline{0, N_2}\}$$

2. На цій сітці апроксимуємо диференціальну задачу у внутрішніх вузлах за центрально-різницевою схемою:

$$\frac{u_{i+1j} - 2u_{ij} + u_{i-1j}}{h_1^2} + \frac{u_{ij+1} - 2u_{ij} + u_{ij-1}}{h_2^2} = f(x_i, y_j), \quad i = \overline{1, N_1 - 1}, \quad j = \overline{1, N_2 - 1}$$

3. Граничні умови апроксимуємо з першим порядком за допомогою направлених різниць:

$$\alpha_1 \frac{u_{1j} - u_{0j}}{h_1} + \beta_1 u_{0j} = \varphi_1(y_j), \quad j = \overline{1, N_2 - 1},$$

$$\alpha_2 \frac{u_{N_1j} - u_{N_1j}}{h_1} + \beta_2 u_{N_1j} = \varphi_2(y_j), \quad j = \overline{1, N_2 - 1},$$

$$\alpha_3 \frac{u_{i1} - u_{i0}}{h_2} + \beta_3 u_{i0} = \varphi_3(x_i), \quad i = \overline{1, N_1 - 1},$$

$$\alpha_4 \frac{u_{iN_2} - u_{iN_2-1}}{h_2} + \beta_4 u_{iN_2} = \varphi_4(x_i), \quad i = \overline{1, N_1 - 1}$$

У результаті отримана СЛАР, що містить $(N_1 + 1) \times (N_2 + 1) - 4$ рівнянь відносно невідомих u_{ij} ($i = \overline{0, N_1}, j = \overline{0, N_2}$, при цьому кутові вузли з координатами (i, j) , рівними $(0, 0), (0, N_2), (N_1, 0), (N_1, N_2)$, не беруть участь в обчисленнях. Як і у випадку граничних умов першого роду, вона має п'ятидіагональний вигляд і може бути вирішена, наприклад, ітераційним методом Лібмана.

Контрольні завдання

1. Дослідити тип рівняння з частинними похідними:

$$y^2 \frac{\partial^2 u}{\partial y^2} - 2x^2 \frac{\partial^2 u}{\partial x^2} = 0.$$

2. Звести до канонічного вигляду таке рівняння:

$$x \frac{\partial^2 u}{\partial x^2} + y \frac{\partial^2 u}{\partial y^2} + 4 \frac{\partial u}{\partial x} + 6 \frac{\partial u}{\partial y} = 0.$$

3. На відрізку $0 \leq x \leq l$ розв'язати рівняння дифузії: $\frac{\partial u}{\partial t} - 2a^2 \frac{\partial^2 u}{\partial x^2} = 0$

при наступних крайових умовах:

$$u(x, 0) = 4x, \quad u(0, t) = u(l, t) = 0, \quad t \in (0, \infty).$$

4. Розв'язати крайову задачу для хвильового рівняння на відрізку $0 \leq x \leq l$

$$\frac{\partial^2 u}{\partial t^2} = 32 \frac{\partial^2 u}{\partial x^2}.$$

Крайові умови мають такий вигляд:

$$u(0, t) = u(l, t) = 0, \quad u(x, 0) = 0, \quad \frac{\partial u(x, 0)}{\partial t} = \sin \frac{2\pi x}{l}.$$

5. Розв'язати крайову задачу для рівняння Лапласа $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$

у прямокутнику $[0, a] \times [0, b]$ при таких крайових умовах:

$$\begin{cases} u(0, y) = a, & u(a, y) = b, \\ u(x, 0) = 0, & u(x, b) = 0. \end{cases}$$

3.18. Сіткові методи розв'язування диференціальних рівнянь

3.18.1. Розв'язування задачі Дирихле методом сіток

Ідея *методу сіток* (або, інакше, методу *скінченних різниць*) для наближеного розв'язування крайових задач для двовимірних диференціальних рівнянь полягає в наступному:

1) у плоскій області G , у якій шукають розв'язок, будують *сіткову область* G_h , що складається з однакових кліток (рис. 3.46). Сіткова область G_h наближає дану область G ;

2) задане диференціальне рівняння заміняють у вузлах побудованої сітки відповідним скінченно-різницеvim рівнянням;

3) на підставі граничних умов встановлюються значення шуканого розв'язку в граничних вузлах області G_h .

Розв'язавши отриману систему скінченно-різницеvих рівнянь, для чого потрібно розв'язати алгебраїчну систему з великою кількістю невідомих, ми знайдемо значення шуканої функції у вузлах сітки, тобто будемо мати чисельний розв'язок нашої задачі [80].

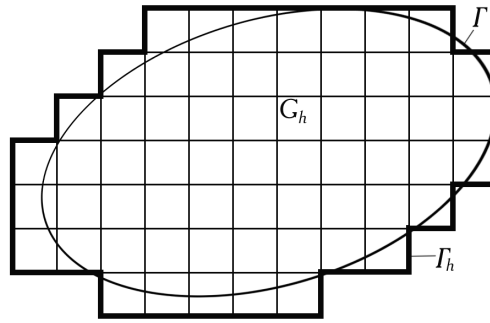


Рис. 3.46. Наближене представлення області G областю G_h

Вибір сіткової області проводиться залежно від конкретної задачі, але у всіх випадках контур Γ_h сіткової області G_h слід вибирати так, щоб він якомога краще апроксимував контур Γ заданої області G .

Сіткова область може складатися із квадратних, прямокутних, трикутних і інших кліток. Від вибору основного розміру клітки h залежить величина залишкового члена R_h при заміні диференціального рівняння скінченно-різницеvim. Отже, розмір h теоретично повинен визначатися вимогою, щоб цей залишковий член був меншим за похибку, припустиму при розв'язуванні. Однак такий шлях не завжди доцільний, тому що одержана при цьому величина h настільки мала і, отже, число кліток настільки велике, що розв'язування виявляється практично нездійсненним.

Зазвичай задачу розв'язують спочатку при великому значенні h , тобто при малому числі кліток, і лише після того, як задача грубо приблизно вирішена для цієї великої сітки, переходять до більш дрібної сітки або у всій розглянутій області, або в будь-якій її частині.

Ідея методу сіток відома давно, за часів Ейлера. Однак практичне використання цього методу наражалось на серйозні труднощі, тому що одержання за його допомоги досить точного розв'язку крайової задачі зазвичай призводило до колосальних систем алгебраїчних рівнянь, на розв'язування яких при ручному обчисленні були потрібні роки

обчислювальної праці. Становище різко змінилося з появою швидкодіючих електронних обчислювальних машин. Метод сіток допускає зручну реалізацію на електронних обчислювальних машинах, тому що застосування його зазвичай зводиться до масової повторюваності однорідних циклів. У наш час метод сіток є одним з найбільш ефективних методів розв'язування лінійних, а також почасти нелінійних задач математичної фізики.

Покажемо застосування методу сіток для побудови розв'язку задачі Дирихле:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \quad (3.95)$$

при $(x, y) \in G$ й $u(P) = \varphi(P)$ при $P \in \Gamma$, де $\varphi(P) = \varphi(x, y)$ – задана неперервна функція, причому для простоти розглянемо лише випадок квадратної сітки. Будемо припускати, що область G обмежена простим замкненим кусочно-гладким контуром Γ .

Вибравши крок h , побудуємо квадратну сітку

$$x_i = x_0 + ih, y_j = y_0 + jh, (i, j = 0, \pm 1, \pm 2, \dots)$$

з таким розрахунком, щоб вузли (x_i, y_j) або належали області G , або відстояли від її границі Γ на відстані, меншій за h .

Точки (вузли) сітки S_h називають *сусідніми*, якщо вони віддалені один від одного в напрямку осі Ox або осі Oy на відстань, що дорівнює кроку сітки h . Вузол A_h називають S_h *внутрішнім*, якщо він належить області G , а всі чотири сусідні з ним вузла – множині вузлів сітки, а якщо ні, то його називають *граничним*.

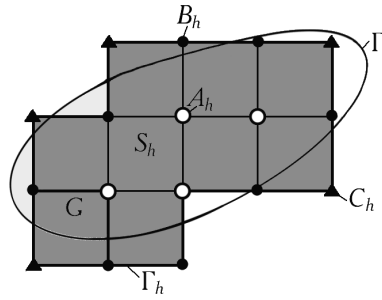


Рис. 3.47. Різні типи вузлів решітки

Як видно з рисунка, вузли B_h і C_h є граничними. Внутрішні вузли позначені світлими кружками, а граничні вузли – темними кружками та трикутниками.

Граничний вузол сітки S_h називають вузлом *першого роду*, якщо він має сусідній внутрішній вузол цієї сітки (наприклад, вузол B_h на рисунку). Якщо граничний вузол не є сусіднім із жодним внутрішнім вузлом, то такий вузол називають граничним вузлом *другого роду* (на рисунку такі вузли C_h). Внутрішні вузли й граничні вузли другого роду сітки S_h називають *розрахунковими точками*. Граничні вузли другого роду не входять в обчислення й можуть бути вилучені із сітки S_h (на рисунку граничні вузли другого роду позначені темними трикутниками).

Щодо сітки S_h припустимо, що множина її розрахункових точок «зв'язна», тобто будь-які дві розрахункові точки можна з'єднати ланцюжком вузлів, кожен два суміжні елементи якого є сусідніми вузлами.

Крім того, будемо вважати багатокутну сіткову область G_h обраною так, щоб її геометрична границя Γ_h якомога ближче примикала до границі Γ області G . Помітимо, що вузлові точки контуру Γ_h можуть лежати як усередині, так і поза областю G .

Значення шуканої функції $u = u(x, y)$ у точках (x_i, y_j) позначимо через $u_{ij} = u(x_i, y_j)$. Прямуючи загальною схемою, для кожної внутрішньої

точки (x_i, y_j) сітки заміняємо диференціальне рівняння (3.95) скінченно-різницеvim рівнянням

$$u_{ij} = \frac{1}{4}(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1}), \quad (3.96)$$

де $(x_{i\pm 1}, y_{j\pm 1})$ – розрахункові точки.

У граничних вузлах першого роду B_h сітки S_h одержуємо

$$u(B_h) = u(B) = \varphi(B), \quad (3.97)$$

де B – найближча до B_h точка границі Γ .

Розв'язавши систему (3.96), одержимо наближені значення шуканої функції $u = u(x, y)$ у вузлах сіткової області G_h . Тим самим буде знайдений наближений чисельний розв'язок задачі Дирихле для області G_h . Можна показати, що в загальному випадку похибка наближеного розв'язку має порядок $O(h)$. Для розв'язування системи (3.96) можуть бути використані розглянуті раніше чисельні методи розв'язування систем алгебраїчних рівнянь.

3.18.2. Процес Лібмана

Якщо число вузлів сітки S_h досить велике, то безпосередній розв'язок системи (3.96) стає утрудненим. Крім того, для криволінійної області G значення функції u у граничних вузлах сітки обрані занадто грубо. Ці обставини змушують для розв'язання зазначеної системи використовувати ітераційні методи з одночасним виправленням граничних значень.

Згідно із процесом усереднення Лібмана, вибравши початкові наближення $u_{ij}^{(0)}$, послідовні наближення $u_{ij}^{(k)}$ для внутрішніх вузлів (x_i, y_j) сітки S_h визначаємо за формулою

$$u_{ij}^{(k)} = \frac{1}{4}(u_{i-1,j}^{(k-1)} + u_{i+1,j}^{(k-1)} + u_{i,j-1}^{(k-1)} + u_{i,j+1}^{(k-1)}) \quad (k = 1, 2, \dots). \quad (3.98)$$

Що стосується граничних вузлів A_k сітки S_h , то значення функції $u(A_h)$ у цих вузлах визначаємо послідовно за формулами лінійної інтерполяції:

$$u^{(0)}(A_h) = u(A) = \varphi(A),$$

$$u^{(k)}(A_h) = u(A) + \frac{u^{(k-1)}(B) - u(A)}{h + \delta} \delta \quad (k=1, 2, \dots), \quad (3.99)$$

де A – найближча до A_h точка границі Γ ($u(A) = \varphi(A)$), B – найближчий до A_h внутрішній вузол сітки S_h й δ – відстань вузла A_h від точки A , причому $\delta > 0$, якщо A_h – внутрішня точка області G і $\delta < 0$ якщо A_h – зовнішня точка області G . (рис. 3.48)

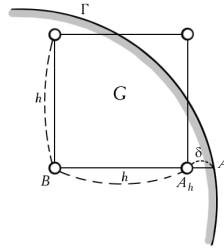


Рис. 3.48. Розташування зовнішньої точки сітки

У окремому випадку, якщо вузол A_h лежить на границі Γ ($A_h = A, \delta = 0$), то маємо точно $u^{(k)}(A_h) = u(A) = \varphi(A)$.

На практиці, після деякого кроку k можна вважати $u^{(k)}(A_h)$ незмінними (наприклад, якщо ці значення встановляться із заданим ступенем точності).

За початкові значення $u_{ij}^{(0)}$ теоретично можна вибрати будь-яку систему чисел. Однак слід мати на увазі, що для значень шуканої функції $u(x, y)$ повинні виконуватися нерівності:

$$m \leq u_{ij} \leq M.$$

$$\text{де } m = \min_{\Gamma} \varphi(P) \text{ й } M = \max_{\Gamma} \varphi(P).$$

Тому доцільно покласти $m \leq u_{ij}^{(0)} \leq M$.

3.18.3. Підготовка шаблонів

Для практичного проведення обчислень за методом ітерації корисно приготувати достатнє число спеціальних *обчислювальних шаблонів*. Спосіб приготування цих шаблонів для випадку незмінних граничних значень наступний. Нехай область G , у якій повинен бути знайдений розв'язок задачі Дирихле, покрита сіткою (рис. 3.49).

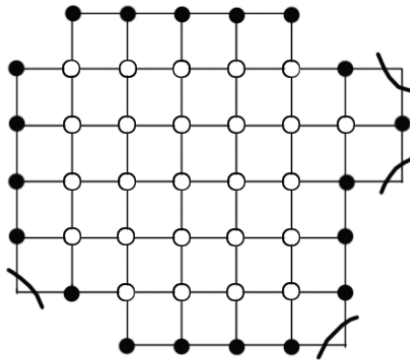


Рис. 3.49. Покриття області сіткою

Внутрішні вузли цієї сітки відмічені білими кружками; граничні вузли, у яких відомі значення шуканої функції, відмічені чорними кружками.

Для побудови обчислювального шаблону будуюмо другу сітку, лінії якої проходять посередині між лініями першої, причому так, що вузли першої сітки (внутрішні й граничні) потрапляють у центри кліток другої сітки (рис. 3.50).

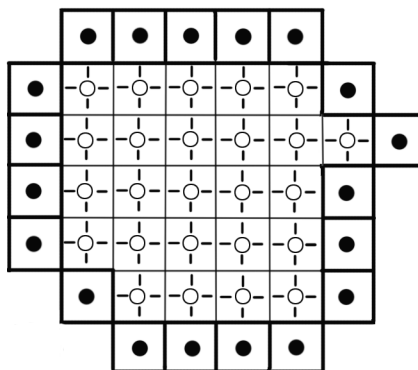


Рис. 3.50. Розташування другої сітки

Клітки другої сітки, у центрах яких лежать граничні вузли першої сітки, обведемо жирною рисою. Готовий обчислювальний шаблон № 1 наведений на рис. 3.51.

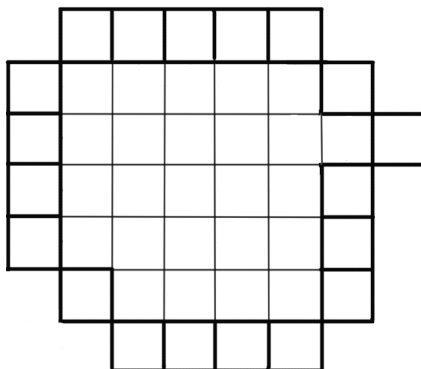


Рис. 3.51. Шаблон № 1

У обведені жирною рисою клітки шаблону № 1 записують незмінні граничні значення, визначені на контурі Γ_h . Внутрішні клітки будуть заповнюватися послідовно ітераційним процесом. Тому потрібно заготовити достатнє число однакових шаблонів № 2, № 3, ..., що складаються із одних внутрішніх кліток такого ж розміру, як клітки шаблону № 1 (рис. 3.52)

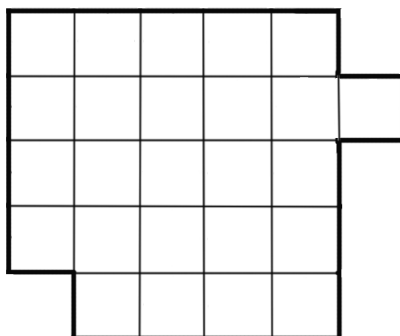


Рис. 3.52. Шаблон № 2

Внутрішні клітки шаблону № 1 заповнюємо початковими значеннями процесу ітерації (довільними числами або розв'язками задачі Дирихле, отриманими для більшої сітки). Коли шаблон № 1 заповнений, починається заповнення шаблону № 2 таким чином, щоб у кожній його клітці було записано середнє арифметичне чотирьох чисел, що розміщені у відповідних клітках шаблону № 1.

Очевидно, що при заповненні шаблону № 2 також беруть участь і значення, що розміщені у граничних клітках шаблону № 1.

Після заповнення шаблону № 2 його накладають на шаблон № 1, залишаючи при цьому відкритими граничні точки останнього, і у аналогічний спосіб заповнюють шаблон № 3, використовуючи шаблон № 2.

Процес триває доти, поки в межах заданої точності не співпадуть два останніх шаблони.

3.18.4. Приклад розв'язування крайової задачі методом сіток

Приклад 3.56. Знайти наближений розв'язок рівняння

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0,$$

який задовольняє на колі $x^2 + y^2 = 16$ (Γ) умові

$$u(x, y)|_{\Gamma} = x^2 y^2.$$

Розв'язок. У силу симетричності розв'язку розглянемо чверть кола.

1-й етап. Беремо велику сітку із кроком $h = 2$ (рис. 3.53).

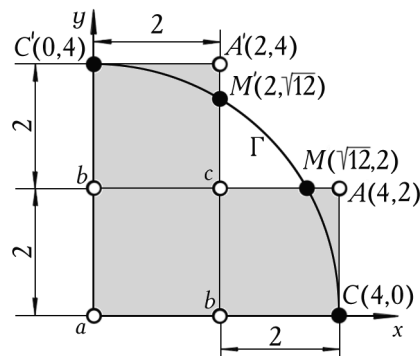


Рис. 3.53. Покриття симетричної області

Найближчою до вузла $A(4, 2)$ сітки точкою границі Γ є $M(\sqrt{12}, 2)$, тому вважаємо $u(A) \approx u(M) = 12 \cdot 2^2 = 48$.

Аналогічно для вузла сітки $A'(2, 4)$ найближчою точкою границі Γ є $M'(2, \sqrt{12})$, тому $u(A') \approx u(M') = 48$. У вузлах $C(4, 0)$ і $C'(0, 4)$ сітки очевидно маємо $u(C) = u(C') = 0$.

Позначаючи через a, b і c значення функції u у внутрішніх вузлах сітки (рис. 3.54).

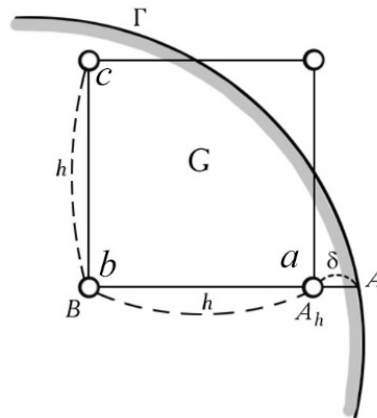


Рис. 3.54. Розташування внутрішніх вузлів a, b, c

і враховуючи симетрію задачі, складаємо систему скінченно-різницевих рівнянь

$$a = \frac{1}{4} \cdot 4b, \quad b = \frac{1}{4}(2c + a + 0), \quad c = \frac{1}{4}(48 + 48 + 2b).$$

Із цієї системи знаходимо:

$$a = 24, \quad b = 24, \quad c = 36.$$

2-й етап. Беремо більш дрібну сітку (рис. 3.55) із кроком $h = 1$ неуточнених граничних значень.

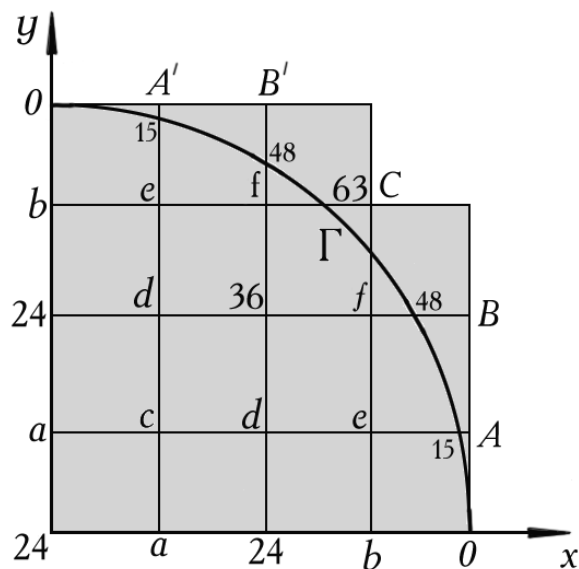


Рис. 3.55. Уточнена сітка дискретизації

Вважаємо

$$u(A) = u(A') = 15, u(B) = u(B') = 48, u(C) = 63.$$

Використовуючи значення функції $u(x, y)$ у вузлах великої сітки із кроком $h = 2$ і в граничних вузлах та симетрію задачі, складаємо скінченно-різницеві рівняння за першою і другою схемою для значень

$$a, b, c, d, e, f$$

шуканої функції u у вузлах сітки із кроком $h = 1$. Маємо:

$$\begin{cases} e = \frac{1}{4}(0 + 36 + 48 + 24), b = \frac{1}{4}(e + e + 0 + 24), \\ d = \frac{1}{4}(e + c + 24 + 36), f = \frac{1}{4}(48 + e + 63 + 36), \\ c = \frac{1}{4}(24 + 24 + 24 + 36), a = \frac{1}{4}(24 + 24 + c + c). \end{cases}$$

Звідси приблизно знаходимо: $a = 26, b = 20, c = 27, e = 27, f = 44$.

3-й етап. Уточнюємо значення $u(x, y)$ в граничних вузлах.

Використовуючи формули (3.100)

$$u^{(0)}(A_h) = u(A) = \varphi(A),$$

$$u^{(k)}(A_h) = u(A) + \frac{u^{(k-1)}(B) - u(A)}{h + \delta} \delta \quad (k = 1, 2, \dots), \quad (3.100)$$

та отримані значення у внутрішніх вузлах сітки, знаходимо:

$$u(A) = u(A') = 13, u(B) = u(B') = 49, u(C) = 73.$$

4-й етап. На основі отриманих даних будуємо систему шаблонів (№ 1–7) і послідовно уточнюємо (з точністю до одиниці) значення шуканої функції $u(x, y)$ у внутрішніх вузлах.

Шаблони № 6 і 7 збігаються з точністю до одиниці. Відзначимо, що точним розв'язком цієї задачі є функція

$$y = x^2 y^2 + \frac{1}{8} \left[256 - (x^2 + y^2)^2 \right]$$

Для порівняння приводимо значення точного розв'язку у вузлах сітки (шаблон № 7а).

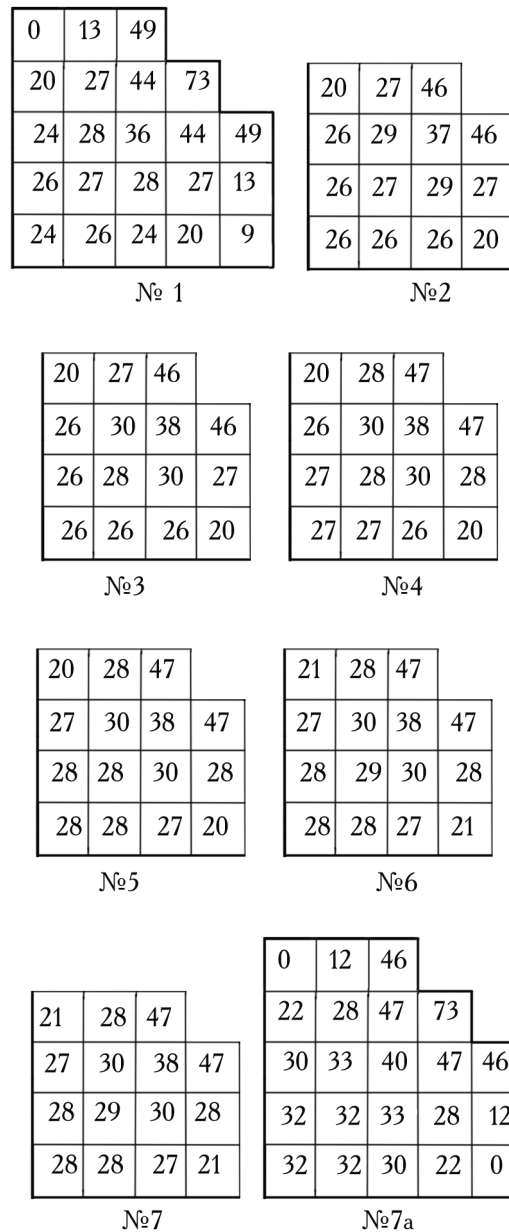


Рис. 3.56. Система шаблонів

Для оцінки точності розв'язку, отриманого за методом сіток, існують теоретичні оцінки. Як правило, ці оцінки досить складні й застосування їх утруднено. Тому на практиці використовують подвійний перерахунок розв'язку із кроками h і $2h$. Якщо відповідні результати збігаються із заданою точністю, то вважають, що шуканий розв'язок задачі знайдений

правильно. А якщо ні, то застосовують перерахування з кроком $\frac{h}{2}$ і порівнюють отриманий результат з результатом, відповідним кроку h , і т. д. Окремо слід проаналізувати вплив помилок округлення. Схема обчислень повинна бути *стійкою*, тобто помилки розв'язку, пов'язані з округленням, не повинні зростати необмежено.

Контрольні завдання

1. Дайте вичерпне формулювання задачі Дирихле.
2. Опишіть етапи розв'язування задач методом сіток.
3. Запишіть вираз скінченно-різницевого рівняння для внутрішніх вузлів.
4. Опишіть суть усереднення з використанням процесу Лібмана.
5. Види шаблонів та порядок їх розрахування.

3.19. Ітераційні асинхронні методи

3.19.1. Базові поняття

Розглядатимемо паралельні ітераційні методи для розв'язування систем лінійних рівнянь типу [81]

$$Ax = b, \quad (3.101)$$

де A – матриця розмірності $n \times n$ з додатними симетричними коефіцієнтами $a_{i,j} \in R, i, j = \overline{0, n-1}$; $x, b \in R^n$.

Позначимо через D діагональну матрицю, для якої

$$d_{ii} = a_{ii}; E = D - A$$

Сформуємо ітераційний метод $x = D^{-1}Ex + D^{-1}b$, виходячи з визначення нерухомої точки для (3.101). Позначивши $B = D^{-1}E$ та $C = D^{-1}$, одержимо

$$x = Bx + Cb. \quad (3.102)$$

Оскільки B, C і b задані, то оператор F має вигляд:

$$F(x) = Bx + Cb. \quad (3.103)$$

Будемо говорити, що матриця B є невід'ємною $B \geq 0$, якщо всі її елементи b_{ij} невід'ємні. Модуль матриці $|B| \in R^n$ також задають через модулі її елементів:

$$|B|_{ij} = |b_{ij}|, i, j = 1, 2, \dots, n. \quad (3.104)$$

Позначимо через $\rho(B)$ спектральний радіус матриці B , тобто найбільший із модулів власних значень даної матриці.

Означення. Нормований лінійний простір – це векторний простір X з нормою $\|x\|$, що задовольняє умови:

- 1) $\|x\| \geq 0 \quad \forall x \in X$,
- 2) $\|x\| = 0$ тоді і тільки тоді, коли $x = 0$,
- 3) $\|x + y\| \leq \|x\| + \|y\|$,
- 4) $\|\alpha x\| = |\alpha| \cdot \|x\|$.

Означення. Метричний простір – це пара (X, d) , яка складається з деякої множини (простору) точок X і відстані, яка однозначно задана невід'ємною дійсною функцією $d(x, y)$ і для неї виконуються такі аксіоми:

$$1) d(x, y) = 0 \text{ тоді і тільки тоді, коли } x = y, \quad (3.105)$$

$$2) d(x, y) = d(y, x) \text{ (аксіома симетрії)}, \quad (3.106)$$

$$3) d(x, z) \leq d(x, y) + d(y, z) \text{ (аксіома трикутника)}. \quad (3.107)$$

Для спрощення метричний простір (X, d) позначають однією буквою $R = (X, d)$, а функцію $d(x, y)$ називають *метрикою* простору R .

Якщо простір складається з n точок, то функція відстані для них може бути заданою, наприклад, так:

$$1) d(x, y) = \|x - y\|_1; \quad d(x, y) = \|x - y\|_1 = \sum_{i=1}^n |x_i - y_i|.$$

$$2) d(x, y) = \|x - y\|_2 = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}.$$

$$3) d(x, y) = \|x - y\|_\infty; d(x, y) = \|x - y\|_\infty = \max_{1 \leq i \leq n} |x_i - y_i|.$$

Означення. Простір R називають повним метричним простором, якщо у ньому збігається довільна фундаментальна послідовність.

Послідовність $\{x_n\}$ метричного простору R будемо називати фундаментальною, якщо вона задовольняє критерій Коші, тобто, якщо для довільного $\varepsilon > 0$ існує таке число $k \in N$, що $d(x_n, x_m) < \varepsilon \forall m, n \geq N$.

Означення. Відображення A метричного простору R у себе $A: R \rightarrow R$ називають стискаючим відображенням, якщо існує таке число $\alpha < 1$, що для довільних двох точок $x, y \in R$ виконується нерівність

$$d(Ax, Ay) \leq \alpha d(x, y). \quad (3.108)$$

Теорема 3.20. *Теорема Банаха про нерухому точку*

Нехай R – повний метричний простір і $A: R \rightarrow R$ – стискаюче відображення. Тоді це відображення має одну і тільки одну нерухому точку в R . Точку x називають нерухомою точкою відображення A , якщо $Ax = x$.

Означення. Повний нормований простір R з нормою $\|\cdot\|$ називають банаховим простором або B -простором, якщо його метрика задана виразом $d(x, y) = \|x - y\|$. Тоді $(R, \|\cdot\|)$ та $(R, \|\cdot\|)$ є повними метричними та банаховими просторами.

3.19.2. Метод хаотичних ітерацій

Означення. Метод хаотичних ітерацій (F, J) – це клас послідовностей n -вимірних векторів $x(j)$, $j = 0, 1, 2, \dots$. Кожну з послідовностей цього класу визначають рекурсивно з виразу:

$$x_i(j+1) = \begin{cases} x_i(j), & i \neq k_{n+1}(j), \\ f_i(x_1(j - k_1(j)), \dots, x_n(j - k_n(j))), & i = k_{n+1}(j), \end{cases} \quad (3.109)$$

де $x(0)$ – заданий початковий вектор.

Оператор $F : R^n \rightarrow R^n$ задають виразом

$$F(x) = (f_1(x), f_2(x), \dots, f_n(x)).$$

Послідовність $J = (k_1(j), k_2(j), \dots, k_{n+1}(j))$, $j = 0, 1, 2, \dots$ містить $(n+1)$ -вимірні вектори. При деякому заданому цілому $s > 0$ ці вектори характеризуються такими властивостями:

а) $0 \leq k_i(j) < s$ при $i = 1, 2, \dots, n$ та $j = 1, 2, \dots$;

б) $1 \leq k_{n+1}(j) \leq n$ при $j = 1, 2, \dots$;

в) $k_{n+1}(j) = i$ необмежено часто для кожного i , $1 \leq i \leq n$.

Ітераційна схема (3.109) задає процес, у якому для кожного поточного моменту часу j відбувається модифікація компонента з номером $k_{n+1}(j)$ вектора $x(j)$ за умови незмінності решти компонентів.

Для модифікації компонента $x_{k_{n+1}(j)}(j)$ використовують n компонентів, значення яких були отримані на одному з s попередніх ітераційних кроків. Затримка для кожного з таких компонентів $x_i(j - k_i(j))$ визначається часовим зсувом $k_i(j)$.

Виходячи з властивостей хаотичних ітерацій, значення компонентів, які були отримані пізніше, ніж за s кроків, не враховуються. Тому для одержання правильного результату необхідно забезпечити оновлення кожного з компонентів вектора $x(j)$. Оскільки послідовність такого оновлення може бути довільною, то очевидним стає той факт, що метод хаотичних ітерацій є узагальненням ітераційних методів із обумовленим

порядком модифікації компонентів. Задавши конкретний вид послідовності J та величину S , можемо одержати конкретний ітераційний метод.

Наприклад, при

$$k_1(j) = k_2(j) = k_3(j) = \dots = k_n(j) = 0,$$

$$k_{n+1}(j) \equiv j \pmod{n} + 1$$

і $s = 1$ одержимо $k_{n+1}(j) = i$ для $i = 1, \dots, n$. Отже, при заданих умовах кожен компонент вектора ітерацій змінюється тільки один раз за n ітерацій, що відповідає методу Гауса-Зейделя.

Якщо виразити послідовність J співвідношеннями

$$k_{n+1}(j) \equiv j \pmod{n} + 1, k_1(j) = k_2(j) = \dots = k_n(j) = \left[k_{n+1}(j) + 1 \right],$$

а значення затримки покласти $S = n$, то також одержимо $k_{n+1}(j) = i$ для $i = 1, \dots, n$. Але у цьому випадку ітераційна схема відповідає методу Якобі.

Умову збіжності методу хаотичних ітерацій задають наступною теоремою.

Теорема 3.21. Нехай (F, J) – метод хаотичних ітерацій, побудований за ітераційною схемою (3.109). Для того, щоб послідовність ітерацій над вектором $x(j)$, виконаних за методом хаотичних ітерацій (F, J) , збігалася до розв'язку рівняння (3.102), необхідно і достатньо, щоб спектральний радіус матриці $|B|$ був меншим за одиницю.

При доведенні умов необхідності та достатності цієї теореми застосовують лему 3.1.

Лема 3.1. Нехай матриця M є невід'ємною з елементами $\{m_{ij}\}_{i,j=1}^n$.

Для того, щоб спектральний радіус даної матриці був меншим за одиницю,

необхідно і достатньо, щоб існував додатний вектор $v \in R^n$ і деяке дійсне число θ , $0 < \theta < 1$, які б задовольняли нерівність

$$Mv \leq \theta v. \quad (3.110)$$

Доведення умови достатності теореми 3.21. випливає з леми 1. Оскільки спектральний радіус $\rho(B) < 1$, то існують додатний вектор $v \in R^n$ і додатне число $\theta < 1$ такі, що

$$|B|v \leq \theta v. \quad (3.111)$$

Нехай $x \in R^n$ та $|x| < v$. Тоді справедлива нерівність

$$|Bx| \leq |B||x| \leq |B|v \leq \theta v, \quad (3.112)$$

яка підтверджує те, що матриця $|B|$ є матрицею стискування і переводить паралелепіпед $\{x : |x| \leq v\}$ у себе. Використовуючи цей факт, доводять, що послідовність $\{x(j)\}_{j=1}^{\infty}$ прямує до нуля.

При доведенні умови необхідності теореми 3.21 показують, що якщо не існує такого v , для якого виконується нерівність

$$|B|v \leq \alpha v, \quad (3.113)$$

то існує така послідовність J_0 , для якої ітерації $\{x(j)\}_{j=1}^{\infty}$ розбігаються. Тоді з леми 1 випливає, що для даного випадку $\rho(|B|) \geq 1$. Отже, теорема 3.21 повністю доведена.

Розглянемо застосування методу хаотичних ітерацій для релаксації з параметром ω . Ітераційна схема для даного випадку має вигляд:

$$x = w(Bx + C) + (1 - w)x. \quad (3.114)$$

Звівши подібні члени, одержимо

$$x = (I - \omega D^{-1}A)x + \omega Cb. \quad (3.115)$$

Введемо матриці $B^\omega = I - \omega D^{-1}A$ та $C^\omega = \omega C$. Позначивши через (F^ω, J) метод хаотичної релаксації, наведемо умови його збіжності.

Теорема 3.22. Якщо метод хаотичної релаксації (F^ω, J) побудований за ітераційною схемою (3.109), то необхідні і достатні умови його збіжності:

$$\rho(|B|) < 1, \quad 0 < \omega < \frac{2}{1 + r(|B|)}.$$

Загальний випадок методу хаотичної ітерації маємо, коли більше, ніж один компонент вектора $x(j)$ може бути оновленим у кожний момент часу j , і оператор F є нелінійним.

Нехай $\Psi = \{J_j\}_{j=1}^\infty$ – послідовність непустих підмножин $J_j \subset \{1, 2, \dots, n\}$. Тоді ітераційну схему (3.109) перепишемо у вигляді:

$$x_i(j+1) = \begin{cases} x_i(j), & i \notin J_j, \\ f_i(x_1(j-k_1(j)), \dots, x_n(j-k_n(j))), & i \in J_j. \end{cases} \quad (3.116)$$

Властивості методу

1) Оператор $F: R^n \rightarrow R^n$, заданий у покоординатному вигляді $F(x) = (f_1(x), f_2(x), \dots, f_n(x))$, може бути нелінійним;

2) $i \in \Psi$ нескінченну кількість разів;

3) $x_i(j)$ підлягає модифікації хоча б один раз протягом

S послідовних кроків.

Узагальнений метод хаотичних ітерацій задає обчислювальний процес, у якому на ітераційному кроці j всі компоненти вектора $x(j)$, номери яких входять у підмножину J_j , підлягають модифікації, а всі інші компоненти цього вектора залишаються незмінними. Модифікація

компонентів $x_i(j)$ з номерами $i \in J_j$ відбувається одночасно. Для цього використовують n компонентів, значення яких були одержані на одному з S попередніх ітераційних кроків. Затримку для кожного з таких компонентів визначають часовим зсувом $x_i(j - k_i(j))$, що має значення $k_i(j)$. Як і в попередньому випадку, значення компонентів, які були отримані пізніше, ніж за S кроків, не враховуються.

Означення. Нехай задано відображення $F : D(F) \rightarrow B$, при $D(F) \subset B$ і $B = \prod_{i=1}^n B_i$, де $\{B_i\}_{i=1}^n$ – множина банахових просторів. Для кожної даної точки $v \in D(F)$ говорять, що оператор F є стискаючим у точку U по векторній нормі η , якщо:

1) існує невід’ємна матриця T розмірності $n \times n$, яка відповідає умові

$$\eta(F(v) - F(v)) \leq T\eta(v - v) \quad \forall v \in D(F), \quad (3.117)$$

2) спектральний радіус матриці T менший за одиницю: $\rho(T) < 1$.

Для $B_i \in R$, $1 \leq i \leq n$, векторна норма має вигляд:

$$\eta(x) = (\|x_1\|, \dots, \|x_n\|) \quad \forall x \in B = R^n.$$

Враховуючи це, перепишемо нерівність (3.118):

$$|F(v) - F(v)| \leq T|v - v| \quad \forall v \in R^n. \quad (3.118)$$

Теорема про збіжність даного методу хаотичних ітерацій спирається на (3.118).

Теорема 3.23. Нехай оператор F має фіксовану точку $x \in D(F)$ і стискає в x для векторної норми η . Тоді ітераційна послідовність

$\{x(j)\}_{j=1}^{\infty}$, задана ітераційною схемою (3.116), збігається в нерухому точку x .

Висновок. Нехай оператор $F : R^n \rightarrow R^n$ має форму $F(x) = B(x) + C$. Якщо $\rho(|B|) < 1$, то F має єдину нерухому точку x і стискає в цю точку для векторної норми η .

Доведення. Розглянемо нормований простір $(R^n, \|\cdot\|_{\omega})$ з векторною нормою $\|x\|_{\omega} = \max_{i=1,n} \left| \frac{x_i}{\omega_i} \right|$, $\omega_i > 0$. (3.119)

Відомо, що такий простір є повним метричним простором. Тому для доведення того, що оператор F має єдину нерухому точку x , достатньо показати, що він є стискаючим на R^n по нормі $\|\cdot\|_{\omega}$.

Нехай $x, y \in R^n$. Тоді

$$\eta(F(x), F(y)) = |B(x-y)| \leq |B||x-y| = B\eta(x-y) \quad (3.120)$$

Можна показати, що оскільки матриця $|B|$ – невід’ємна і $\rho(|B|) < 1$, то за умови $\alpha < 1$ справедлива нерівність:

$$\|F(x) - F(y)\|_{\omega} \leq \alpha \|x - y\|_{\omega} \quad (3.121)$$

Нехай відстань $d(x, y) = \|x - y\|_{\omega}$. Тоді за теоремою Банаха про нерухому точку для оператора F існує єдина нерухома точка $x \in R^n$. Оскільки F – стискаючий оператор на R^n , то він стискає в $x \in R^n$ по векторній нормі η .

3.19.3. Метод асинхронних ітерацій

Основна відмінність даного методу від розглянутих методів хаотичних ітерацій полягає в знятті обмеження на глибину ітераційних

кроків S , що використовують при обчисленні ітераційної послідовності $\{x(j)\}_{j=1}^{\infty}$. При цьому оператор F може бути як лінійним, так і нелінійним.

Означення. Нехай $F(x) = (f_1(x), f_2(x), \dots, f_n(x))$ – оператор, що задає відображення $F: R^n \rightarrow R^n$. Тоді асинхронною ітераційною послідовністю, яка відповідає оператору F , називають послідовність $\{x(j)\}_{j=1}^{\infty}$ векторів $x(j) \in R^n$, що визначається рекурсивно за ітераційною схемою [82]:

$$x_i(j+1) = \begin{cases} x_i(j), & i \notin J_j, \\ f_i[x_1(s_1(j)), x_2(s_2(j)), \dots, x_n(s_n(j))], & i \in J_j, \end{cases} \quad (3.122)$$

де $\Psi = \{J_j\}_{j=1}^{\infty}$ – послідовність непустих підмножин, $J_j \subset \{1, 2, \dots, n\}$;

$S = \{s_i(j)\}_{j=1}^{\infty}$, $i = 1, 2, \dots, n$ – послідовність цілих невід'ємних чисел, що

відповідають умовам: $0 \leq s_i(j) \leq j$ при $j = 1, 2, \dots$; $\lim_{j \rightarrow \infty} (s_i(j)) = \infty$;

$i \in \Psi$ нескінченну кількість разів.

Асинхронний ітераційний метод, який відповідає оператору F з початковим вектором $x(0)$ та множинам Ψ і S , позначають $(F, x(0), \Psi, S)$.

Виходячи з визначення хаотичного ітераційного методу, існує деяке τ , для якого справедлива нерівність $s_i(j) \geq j - \tau \geq 0$ при $j = 1, 2, \dots$, $i = 1, 2, \dots$. Ця умова випливає з того, що $\lim_{j \rightarrow \infty} (s_i(j)) = \infty$. Отже, метод асинхронних ітерацій можна розглядати як узагальнення методів хаотичних ітерацій.

Наведемо означення стискаючого оператора F [35].

Означення. Оператор F , що задає відображення $F : R^n \rightarrow R^n$, є стискаючим оператором на множині $D \subset R^n$ за умови, що:

– існує невід’ємна матриця T розмірності $n \times n$, яка відповідає умові:

$$\left| F(x) - F(y) \right| \leq T |x - y| \quad \forall x, y \in D \quad (3.123)$$

– спектральний радіус матриці T менший за одиницю: $\rho(T) < 1$.

Доведена теорема, що задає умови збіжності методу асинхронних ітерацій.

Теорема 3.24 Якщо F є стискаючим оператором на замкнутій підмножині $D \subset R^n$ і $F(D) \subset D$, то асинхронний ітераційний метод $(F, x(0), \Psi, S)$, що відповідає оператору F з початковим вектором $x(0)$ на підмножині D , збігається до нерухомої точки оператора F з підмножини D .

Оскільки умовою збіжності асинхронного ітераційного методу $(F, x(0), \Psi, S)$ є наявність властивості стискання для оператора F у всіх точках замкнутої підмножини, то таку умову вважають більш жорсткою, ніж умова збіжності методу хаотичних ітерацій (F, J) , яка впливає з властивості стискання оператора F тільки в його нерухомій точці.

3.19.4. Метод асинхронних ітерацій з нерухомими точками

Ці методи можна розглядати як подальше узагальнення попередніх методів.

Означення. Нехай задано відображення $F : R^n \rightarrow R^n$. Тоді асинхронною ітераційною послідовністю, яка відповідає оператору F ,

будемо називати послідовність $\{x(t)\}_{t=1}^{\infty}$ векторів $x(t) \in R^n$, що визначається рекурсивно за ітераційною схемою:

$$x_i(t+1) = \begin{cases} x_i(t) & \forall t \notin T^i, \\ f_i \left[x_1(s_1^i(t)), x_2(s_2^i(t)), \dots, x_n(s_n^i(t)) \right] & \forall t \in T^i, \end{cases} \quad (3.124)$$

де $S^i = \left\{ s_k^i(t) \right\}_{t=1}^{\infty}$, $i, k = 1, 2, \dots, n$, – множини цілих невід'ємних

чисел, що відповідають умовам: $0 \leq s_k^i(t) \leq t$, $\forall t \geq 0$.

На відміну від асинхронного ітераційного методу (3.122), у якому всім компонентам $f_i(x)$ оператора F відповідає одна і та ж множина S , для даного методу з кожним компонентом $f_i(x)$ співставляється своя множина S^i . З цієї причини вибір конкретної множини $S = S^i$, $i = 1, \dots, n$, та T^i зводить ітераційну схему (3.124) до одного з розглянутих раніше випадків, які називають сценаріями.

Розрізняють ітераційні методи з повною та частковою асинхронністю.

Означення. Ітераційним методом з повною асинхронністю називають метод, що базується на ітераційній послідовності (3.124) за умов:

$$1) t_l \in T^i, i = 1, 2, \dots, n, \lim_{l \rightarrow \infty} t_l \rightarrow \infty ;$$

$$2) \lim_{k \rightarrow \infty} s_k^i(t_l) \rightarrow \infty .$$

Означення. Ітераційним методом з частковою асинхронністю називають метод, що базується на ітераційній послідовності (3.124), якщо існує міра асинхронності, яка визначається константою τ , і справджуються наступні залежності:

1) для кожного $t \geq 0$ і для кожного $i = 1, 2, \dots, n$ існує хоча б один елемент множини $\{t, t+1, t+2, \dots, t+\tau-1\}$, який належить до T^i ;

2) для елементів множин S^i справджується нерівність

$$t - \tau < s_k^i(t) \leq t, \quad \forall t \in T^i;$$

3) $s_i^i(t) = t, \quad \forall t \in T^i, \quad i = 1, 2, \dots, n.$

Умови достатності для збіжності ітераційного методу з повною асинхронністю задають теоремою:

Теорема 3.25. Нехай $X = \prod_{i=1}^n X_i$ та для кожного $i \in \{1, 2, \dots, n\}$ існує

послідовність $\{X_i(k)\}$ підмножин множини X_i , для якої виконуються умови:

1) $X(k) = X_1(k) \times X_2(k) \times \dots \times X_n(k)$ – умова існування блоків;

2) $X_i(k+1) \subseteq X_i(k)$ – умова існування вкладених множин;

3) $F(X(k)) \subseteq X(k+1)$ – умова існування вкладених відображень множин;

4) для $x(k) \in X(k), k \in N, \lim_{k \rightarrow \infty} x(k) \rightarrow x$, тобто кожна гранична точка

послідовності $\{x(k)\}$ є нерухомою точкою оператора F – умова синхронної збіжності.

Тоді при $x(0) \in X(0)$ кожна гранична точка послідовності $\{x(k)\}$, яка одержана шляхом застосування ітераційної схеми (3.124), є нерухомою точкою оператора F .

Згідно з означенням ітераційного методу з частковою асинхронністю кожне наступне значення вектора $x(t+1)$ формується на основі обмеженої підмножини попередніх значень $z(t) = \{x(t), x(t-1), \dots, x(t-\tau+1)\}$, кількість елементів якої залежить від міри асинхронності τ . Підмножини

$z(t)$ формують множину Z при $t \geq 0$. Позначимо через X° всі можливі нерухомі точки $\left\{ x \in X \mid x = f(x) \right\}$, а через Z множину всіх можливих елементів Z , що мають вигляд $\left\{ x^\circ, \dots, x^\circ \right\}$, де $x^\circ \in X^\circ$.

Теорема 3.26. *Теорема Ляпунова*

Нехай F – неперервний оператор в умовах часткової асинхронності. Припустимо, що існує додатне ціле t' і неперервна функція $d : Z \mapsto [0, \infty)$ з властивостями:

- 1) для кожного $z(0) \notin Z^\circ$ і для кожного сценарію справедлива нерівність $d(z(t')) < d(z(0))$;
- 2) для кожного $z(t) \in Z$, для кожного $t \geq 0$ і для кожного сценарію справедлива нерівність $d(z(t+1)) < d(z(t))$.

Тоді $z^\circ \in Z^\circ$ для кожної граничної точки послідовності $\{z(t)\}$.

Доведення теореми 3.25 та теореми 3.26 базуються на ідеї вкладених множин. Нехай всі компоненти на початку ітераційного процесу належать до $X(k)$. Після кількох ітерацій, відповідно до умови теореми, отримаємо $x_i \in X_i(l)$, $l > k$. Всі отримані значення задовольняють умову: $X_i(l) \subset X(k+1)$.

Висновок. Замінімо в теоремі 3.25 глобальну умову 3 на локальну $f_i(X(k)) \subseteq X(k+1)$. Тоді асинхронний нестационарний ітераційний метод, що базується на ітераційній схемі (3.124), збігатиметься до єдиної нерухомої точки x° , що є спільною для всіх f_i .

Існують теореми, що дають строге доведення твердження, наведеного у висновку, для конкретних випадків. Одним із них є випадок, коли кожний

з компонентів X_i простору X є нормованим лінійним простором $(X_i, \|\cdot\|_i)$.

Визначимо норму $\|\cdot\|_\omega$ за формулою:

$$\|x\|_\omega = \max_{i=1,n} \frac{\|x\|_i}{\omega_i}, \quad (3.125)$$

де $\omega_i > 0$, $i = 1, 2, \dots, n$.

Теорема 3.27. Нехай існує нерухома точка $f_i(x^\circ) = x^\circ$ для $x^\circ \in X$, $i = 1, \dots, n$, і деяке таке число $\alpha \in [0, 1)$, що виконуються нерівності

$$\|f_i(x) - x^\circ\|_\omega \leq \alpha \|x - x^\circ\|_\omega. \quad (3.126)$$

Тоді асинхронний нестационарний ітераційний метод, що базується на ітераційній схемі (3.124), збігатиметься до єдиної нерухомої точки x° , яка є спільною для всіх f_i .

Ще один випадок, для якого знайдено строге доведення збіжності ітераційного методу з повною асинхронністю, базується на застосуванні операторів з властивістю парастискання.

Означення. Оператор $F : R^n \rightarrow R^n$ називають оператором з властивістю парастискання для нерухомої точки x° , якщо існує така невід'ємна матриця $P \in R^{n \times n}$ зі спектральним радіусом $\rho(P) < 1$, що для всіх $x \in X$ справедлива нерівність

$$\begin{pmatrix} \|(Fx)_1 - x_1^\circ\|_1 \\ \dots \\ \|(Fx)_n - x_n^\circ\|_n \end{pmatrix} \leq P \cdot \begin{pmatrix} \|x_1 - x_1^\circ\|_1 \\ \dots \\ \|x_n - x_n^\circ\|_n \end{pmatrix}, \quad (3.127)$$

яка є покомпонентною нерівністю в просторі R^n [149].

Теорема 3.28. Нехай f_i має властивість парастискання по відношенню до нерухомої точки x° з матрицею P незалежно від i . Тоді асинхронний нестационарний метод, що базується на ітераційній схемі (3.124), збігається до нерухомої точки x° , яка є єдиною для всіх f_i .

Ця теорема була вперше доведена для асинхронного ітераційного методу з ітераційною послідовністю (3.122), а потім поширена на ітераційну послідовність (3.124).

Умова стискання (3.126) може бути ослабленою при використанні M -функцій та діагонально-домінантних функцій. Якщо умова стискання

$$\|F(x) - x^\circ\|_\omega < \|x - x^\circ\|_\omega \quad (3.128)$$

при $x \neq x^\circ$ є справедливою в обох напрямках, то таке відображення називають *парастисканням*.

Контрольні завдання

1. Дайте визначення теореми Банаха про нерухому точку.
2. Ітераційна схема та умови збіжності методу хаотичних ітерацій.
3. У чому полягає відмінність методу асинхронних ітерацій від методу асинхронних ітерацій з нерухомими точками.
4. Умови збіжності асинхронних методів.
5. Визначення теореми Ляпунова.

3.20. Метод скінченних об'ємів

3.20.1. Базові відомості

Основна ідея методу скінченних об'ємів (МСО), названого також методом контрольних об'ємів, полягає в наступному.

Розрахункова область розбивається на елементарні об'єми, і диференціальне рівняння в крайовій задачі заміняється інтегральними балансними співвідношеннями для кожного із цих елементарних об'ємів.

Після цього інтеграли в інтегробалансних співвідношеннях апроксимуються з використанням значень шуканої функції у вузлах сітки або значень похідних шуканої функцій, узятих із крайових умов. У результаті отримуємо систему лінійних алгебраїчних рівнянь (СЛАР), розв'язком якої є вектор значень шуканої функції у вузлах сітки.

Техніка апроксимації при використанні МСО для розв'язування багатовимірних (двовимірних і тривимірних) задач суттєво залежить від того, на комірки якої форми розбивається розрахункова область. При цьому скінченні об'єми, у яких інтегробалансні співвідношення замінюються своїми дискретними аналогами, не є комірками дискретизації розрахункової області. Комірки дискретизації розрахункової області характеризуються тим, що вузли сітки, у яких будуть обчислюватися значення розв'язку u початкової крайової задачі, є, як правило, вершинами цих комірок (для більш складних апроксимацій ця умова може виконуватися не для всіх вузлів, але в будь-якому випадку повинна виконуватися інша, більш складна умова: інтерполюючі поліноми шуканої функції u , побудовані по її вузлових значеннях, визначаються на комірках дискретизації розрахункової області, а не на самих скінченних об'ємах). Кожний же з скінченних об'ємів будується навколо вузлів сітки з окремих частин комірок дискретизації розрахункової області, що примикають до відповідних вузлів, причому форма скінченного об'єму може суттєво відрізнятися від форми комірок дискретизації розрахункової області.

Розглянемо основні аспекти техніки виконання скінченно-об'ємних апроксимацій на прикладі розв'язування одновимірної еліптичної крайової задачі.

3.20.2. Розв'язування одновимірної еліптичної задачі методом МСО

Побудуємо дискретний аналог еліптичної крайової задачі, використовуючи її скінченно-об'ємну апроксимацію рівняння:

$$-\frac{d}{dx}\left(p\frac{du}{dx}\right) + \gamma u = f \quad (3.129)$$

Рівняння задано в деякій області G із границею $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$

$$u\Big|_{\Gamma_1} = u_g \quad (3.130)$$

$$p\frac{\partial u}{\partial n}\Big|_{\Gamma_2} = \theta, \quad (3.131)$$

$$p\frac{\partial u}{\partial n}\Big|_{\Gamma_3} + \beta\left(u\Big|_{\Gamma_3} - u_\beta\right) = 0 \quad (3.132)$$

де $u\Big|_{\Gamma_i}$ – значення функції u на границі Γ_i ,

$\frac{\partial u}{\partial n}\Big|_{\Gamma_i}$ – значення похідної функції u по напрямку зовнішньої нормалі

до поверхні Γ_i .

Крайова задача (3.129–3.132) може описувати стаціонарний розподіл температури u в області G , якщо p – коефіцієнт теплопровідності, а $(f - \gamma u)$ – густина об'ємних джерел тепла. При цьому крайова умова першого роду (3.130) відповідає задаванню на частині Γ_1 границі Γ області G температури u_g , крайова умова другого роду (3.131) – задаванню на Γ_2 густини теплового потоку, рівного θ , а крайова умова третього роду (3.132) – задаванню на Γ_3 густини теплового потоку, пропорційній різниці температури u на границі Γ_3 й температури u_β навколишнього середовища (коефіцієнт теплопровідності $\beta > 0$ у цьому випадку називають коефіцієнтом тепловіддачі).

Областю визначення G в одновимірному випадку є інтервал (a, b) , а границею Γ , на якій задано крайові умови (3.130–3.132), або точка a , або точка b , або порожня множина (якщо відповідні крайові умови відсутні в крайовій задачі).

Нанесемо на інтервал (a, b) сітку з вузлами x_k , що задовольняють співвідношенням: $a = x_1 < x_2 < x_3 < \dots < x_k < \dots < x_{n-1} < x_n = b$. Вузли x_k розбивають розрахункову область G на $(n - 1)$ комірку

$$G_k = (x_k, x_{k+1}), \quad k = 1, \dots, n - 1.$$

Позначимо через G'_k ліву половину $\left(x_k, \frac{x_k + x_{k+1}}{2}\right)$ комірки G_k , а через G''_k – праву половину $\left(\frac{x_k + x_{k+1}}{2}, x_{k+1}\right)$ комірки G_k . Навколо кожного внутрішнього вузла x_k ($k = 2, \dots, n - 1$) побудуємо скінченний об'єм \tilde{G}_k , що складається із двох половинок, що примикають до x_k комірок дискретизації $\tilde{G}_k = G''_{k-1} \cup G'_k$. Відповідні ж граничним вузлам x_1 і x_n скінченні об'єми \tilde{G}_1 та \tilde{G}_n будуть складатися з однієї з половинок комірок G_1 і G_{n-1} : $\tilde{G}_1 = G'_1$, $\tilde{G}_n = G''_{n-1}$.

Розбивка розрахункової області $G = (a, b)$ на комірки дискретизації G_k й скінченні об'єми \tilde{G}_k показана на рис. 3.57.

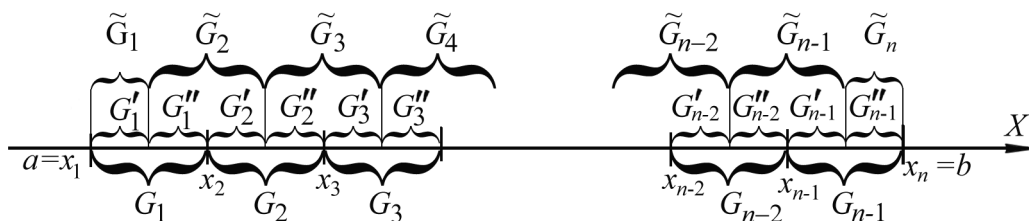


Рис. 3.57. Розбивка розрахункової області $G = (a, b)$ на комірки

дискретизації G_k й скінченні об'єми \tilde{G}_k

Проінтегруємо рівняння (1) по кожному з скінченних об'ємів \tilde{G}_k .

У результаті одержимо n рівнянь, які мають такий вигляд:

$$-\int_{\tilde{G}_k} \frac{d}{dx} p \frac{du}{dx} dG + \int_{\tilde{G}_k} \gamma u dG = \int_{\tilde{G}_k} f dG, \quad (3.133)$$

Враховуючи, що для внутрішніх вузлів x_k ($k = 2, 3, \dots, n-1$) скінченні об'єми, які їх утримують, мають вигляд

$$\tilde{G}_k = G''_{k-1} \cup G'_k = \left(\frac{x_{k-1} + x_k}{2}, \frac{x_k + x_{k+1}}{2} \right), \text{ для граничних вузлів } \tilde{G}_1 = \left(x_1, \frac{x_1 + x_2}{2} \right)$$

(для x_1) та $\tilde{G}_n = \left(\frac{x_{n-1} + x_n}{2}, x_n \right)$ (для x_n), систему (3.133) можна записати

у вигляді.

$$-\int_{x_1}^{\frac{x_1+x_2}{2}} \frac{d}{dx} \left(p \frac{du}{dx} \right) dx + \int_{x_1}^{\frac{x_1+x_2}{2}} \gamma u dx = \int_{x_1}^{\frac{x_1+x_2}{2}} f dx, \quad (3.134)$$

$$-\int_{x_{k-1/2}}^{x_{k+1/2}} \frac{d}{dx} \left(p \frac{du}{dx} \right) dx + \int_{x_{k-1/2}}^{x_{k+1/2}} \gamma u dx = \int_{x_{k-1/2}}^{x_{k+1/2}} f dx, \quad (k = 2, \dots, n-1), \quad (3.135)$$

$$-\int_{x_{n-1/2}}^{x_n} \frac{d}{dx} \left(p \frac{du}{dx} \right) dx + \int_{x_{n-1/2}}^{x_n} \gamma u dx = \int_{x_{n-1/2}}^{x_n} f dx, \quad (3.136)$$

де під $x_{k-1/2}$ розуміється точка $\frac{x_{k-1} + x_k}{2}$, а під $x_{k+1/2}$ — відповідно точка

$$\frac{x_k + x_{k+1}}{2}.$$

Значення інтегралів у перших доданків лівих частин рівнянь (3.134–3.136) очевидні – це різниця значень величини $p \frac{du}{dx}$ на границях інтервалу інтегрування:

$$\int_{x_1}^{\frac{x_1+x_2}{2}} \frac{d}{dx} \left(p \frac{du}{dx} \right) dx = p \frac{du}{dx} \Big|_{\frac{x_1+x_2}{2}} - p \frac{du}{dx} \Big|_{x_1}, \quad (3.137)$$

$$\int_{x_{k-1/2}}^{x_{k+1/2}} \frac{d}{dx} \left(p \frac{du}{dx} \right) dx = p \frac{du}{dx} \Big|_{x_{k+1/2}} - p \frac{du}{dx} \Big|_{x_{k-1/2}}, \quad k = 2, \dots, n-1, \quad (3.138)$$

$$\int_{x_{n-1/2}}^{x_n} \frac{d}{dx} \left(p \frac{du}{dx} \right) dx = p \frac{du}{dx} \Big|_{x_n} - p \frac{du}{dx} \Big|_{x_{n-1/2}}. \quad (3.139)$$

Щоб обчислити значення $p \frac{du}{dx} \Big|_{x_{k\pm 1/2}}$ у правих частинах співвідношень

(3.137–3.139), а також значення інтегралів від γu та f у рівняннях (3.134–3.136), застосуємо наступні припущення про поведінку наближеного розв'язку u , коефіцієнту γ та функції f на комірках дискретизації $G_k = (x_k, x_{k+1})$. Будемо вважати, що функція u лінійно змінюється на G_k від значення u_k у точці $x = x_k$ до значення u_{k+1} у точці $x = x_{k+1}$. Аналогічно й функція f лінійно змінюється на G_k від значення f_k у точці $x = x_k$ до значення f_{k+1} у точці $x = x_{k+1}$. Коефіцієнт γ будемо вважати на G_k постійним і рівним значенню $\gamma_{k+1/2}$ (тобто таким, що дорівнює точному значенню коефіцієнта γ у точці $x_{k+1/2} = \frac{x_k + x_{k+1}}{2}$). Позначимо через h_k довжину комірки G_k : $h_k = x_{k+1} - x_k$. Тоді

$$p \frac{du}{dx} \Big|_{x_{k-1/2}} = p_{k-1/2} \frac{u_k - u_{k-1}}{h_{k-1}}, \quad p \frac{du}{dx} \Big|_{x_{k+1/2}} = p_{k+1/2} \frac{u_{k+1} - u_k}{h_k}, \quad (3.140)$$

$$\int_{x_{k-1/2}}^{x_k} \gamma u dx = \gamma_{k-1/2} \frac{1}{2} \left(\frac{u_{k-1} + u_k}{2} + u_k \right) \frac{h_{k-1}}{2} = \frac{\gamma_{k-1/2} h_{k-1}}{8} (3u_k + u_{k-1}), \quad (3.141)$$

$$\int_{x_k}^{x_{k+1/2}} \gamma u dx = \gamma_{k+1/2} \frac{1}{2} \left(\frac{u_k + u_{k+1}}{2} + u_k \right) \frac{h_k}{2} = \frac{\gamma_{k+1/2} h_k}{8} (3u_k + u_{k+1}), \quad (3.142)$$

$$\int_{x_{k-1/2}}^{x_k} f dx = \frac{1}{2} \left(\frac{f_{k-1} + f_k}{2} + f_k \right) \frac{h_{k-1}}{2} = \frac{h_{k-1}}{8} (3f_k + f_{k-1}), \quad (3.143)$$

$$\int_{x_k}^{x_{k+1/2}} f dx = \frac{1}{2} \left(\frac{f_k + f_{k+1}}{2} + f_k \right) \frac{h_k}{2} = \frac{h_k}{8} (3f_k + f_{k+1}) \quad (3.144)$$

При обчисленні інтегралів у співвідношеннях (3.141–3.144) ми використовували той факт, що визначений інтеграл від лінійної на інтервалі інтегрування функції дорівнює напівсумі значень функції на кінцях інтервалу, помноженій на довжину інтервалу інтегрування (тобто площі відповідної трапеції).

Підставляючи співвідношення (3.137–3.139) і (3.141–3.144) у систему (3.134–3.136) та враховуючи співвідношення (3.140), одержимо

$$-\left(p_{1+1/2} \frac{u_2 - u_1}{h_1} - p \frac{du}{dx} \Big|_{x_1} \right) + \frac{\gamma_{1+1/2} h_1}{8} (3u_1 + u_2) = \frac{h_1}{8} (3f_1 + f_2), \quad (3.145)$$

$$-\left(p_{k+1/2} \frac{u_{k+1} - u_k}{h_k} - p_{k-1/2} \frac{u_k - u_{k-1}}{h_{k-1}} \right) + \frac{\gamma_{k-1/2} h_{k-1}}{8} (3u_k + u_{k-1}) + \frac{\gamma_{k+1/2} h_k}{8} (3u_k + u_{k+1}) = \frac{h_{k-1}}{8} (3f_k + f_{k-1}) + \frac{h_k}{8} (3f_k + f_{k+1}), \quad (3.146)$$

$$-\left(p \frac{du}{dx} \Big|_{x_n} - p_{n-1/2} \frac{u_n - u_{n-1}}{h_{n-1}} \right) + \frac{\gamma_{n-1/2} h_{n-1}}{8} (3u_n + u_{n-1}) = \frac{h_{n-1}}{8} (3f_n + f_{n-1}) \quad (3.147)$$

Якщо на границях області G задані крайові умови другого або третього роду, доданки, що входять у рівняння (3.145) та (3.147) $p \frac{du}{dx} \Big|_{x_1}$ й $p \frac{du}{dx} \Big|_{x_n}$ можуть бути отримані з них з врахуванням того, що в точці x_1 похідна по нормалі $\frac{du}{dn} = -\frac{du}{dx}$, а в точці x_n ця похідна $\frac{du}{dn} = \frac{du}{dx}$ (тобто на лівій границі G напрямок зовнішньої нормалі n протилежний напрямку x , а на правій – збігається з напрямком осі x). Для визначеності будемо вважати, що на лівому кінці G задана крайова умова другого роду (3.131), тобто Γ_2 – це точка x_1 , а на правому – крайова умова третього роду (3.132) (тобто Γ_3 – це точка x_n).

Тоді в рівнянні (3.145) доданок $p \frac{du}{dx} \Big|_{x_1}$ може бути замінений на $-\theta$, (оскільки $p \frac{du}{dx} \Big|_{x_1} = -p \frac{du}{dn} \Big|_{\Gamma_2}$ й з (3.131) $-p \frac{du}{dn} \Big|_{\Gamma_2} = -\theta$), а в рівнянні (3.147) доданок $p \frac{du}{dx} \Big|_{x_n}$ може бути замінений на $-\beta(u_n - u_\beta)$ (оскільки $p \frac{du}{dx} \Big|_{x_n} = p \frac{du}{dn} \Big|_{\Gamma_3}$ й з (3.132) $p \frac{du}{dn} \Big|_{\Gamma_3} = -\beta(u \Big|_{\Gamma_3} - u_\beta)$, причому $u \Big|_{\Gamma_3} = u_n$).

Враховуючи це й групуючи в рівняннях (3.145–3.147) члени, що містять ті самі невідомі з u_j , одержимо

$$\left(\frac{p_{1+1/2}}{h_1} + \frac{3\gamma_{1+1/2}h_1}{8} \right) u_1 + \left(\frac{p_{1+1/2}}{h_1} + \frac{3\gamma_{1+1/2}h_1}{8} \right) u_2 = \frac{h_1}{8} (3f_1 + f_2) + \theta, \quad (3.148)$$

$$\begin{aligned} & \left(-\frac{p_{k-1/2}}{h_{k-1}} + \frac{\gamma_{k-1/2} h_{k-1}}{8} \right) u_{k-1} + \left(\frac{p_{k-1/2}}{h_{k-1}} + \frac{p_{k+1/2}}{h_k} + \frac{3}{8} (\gamma_{k-1/2} h_{k-1} + \gamma_{k+1/2} h_k) \right) u_k + \\ & + \left(-\frac{p_{k+1/2}}{h_k} + \frac{\gamma_{k+1/2} h_k}{8} \right) u_{k+1} = \frac{h_{k-1}}{8} (3f_k + f_{k-1}) + \frac{h_k}{8} (3f_k + f_{k+1}) \end{aligned} \quad (3.149)$$

$$k = 2, \dots, n-1.$$

$$\begin{aligned} & \left(-\frac{p_{n-1/2}}{h_{n-1}} + \frac{\gamma_{n-1/2} h_{n-1}}{8} \right) u_{n-1} + \left(\frac{p_{n-1/2}}{h_{n-1}} + \frac{3\gamma_{n-1/2} h_{n-1}}{8} + \beta \right) u_n = \\ & = \frac{h_{n-1}}{8} (3f_n + f_{n-1}) + \beta u_\beta. \end{aligned} \quad (3.150)$$

Рівняння (3.148–3.150) є системою лінійних алгебраїчних рівнянь (СЛАР) для вектора $U = (u_1, u_2, \dots, u_n)^T$ значень шуканої функції u у вузлах сітки. При цьому в першому рядку матриці A цієї СЛАР тільки два ненульових компоненти a_{11} й a_{12} – це коефіцієнти, на які в рівнянні (3.148) множаться невідомі u_1 й u_2 відповідно:

$$a_{11} = \frac{p_{1+1/2}}{h_1} + \frac{3\gamma_{1+1/2} h_1}{8}, \quad a_{12} = -\frac{p_{1+1/2}}{h_1} + \frac{\gamma_{1+1/2} h_1}{8} \quad (3.151)$$

Дійсно, перше рівняння СЛАР $AU = F$ має вигляд $\sum_{j=1}^n a_{1j} u_j = f_1$, і відсутність у рівнянні (3.148) членів з невідомими u_3, \dots, u_n означає, що відповідні компоненти a_{13}, \dots, a_{1n} у матриці A повинні дорівнювати нулю. Аналогічно кожен з рядків матриці A з номером $k = 2, \dots, n-1$ містить три ненульових компонента

$$a_{kk-1} = -\frac{p_{k-1/2}}{h_{k-1}} + \frac{\gamma_{k-1/2} h_{k-1}}{8}, \quad (3.152)$$

$$a_{kk} = \frac{P_{k-1/2}}{h_{k-1}} + \frac{P_{k+1/2}}{h_k} + \frac{3}{8} \left(\gamma_{k-1/2} h_{k-1} + \gamma_{k+1/2} h_k \right), \quad (3.153)$$

$$a_{kk+1} = -\frac{P_{k+1/2}}{h_k} + \frac{\gamma_{k+1/2} h_k}{8}. \quad (3.154)$$

А останній рядок – два ненульових компонента

$$a_{m-1} = -\frac{P_{n-1/2}}{h_{n-1}} + \frac{\gamma_{n-1/2} h_{n-1}}{8}, \quad a_m = \frac{P_{n-1/2}}{h_{n-1}} + \frac{3\gamma_{n-1/2} h_{n-1}}{8} + \beta \quad (3.155)$$

Таким чином, у результаті скінченно-об'ємної апроксимації крайової задачі (3.130–3.132) нами отриманий її дискретний аналог у вигляді СЛАР

$$AU = W \quad (3.156)$$

з симетричною трьохдіагональною матрицею виду

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 & \dots & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & \dots & 0 & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & a_{m-1} & a_{nn} \end{pmatrix} \quad (3.157)$$

ненульові компоненти якої визначаються формулами (3.151–3.155). Компоненти ж вектора правої частини W скінченно-об'ємної СЛАР (3.156) повністю визначаються правими частинами відповідних рівнянь системи (3.148–3.150).

Відзначимо, що крайова умова другого роду (3.131) дає внесок тільки у відповідний компонент правої частини скінченно-об'ємної СЛАР (див. (3.148)), а крайова умова третього роду (3.132) дає внесок і в компонент матриці СЛАР, і в компонент її правої частини (див. (3.150) і (3.155)). Якщо ж у граничному вузлі x_1 (або x_n) задана крайова умова першого роду (3.130), то значення шуканої функції u в цьому вузлі визначене і тому немає необхідності складати для відповідного скінченного об'єму \tilde{G}_1 (або \tilde{G}_n)

інтегробалансне співвідношення. Замість нього в СЛАР може бути вставлене рівняння $u_1 = u_g$ (або $u_n = u_g$). Очевидно, що рівняння виду $u_1 = u_g$ (або $u_n = u_g$) порушує симетричність матриці (3.157) СЛАР (3.156), оскільки йому в матриці A відповідає рядок з одиницею на діагоналі й усіма нульовими позадіагональними елементами.

Однак неважно симетризувати матрицю скінченно-об'ємної СЛАР і за наявності в крайовій задачі крайової умови першого роду.

Для цього достатньо за допомогою рядків СЛАР, які описують перші крайові умови, виключити ненульові компоненти матриці A зі стовпців з номерами, що збігаються з номерами вузлів, у яких задана крайова умова першого роду. Так, при побудові крайової умови першого роду у вузлі x_1 для виключення компонента a_{21} з першого стовпця другого рядка достатньо відняти від другого рівняння СЛАР перше рівняння, помножене на значення a_{21} . Враховуючи, що після врахування крайової умови першого роду, заданої у вузлі x_1 , перший рядок матриці A має єдиний ненульовий елемент, який розміщений на головній діагоналі й дорівнює одиниці, процедура виключення компонента a_{21} буде полягати у відніманні із другого компонента вектора правої частини W величини $a_{21}u_g$ й обнулінні елемента a_{21} у матриці A . Таким чином, після врахування крайової умови першого роду, заданої у вузлі x_1 , матриця A і вектор правої частини W набудуть вигляду

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots \\ 0 & a_{22} & a_{23} & 0 & \dots \\ 0 & a_{32} & a_{33} & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix}, \quad W = \begin{pmatrix} u_g \\ g_2 - a_{21}u_g \\ g_3 \\ \vdots \end{pmatrix}$$

На закінчення даного пункту розглянемо питання технології складання скінченно-об'ємної СЛАР. Можливі два підходи до формування матриці й вектора правої частини скінченно-об'ємної СЛАР. У першому підході для кожного скінченного об'єму записується рівняння, отримане в результаті апроксимації на ньому відповідного інтегробалансного співвідношення – для одновимірної еліптичної крайової задачі це рівняння виду (3.145), (3.146) або (3.147). Потім по цьому рівнянню визначаються всі компоненти одного рядка матриці СЛАР у вигляді коефіцієнтів, на які множаться невідомі u_j , а також відповідний компонент вектора правої частини у вигляді доданків цього рівняння, що не містять як співмножники невідомі u_j .

Таким чином, рядки скінченно-об'ємної СЛАР генеруються по черзі проходом по всіх вузлах сітки, причому кожний рядок формується відразу цілком (а не окремими частинами) у результаті обробки одного інтегробалансного співвідношення для скінченного об'єму, побудованого навколо відповідного вузла сітки. Тому такий підхід отримав назву повузлового збирання матриці й вектора правої частини скінченно-об'ємної СЛАР. На перший погляд цей підхід представляється найбільш природним і наочним при реалізації МСО. Однак при розв'язуванні двовимірних та тривимірних крайових задач він виявляється набагато менш зручним у порівнянні з іншим підходом, названим поелементним збиранням СЛАР.

Суть поелементного складання матриці й вектора правої частини скінченно-об'ємної СЛАР полягає в наступному. Кожна комірка дискретизації G_k розрахункової області G має як свої частини підобласті, що є частинами різних скінченних об'ємів. Нескладно виділити внески від одної комірки дискретизації в інтегробалансні співвідношення для тих скінченних об'ємів, у які входять його складові частини. Так, частина G'_k

комірки дискретизації G_k (див. рис. 3.57), що є також частиною скінченного об'єму \tilde{G}_k , дає внесок

$$-p \frac{du}{dx} \Big|_{x_{k+1/2}} + \int_{x_k}^{x_{k+1/2}} \gamma u dx = -p_{k+1/2} \frac{u_{k+1} - u_k}{h_k} + \frac{\gamma_{k+1/2} h_k}{8} (3u_k + u_{k+1}) \quad (3.158)$$

у ліву частину інтегробалансного співвідношення для скінченного об'єму \tilde{G}_k й внесок

$$\int_{x_k}^{x_{k+1/2}} f dx = \frac{h_k}{8} (3f_k + f_{k+1}) \quad (3.159)$$

у його праву частину (див. співвідношення (3.135), (3.138), (3.140), (3.142) і (3.144)), тобто це внески в k -е (відповідно до номера скінченного об'єму й вузла сітки) рівняння скінченно-об'ємної СЛАР. Частина ж G_k'' комірки дискретизації G_k , що є частиною скінченного об'єму \tilde{G}_{k+1} , дає внесок

$$p \frac{du}{dx} \Big|_{x_{k+1/2}} + \int_{x_{k+1/2}}^{x_{k+1}} \gamma u dx = p_{k+1/2} \frac{u_{k+1} - u_k}{h_k} + \frac{\gamma_{k+1/2} h_k}{8} (u_k + 3u_{k+1}) \quad (3.160)$$

у ліву частину інтегробалансного співвідношення для скінченного об'єму \tilde{G}_{k+1} та внесок

$$\int_{x_{k+1/2}}^{x_{k+1}} f dx = \frac{h_k}{8} (f_k + 3f_{k+1})$$

у його праву частину, тобто це внески в $(k+1)$ -е рівняння скінченно-об'ємної СЛАР.

Введемо на комірни дискретизації G_k локальну нумерацію вузлів сітки $\hat{x}_1^k = x_k, \hat{x}_2^k = x_{k+1}$, тобто ліва вершина комірки G_k одержала локальний номер 1, а права – локальний номер 2 (верхній індекс k у \hat{x}_1^k й \hat{x}_2^k – це номер розглянутої комірки дискретизації). Позначимо відповідно до локальної

нумерації вузлів комірки G_k значення невідомих u_k і u_{k+1} у цих вузлах \hat{u}_1^k і \hat{u}_2^k . Запишемо внески (3.158) і (3.160) в k -ю й $(k+1)$ -ю рівняння скінченно-об'ємної СЛАР у вигляді

$$\begin{cases} -p_{k+1/2} \frac{u_{k+1} - u_k}{h_k} + \frac{\gamma_{k+1/2} h_k}{8} (3u_k + u_{k+1}) \\ p_{k+1/2} \frac{u_{k+1} - u_k}{h_k} + \frac{\gamma_{k+1/2} h_k}{8} (u_k + 3u_{k+1}) \end{cases} = \begin{Bmatrix} \hat{a}_{11}^k \hat{u}_1^k + \hat{a}_{12}^k \hat{u}_2^k \\ \hat{a}_{21}^k \hat{u}_1^k + \hat{a}_{22}^k \hat{u}_2^k \end{Bmatrix} =$$

$$= \begin{pmatrix} \hat{a}_{11}^k & \hat{a}_{12}^k \\ \hat{a}_{21}^k & \hat{a}_{22}^k \end{pmatrix}, \hat{U}^k = \begin{pmatrix} \hat{u}_1^k \\ \hat{u}_2^k \end{pmatrix} \quad (3.161)$$

тобто внески від комірки G_k у ліві частини k -го й $(k+1)$ -го рівнянь скінченно-об'ємної СЛАР можуть бути представлені в матричному вигляді $\hat{A}^k \hat{U}^k$, де \hat{A}^k – локальна матриця, а \hat{U}^k – локальний вектор невідомих комірок G_k :

$$\hat{A}^k = \begin{pmatrix} \hat{a}_{11}^k & \hat{a}_{12}^k \\ \hat{a}_{21}^k & \hat{a}_{22}^k \end{pmatrix}, \hat{U}^k = \begin{pmatrix} \hat{u}_1^k \\ \hat{u}_2^k \end{pmatrix} \quad (3.162)$$

З (3.162) легко одержати вирази для компонентів локальної матриці \hat{A}^k скінченного об'єму G_k :

$$\begin{aligned} \hat{a}_{11}^k &= \frac{p_{k+1/2}}{h_k} + \frac{3\gamma_{k+1/2} h_k}{8}, & \hat{a}_{12}^k &= -\frac{p_{k+1/2}}{h_k} + \frac{\gamma_{k+1/2} h_k}{8}, \\ \hat{a}_{21}^k &= -\frac{p_{k+1/2}}{h_k} + \frac{\gamma_{k+1/2} h_k}{8}, & \hat{a}_{22}^k &= \frac{p_{k+1/2}}{h_k} + \frac{3\gamma_{k+1/2} h_k}{8}. \end{aligned} \quad (3.163)$$

Процедура формування елементів глобальної матриці A скінченно-об'ємної СЛАР з елементів локальних матриць \hat{A}^k може бути реалізована в такий спосіб. Спочатку обнуляються всі елементи глобальної матриці A .

Потім у циклі по комірках дискретизації G_k формуються локальні матриці \hat{A}^k за формулами (3.163) і їх елементи заносяться в глобальну

матрицю за наступним правилом. Елемент \hat{a}_{IJ}^k локальної матриці \hat{A}^k повинен бути доданий до елемента \hat{a}_{IJ}^k глобальної матриці \hat{A}^k з номером рядка I , відповідним до локального номера i , і номером стовпця J , відповідним до локального номера j . Тобто елемент \hat{a}_{11}^k повинен бути доданий до елемента a_{kk} , \hat{a}_{12}^k – до елемента a_{kk+1} , \hat{a}_{21}^k – до елемента a_{k+1k} й \hat{a}_{22}^k – до елемента a_{k+1k+1} . Неважко переконатися, що k -ий рядок глобальної матриці A буде повністю сформований після занесення в нього локальних матриць комірок дискретизації G_k й G_{k+1} . Таке збирання матриці A фактично відповідає формуванню на скінченному об'ємі \tilde{G}_k балансних співвідношень за допомогою об'єднання внесків у них від комірок дискретизації G_k й G_{k+1} , із частин яких і був створений скінченний об'єм \tilde{G}_k . Після обробки всіх комірок G_k одержимо матрицю A скінченно-об'ємної СЛАР у вигляді

$$A = \begin{pmatrix} \hat{a}_{11}^1 & \hat{a}_{12}^1 & 0 & 0 & 0 & 0 & \dots \\ \hat{a}_{21}^1 & \hat{a}_{22}^1 + \hat{a}_{11}^2 & \hat{a}_{12}^2 & 0 & 0 & 0 & \dots \\ 0 & \hat{a}_{21}^2 & \hat{a}_{22}^2 + \hat{a}_{11}^3 & \hat{a}_{12}^2 & 0 & 0 & \dots \\ 0 & 0 & \hat{a}_{21}^3 & \hat{a}_{22}^3 + \hat{a}_{11}^4 & \hat{a}_{12}^2 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{pmatrix}$$

На заключному етапі поелементного збирання матриці скінченно-об'ємної СЛАР необхідно додати внески від крайових умов – до відповідного діагонального елемента матриці A повинен бути доданий лише коефіцієнт β із крайової умови третього роду.

Так, при задаванні крайової умови третього роду (3.132) на правій границі розрахункової області G потік $-\frac{du}{dx}\Big|_{x=x_n}$ у балансному співвідношенні для скінченного об'єму \tilde{G}_n замінюється величиною

$\beta (u_n - u_\beta)$ – це означає, що від крайової умови третього роду в елемент a_{nn} повинен бути доданий внесок β .

Після проходження по всіх комірках дискретизації й додавання внесків від крайових умов ми одержимо матрицю A з елементами a_{ij} , визначеними співвідношеннями (3.151–3.155).

Процес формування матриці скінченно-об’ємної СЛАР проходимо по комірках дискретизації й збиранням глобальної матриці з локальних матриць комірок G_k часто називають поелементним збиранням глобальної матриці.

Процедура поелементного збирання вектора правої частини скінченно-об’ємної СЛАР реалізується аналогічно процедурі поелементного складання матриці.

Одночасно з обнулінням компонентів матриці скінченно-об’ємної СЛАР обнуляються компоненти її правої частини.

У процесі обходу комірок дискретизації G_k одночасно з компонентами локальної матриці (3.163) формуються компоненти локального вектора \hat{G}^k у вигляді внесків у балансові співвідношення від частин $G'_k \subset \tilde{G}_k$ і $G''_k \subset \tilde{G}_{k+1}$ комірки дискретизації G_k (див. співвідношення (3.159) і (3.161)):

$$\hat{W}^k = \begin{pmatrix} \hat{g}_1^k \\ \hat{g}_2^k \end{pmatrix} = \begin{pmatrix} \frac{h_k}{8} (3\hat{f}_1^k + \hat{f}_2^k) \\ \frac{h_k}{8} (3\hat{f}_1^k + 3\hat{f}_2^k) \end{pmatrix} \quad (3.164)$$

де \hat{f}_1^k й \hat{f}_2^k – значення функції правої частини f рівняння (3.129) у вузлах \hat{x}_1^k і \hat{x}_2^k комірки дискретизації G_k . Потім компоненти \hat{g}_1^k й \hat{g}_2^k локального вектора правої частини \hat{W}^k заносяться в глобальний вектор

правої частини W скінченно-об'ємної СЛАР відповідно до локальної й глобальної нумерації вузлів на оброблюваному елементі сітки G_k , тобто

$$W = \begin{pmatrix} \hat{g}_1^1 \\ \hat{g}_2^1 + \hat{g}_1^2 \\ \hat{g}_2^2 + \hat{g}_1^3 \\ \vdots \\ \hat{g}_2^{n-2} + \hat{g}_1^{n-1} \\ \hat{g}_2^{n-1} \end{pmatrix} \quad (3.165)$$

Після закінчення проходу по всіх комірках G_k враховуються крайові умови другого й третього роду (3.131) і (3.132). Ці умови дають внесок θ або βu_β у компонент g_1 або g_n глобального вектора W . За наявності крайової умови першого роду (3.130) вона враховується заміною першого або останнього рівняння отриманої скінченно-об'ємної СЛАР рівнянням виду $u_1 = u_g$ або $u_n = u_g$ з можливою наступною симетризацією матриці скінченно-об'ємної СЛАР.

Для більшості практичних задач об'ємне джерело f поля \mathcal{U} відмінне від нуля тільки в деяких підобластях розрахункової області G , тобто функція f із правої частини рівняння (3.129) є, як правило, розривною функцією. Технологія поелементного складання вектора правої частини скінченно-об'ємної СЛАР природно враховує таку ситуацію – якщо у вузлі x_k функція f розривна, то на комірці дискретизації G_{k-1} , у якому x_k є правою границею, як \hat{f}_2^{k-1} потрібно взяти значення $f(x_k - 0)$ (тобто значення f ліворуч від x_k), а на комірці, у якій x_k є лівою границею, як \hat{f}_1^k потрібно взяти значення $f(x_k + 0)$ (тобто значення f праворуч від x_k). При цьому, природно, усі точки розриву функції f , як і точки розриву коефіцієнтів p і γ диференціального рівняння (3.129), повинні бути включені в скінченно-об'ємну сітку як її вузли. Взагалі, розрахункову область G краще представляти у вигляді окремих підобластей, що

описують основні складові частини об'єкта моделювання. У кожній же з таких складових частин коефіцієнти й права частина диференціального рівняння (3.129), як правило, однотипні (є деякими константами або функціями певного виду), що дуже зручно для програмної реалізації МСО. При такому підході кожній комірці G_k скінченно-об'ємної сітки може бути надане число, що позначає номер складеної частини досліджуваного об'єкта, і значення коефіцієнтів p і γ усередині комірки дискретизації G_k й функції f на її границі можуть бути обчислені відповідно до цього номера. При цьому неважко побудувати сітку дискретизації розрахункової області G так, щоб граничні точки окремих частин об'єкта моделювання були включені в цю сітку як її вузли. Для цього достатньо спочатку включити в сітку всі граничні точки окремих частин досліджуваного об'єкта, а потім згенерувати всі інші вузли сітки між цими граничними точками з урахуванням необхідних розмірів комірок дискретизації G_k . Звернімо увагу на те, що в результаті складання глобального вектора правої частини скінченно-об'ємної СЛАР за описаною вище технологією поелементного складання його компонент g_k для вузла x_k , у якому функція f має розрив, набуде наступного вигляду (див. (3.164) і (3.165)):

$$g_k = \hat{g}_2^{k-1} + \hat{g}_1^k = \frac{h_{k-1}}{8} (\hat{f}_1^{k-1} + 3\hat{f}_2^{k-1}) + \frac{h_k}{8} (3\hat{f}_1^k + \hat{f}_2^k) = \\ = \frac{h_{k-1}}{8} (f(x_{k-1} + 0) + 3f(x_k + 0)) + \frac{h_k}{8} (3f(x_k + 0) + f(x_{k+1} - 0)) \quad (3.166)$$

який збігається з правою частиною співвідношення (3.136) тільки за умови неперервності f у вузлах сітки. Очевидно, що співвідношення (3.166) є більш точним для розривної f .

При реалізації повузлового складання вектора правої частини W скінченно-об'ємної СЛАР не є проблемою врахувати можливу

розривність функції f у вузлах сітки (у результаті в правій частині рівняння (3.136) буде отриманий вираз, що повністю збігається зі значенням g_k з (3.166)), але при поелементному складанні W зробити це простіше й зручніше.

У одновимірному випадку технологія поелементного складання матриці й вектора правої частини скінченно-об'ємної СЛАР не дає істотних переваг у порівнянні з технологією повузлового складання, хоча більша зручність поелементного складання помітна навіть тут: при її використанні всі комірки дискретизації G_k обробляються абсолютно однаково, у той час як при повузловому складанні контрольні об'єми \tilde{G}_k повинні бути оброблені по-різному залежно від того, побудовані вони поблизу внутрішнього або граничного вузла сітки. При використанні технології поелементного складання скінченно-об'ємної СЛАР зручніше також урахувати розривність функції f у правій частині рівняння (3.129). Зазначені переваги технології поелементного складання стають набагато більш вагомими при розв'язуванні двовимірних і тривимірних задач.

Контрольні завдання

1. Запишіть рівняння, які описують еліптичну крайову задачу.
2. Опишіть принцип розбивки розрахункової області на комірки.
3. Опишіть принципи формування СЛАР, яка дискретизує еліптичну крайову задачу.
4. У чому полягає суть поелементного складання матриці й вектора правої частини скінченно-об'ємної СЛАР?
5. У чому полягає процедура поелементного збирання вектора правої частини скінченно-об'ємної СЛАР?

СПИСОК ЛІТЕРАТУРИ

1. Алгоритмы: построение и анализ / [Кормен Т., Лейзерсон Ч. Э., Ривест Р. Н., Штайн К.]. – М. : Вильямс, 2015. – 1328 с.
2. Родосский К. А. Алгоритм Евклида / К. А. Родосский. – М. : Наука, 1968. – 240 с.
3. Литвин А. В. Алгоритмы. Введение в разработку и анализ / А. В. Литвин. – М. : Вильямс, 2006. – 576 с.
4. Воеводин В. В. Параллельные вычисления / В. В. Воеводин, Вл. В. Воеводин. – СПб : Питер, 2002. – 608 с.
5. Демьянович Ю. К. Параллельные алгоритмы. Разработка и реализация : учеб. пособие / Ю. К. Демьянович, И. Г. Бурова, Т. О. Евдокимова. – Бином, 2017. – 344 с.
6. Котов В. Е. Сети Петри / В. Е. Котов. – М. : Наука, 1984. – 158 с.
7. Абрамов С. А. Вычислительная сложность алгоритмов (конспект лекций) / С. А. Абрамов. – М. : МГУ, 2005. – 68 с.
8. Степанов А. Н. Курс информатики для студентов информационно-математических специальностей / А. Н. Степанов. – СПб. : Питер, 2018. – 1008 с.
9. Харари Ф. Теория графов / Ф. Харари. – М. : Мир, 2003. – 296 с.
10. Третьяк Л. Н. Обработка результатов наблюдений : учебное пособие / Л. Н. Третьяк. – Оренбург : ГОУ ОГУ, 2004. – 171 с.
11. Бердянский В. М. Вечный двигатель – прежде и теперь / В. М. Бердянский. – М. : ФИЗМАТЛИТ, 2001. – 264 с.
12. Верещагин Н. К. Вычислимые функции / Н. К. Верещагин, А. Шень. – М. : МЦНМО, 2008. – 192 с.
13. Матрос Д. Ш. Теория алгоритмов: учебник / Д. Ш. Матрос, Г. Б. Поднебесова. – М. : БИНОМ, 2008. – 202 с.

14. Godel K. Collected works. – vol. I, (Publications 1929–1936), 1986, Oxford University Press, 504 p.
15. Turing A. M. On computable numbers, with an application to the Entscheidungs problem. Proceedings of the London Mathematical Society, 2 s. vol. 42 (1936–1937), pp. 230–265.
16. Поляков В. И. Основы теории алгоритмов / В. И. Поляков, В. И. Скорубский. – СПб : СПб НИУ ИТМО, 2012. – 51 с.
17. Пономарев В. Ф. Основы теории алгоритмов: учеб. пособие / В. Ф. Пономарев. – Калининград : КГТУ, 2005. – 53 с.
18. Ильин М. Е. Аппроксимация и интерполяция. Методы и приложения : учеб. пособие / М. Е. Ильин. – Рязань : РГРА, 2010. – 56 с.
19. Анджейчак І. А. Практикум з обчислювальної математики частина 1 : навч. посіб / І. А. Анджейчак, Є. М. Федюк. – Львів : ДУ «Львів. Політех.», 2000. – 100 с.
20. Анджейчак І. А. Практикум з обчислювальної математики. Лекції : навч. посіб. частина 2 / І. А. Анджейчак, Є. М. Федюк, В. Є. Анохін та ін. – Львів : ДУ «Львів. Політех.», 2001. – 152 с.
21. Гаврилюк І. П. Методи обчислень / І. П. Гаврилюк, В. Л. Макаров. – К. : Вища шк., 1995. – 367 с.
22. Ловецкий К. П. Вычислительный эксперимент и методы вычислений : учеб. пособие / К. П. Ловецкий, Л. А. Севастьянов, Е. Б. Ланеев. – М. : РУДН, 2007. – 35 с.
23. Вержбицкий В. М. Основы численных методов : учебник для вузов / В. М. Вержбицкий. – М. : Директ-Медиа, 2013. – 847 с.
24. Вержбицкий В. М. Численные методы : учеб. пособие / В. М. Вержбицкий. – М. : В. Ш., 2001. – 382 с.
25. Огородникова О. М. Вычислительные методы в компьютерном инжиниринге : учебное пособие / О. М. Огородникова. – Екатеринбург : УрФУ, 2013. – 130 с.

26. Трегубников С. В. Численные методы: учебно-методическое пособие / С. В. Трегубников. – Брянск : БГТУ, 2008. – 81 с.
27. Мустафина Д. А. Дифференцирование функций одной и нескольких переменных с приложениями : учебное пособие / Д. А. Мустафина, И. В. Ребро, С. Ю. Кузьмин, Н. Н. Короткова. – Волгоград : РПК «Политехник», 2009. – 118 с.
28. Самборська О. М. Числові методи : навч. посібник / О. М. Самборська, Б. Г. Шелестовський. – Тернопіль : ТДТУ ім. Івана Пулюя, 2008. – 140 с.
29. Бігун Я. Й. Числові методи розв'язування нелінійних рівнянь і систем : навч. посібник / Я. Й. Бігун, І. В. Березовська. – Чернівці : Чернівецький нац. ун-т, 2011. – 104 с.
30. Синкевич Г. И. Отделение корней алгебраического уравнения в XVII и XVIII веке / Г. И. Синкевич // «Математическое моделирование, численные методы и комплексы программ». – 2014. – Вып. 20. – СПб. : СПбГАСУ. – С. 22–38.
31. Гловацкая А. П. Методы и алгоритмы вычислительной математики : учеб. пособие / А. П. Гловацкая. – М. : «Радио и связь», 1999. – 408 с.
32. Калашников В. Л. Введение в численные методы / В. Л. Калашников. – К. : НТУУ «КПИ», 2002. – 132 с.
33. Каленюк П. І. Вступ до числових методів / П. І. Каленюк, В. Л. Бакалець. – Львів : Н. У., 2000. – 184 с.
34. Косарев В. И. 12 лекций по вычислительной математике / В. И. Косарев. – М. : МФТИ, 2000. – 164 с.
35. Лященко М. Я. Чисельні методи / М. Я. Лященко, М. С. Головань. – К. : «Либідь», 1996. – 288 с.
36. Мороз В. В. Обчислювальна математика : навч. посібник / В. В. Мороз, Л. М. Трасковецька. – Хмельницький : ХНУ, 2004. – 124 с.
37. Цегелик Г. Г. Чисельні методи : підручник / Г. Г. Цегелик. – Львів : Н. У., 2004. – 407 с.

38. Волков Е. А. Численные методы / Е. А. Волков. – М. : «Наука», 1982. – 317 с.
39. Демидович Б. П. Основы вычислительной математики / Б. П. Демидович, И. Л. Марон. – М. : Наука, 1966. – 664 с.
40. Элементы линейной алгебры / [Р. Ф. Апатенок, А. М. Маркина, Н. В. Попова, В. Б. Хейнман]. – Мн. : «Вышэйш школа», 1977. – 256 с.
41. Белоусов И. В. Матрицы и определители : учеб. пособие / И. В. Белоусов. – Кишинев : ИПФ АНРМ, 2006. – 101 с.
42. Конев В. В. Линейная алгебра : учеб. пособие / В. В. Конев. – Томск : Изд. ТПУ, 2008. – 65 с.
43. Кремер Н. Ш. Линейная алгебра / Н. Ш. Кремер, М. Н. Фридман, И. М. Тришин. – М. : «Юрайт», 2018. – 422 с.
44. Шерстнева А. И. Линейные пространства. Линейные операторы / А. И. Шерстнева, О. В. Янущик. – Томск : Изд. ТПУ, 2010. – 92 с.
45. Черпушева Т. В. Алгебра матриц : учеб. пособие / Т. В. Черпушева, О. Б. Васюнина. – Пенза : Изд. ПГУ, 2010. – 61 с.
46. Калиткин Н. Н. Численные методы / Н. Н. Калиткин. – М. : Наука, 1978. – 512 с.
47. Стренг Г. Линейная алгебра и ее применения / Г. Стренг. – М. : «Мир», 1980. – 456 с.
48. Высшая математика для технических университетов. Линейная алгебра : учеб. пособие / [В. Н. Задорожный, В. Ф. Зальмеж, А. Ю. Трифонов, А. В. Шаповалов]. – Томск : Изд. ТПУ, 2009. – 310 с.
49. Ващенко Г. В. Вычислительная математика. Основы конечных методов решения систем линейных алгебраических уравнений : учеб. пособие / Г. В. Ващенко. – Красноярск : СибГТУ, 2005. – 80 с.
50. Сулова А. С. Матрицы и системы линейных алгебраических уравнений : учеб. пособие / А. С. Сулова. – Саратов : Изд. СГУ, 2011. – 55 с.

51. Высшая математика в примерах и задачах / [Ю. Л. Геворкян, Л. А. Балака, С. С. Габриелян и др.]. – Харьков: Вид. «Підручник НТУ «ХП», 2011. – 408 с.
52. Кветний Р. Н. Методи комп'ютерних обчислень : навч. посібник / Р. Н. Кветний. – Вінниця: ВДГУ, 2001. – 148 с.
53. Самарский А. А. Численные методы / А. А. Самарский, А. В. Гулин. – М. : «Наука», 1989. – 432 с.
54. Осиновская И. В. Численные методы решения алгебраических уравнений и их систем / И. В. Осиновская, А. Г. Шляпугин, Я. А. Ерисов. – Самара : Самар. гос. аэрокосм. ун-т им. С. П. Королева, 2012. – 69 с.
55. Численные методы : учеб. пособие / [С. Н. Кабанов, Д. С. Лукомский, С. И. Поликарпов и др.]. – Саратов : Изд. СГУ, 2013. – 26 с.
56. Амосов А. А. Вычислительные методы для инженеров : учеб. пособие / А. А. Амосов, Ю. А. Дубинский, Н. В. Копченова. – М. : Высшая школа, 1994. – 544 с.
57. Тарасов В. Н. Численные методы. Теория, алгоритмы программы / В. Н. Тарасов, Н. Ф. Бахарева. – Оренбург : ИПК ОГУ, 2008. – 264 с.
58. Захаров В. Н. Итерационные методы решения систем линейных алгебраических уравнений : учебное пособие / В. Н. Захаров. – Кемерово : КемГУ, 2011. – 170 с.
59. Пирумов У. Г. Численные методы : учеб. пособие / У. Г. Пирумов. – М. : Дрофа, 2007. – 221 с.
60. Копченова Н. В. Вычислительная математика в примерах и задачах / Н. В. Копченова, И. А. Марон. – М. : Наука, 1972. – 367 с.
61. Попов В. В. Методи обчислень: конспект лекцій / В. В. Попов. – К. : Видавничо-поліграфічний центр «Київський університет», 2012. – 303 с.
62. Шарый С. П. Курс вычислительных методов / С. П. Шарый. – Новосибирск : НГУ, 2018. – 611 с.

63. Бураго Н. Г. Основы вычислительной математики / Н. Г. Бураго. – М. : МГТУ им. Н. Э. Баумана, 2011. – 210 с.
64. Самарский А. А. Введение в численные методы / А. А. Самарский. – М. : Наука, 1982. – 272 с.
65. Задачин В. М. Чисельні методи: навч. посібник / В. М. Задачин, І. Г. Конюшенко. – Харків : Вид. ХНЕУ, 2014. – 180 с.
66. Котюргина А. С. Численные методы: учеб. пособие / А. С. Котюргина. – Омск : Изд. ОмГТУ, 2010. – 84 с.
67. Прокопенко Ю. В. Обчислювальна математика: навч. посібник / Ю. В. Прокопенко, Д. Д. Татарчук, В. А. Казміренко. – К. : «Політехніка», 2013. – 224 с.
68. Бахвалов Н. С. Численные методы / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. – М. : Бинوم, 2011. – 640 с.
70. Пименов В. Г. Численные методы: [в 2 ч., ч.1: учеб. пособие] / В. Г. Пименов. – Екатеринбург : Изд. УрФу, 2013. – 112 с.
71. Тимербаев М. Р. Численные методы: учеб. пособие / М. Р. Тимербаев. – Казань: Изд. КазФУ, 2015. – 92 с.
72. Киреев В. И. Численные методы в примерах и задачах / В. И. Киреев, А. В. Пантелеев. – М. : OZON, 2008. – 380 с.
73. Демидович Б. П. Численные методы анализа. Приближение функций, дифференциальные и интегральные уравнения / Б. П. Демидович, И. А. Марон, Э. З. Шувалова. – М. : «Наука», 1967. – 368 с.
74. Асламова В. С. Вычислительная математика / В. С. Асламова, А. Г. Колмогоров, Н. Н. Сумарокова. – Ангарск : АГТА, 2005. – 94 с.
75. Фалейчик Б. В. Одношаговые методы численного решения задачи Коши : учеб. пособие / Б. В. Фалейчик. – Минск : БГУ, 2010. – 42 с.
76. Демченко В. В. Краевые задачи для обыкновенных дифференциальных уравнений второго порядка. Метод прогонки: учеб. пособие / В. В. Демченко. – М. : МФТИ, 2004. – 56 с.

77. Шуп Т. Решение инженерных задач на ЭВМ : практическое руководство / Т. Шуп. – М. : Мир, 1982. – 238 с.
78. Борковская И. М. Уравнения математической физики: учеб. пособие / И. М. Борковская, О. Н. Пыжкова. – Минск : БГТУ, 2010. – 75 с.
79. Вычислительная математика : учеб. пособ / [Н. И. Данилина, Н. С. Дубровская, О. П. Кваша, Г. Л. Смирнов]. – М. : «Высшая школа», 1985. – 472 с.
80. Крылов В. И. Начала теории вычислительных методов. Уравнения в частных производных / В. И. Крылов, В. В. Бобков, П. И. Монастырный. – Минск : Наука и техника, 1986. – 311 с.
81. Рояк М. Э. Сеточные методы решения краевых задач математической физики: учеб. пособие / М. Э. Рояк, Ю. Г. Соловейчик, Э. П. Шурина. – Новосибирск : Изд. НГТУ, 1998. – 120 с.
82. Нестеренко Б. Б. Основы асинхронных методов параллельных вычислений / Б. Б. Нестеренко, В. А. Марчук. – К. : «Наукова думка», 1989. – 176 с.
83. Новотарський М. А. Штучні нейронні мережі: обчислення / М. А. Новотарський, Б. Б. Нестеренко. – К. : Інститут математики НАН України, 2004. – 408 с.
84. Ковеня В. М. Метод конечных разностей и конечных объемов для решения задач математической физики: учеб. пособие / В. М. Ковеня, Д. В. Чирков. – Новосибирск : Изд. НГУ, 2013. – 87 с.