

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Є. І. БАЙДА, О. Ю. КРОПАЧЕК

ЧИСЕЛЬНІ МЕТОДИ

Харків 2020

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Є.І. БАЙДА, О. Ю. КРОПАЧЕК

ЧИСЕЛЬНІ МЕТОДИ

**Навчально-методичний посібник для магістрів та аспірантів
спеціальності 141 «Електроенергетика, електротехніка і
електромеханіка» усіх форм навчання**

Затверджено
редакційно - видавничою
радою університету
протокол № 2 от 17.05.2019

Харків
НТУ «ХПІ»
2020

УДК 519.6

Рецензенти:

Б.В. Клименко, д-р техн. наук, проф., НТУ «ХПИ»;

М.Г. Пантелят, канд. физ.- мат. наук, НТУ «ХПИ»

Байда Є. І., Кропачек О. Ю.

Чисельні методи: навч.-метод посіб. / Є. І. Байда, О. Ю. Кропачек – Харків. :
НТУ «ХПИ», 2020. – 118 с.

Навчально-методичний посібник містить ряд теоретичних та практичних положень за численним методом рішень електротехнічних задач. Призначається для магістрів та аспірантів спеціальності 141 «Електроенергетика, електротехніка і електромеханіка» усіх форм навчання

Стор. 118. Іл. 19. Табл. 4. Бібліогр. : 5 наім. Додатків 2.

УДК 519.6

© Є. І. Байда, О. Ю. Кропачек

© НТУ "ХПИ", 2020

ЗМІСТ

ЗМІСТ	3
ВСТУП	7
1. ЕТАПИ РОЗВ'ЯЗАННЯ ЗАДАЧ НА ЕОМ	8
2. ОСНОВИ РОБОТИ З СИСТЕМОЮ MAPLE	9
2.1. Алфавіт мови. Поняття про оператори і функції	9
2.2. Типи даних	10
2.3. Найменування змінних	10
2.4. Деякі математичні вбудовані функції мови MAPLE	11
2.5. Деякі математичні операції з векторами і матрицями	11
2.6. Типові засоби програмування	12
2.6.1. <i>Умовні оператори</i>	12
2.6.2. <i>Оператори циклу</i>	12
2.6.3. <i>Функції користувача</i>	13
2.6.4. <i>Найпростіші процедури</i>	13
Висновки	13
3. АПРОКСИМАЦІЯ ФУНКЦІЙ	14
3.1. Локальна апроксимація	14
3.1.1. <i>Точкова апроксимація</i>	14
3.1.2. <i>Лінійна та квадратична інтерполяція</i>	14
3.1.3. <i>Сплайни</i>	16
3.2. Глобальна апроксимація	16
3.2.1. <i>Метод обраних точок</i>	17
3.2.2. <i>Метод найменших квадратів</i>	17
4. ДИФЕРЕНЦІЮВАННЯ І ІНТЕГРУВАННЯ	20
4.1. Диференціювання	20
4.1.1. <i>Апроксимація похідних</i>	20
4.1.2. <i>Похибки апроксимації</i>	20
4.2. Інтегрування	21
4.2.1. <i>Метод прямокутників</i>	22
4.2.2. <i>Метод трапецій</i>	22
4.2.3. <i>Метод Сімпсона</i>	22
4.3. Кратні інтеграли	24

4.3.1. <i>Метод прямокутників</i>	24
4.3.2. <i>Обчислення кратного інтегралу методом трапецій</i>	26
4.3.3. <i>Метод Монте-Карло</i>	26
5. СИСТЕМИ ЛІНІЙНИХ РІВНЯНЬ	29
5.1. Деякі поширені види матриць і методи розв'язання систем рівнянь	29
5.2. Прямі методи	31
5.2.1. <i>Метод визначників</i>	31
5.2.2. <i>Метод Гауса</i>	31
5.2.3. <i>Метод прогонки</i>	33
5.3. Ітераційні методи	34
6. НЕЛІНІЙНІ РІВНЯННЯ	38
6.1. Рівняння з одним невідомим	38
6.1.1. <i>Метод ділення відрізка навпіл (метод дихотомії)</i>	38
6.1.2. <i>Метод хорд</i>	38
6.1.3. <i>Метод Ньютона</i>	39
6.1.4. <i>Метод простої ітерації</i>	40
6.2. Системи нелінійних рівнянь	40
6.2.1. <i>Метод простої ітерації</i>	40
7. ЧИСЕЛЬНІ МЕТОДИ РОЗВ'ЯЗКУ ДИФЕРЕНЦІЙНИХ РІВНЯНЬ ..	42
7.1. Звичайні диференційні рівняння	42
7.2. Однокрокові методи	43
7.2.1. <i>Метод Ейлера</i>	43
7.2.2. <i>Розрахункові схеми другого порядку точності. Модифікований метод Ейлера</i>	44
7.2.3. <i>Розрахункові схеми другого порядку точності (неявні схеми). Ітераційні методи</i>	45
7.2.4. <i>Метод Рунге-Кутта</i>	46
7.3. Крайові задачі	47
7.4. Розв'язок нелінійних звичайних диференційних рівнянь другого порядку	48
7.5. Диференційні рівняння в частинних похідних	49
7.5.1. <i>Одновимірне рівняння теплопровідності</i>	50

7.5.2. Багатовимірні рівняння теплопровідності.....	52
7.5.2. Рівняння Лапласа та Пуассона.....	53
8. ОПТИМІЗАЦІЙНІ МЕТОДИ.....	55
8.1. Одновимірна оптимізація.....	56
8.2. Багатовимірна оптимізація.....	57
СПИСОК ЛІТЕРАТУРИ.....	61
ДОДАТОК А	62
A.1. Лінійна інтерполяція.....	62
A.2. Диференціали і інтеграли.....	63
A.2.1. Метод Сімпсона.....	63
A.2.2. Обчислення подвійного інтегралу.....	64
A.2.3. Метод Монте-Карло.....	65
A.3. Системи лінійних рівнянь.....	66
A.3.1. Метод Гауса.....	66
A.3.2. Вибір головного елемента.....	67
A.3.3. Метод прогонки.....	68
A.3.4. Метод Гауса-Зейделя.....	69
A.4. Нелінійні рівняння.....	70
A.4.1. Метод дихотомії.....	70
A.4.2. Метод простої ітерації.....	71
A.5. Вирішення ОДР. Метод Ейлера.....	72
A.6. Алгоритми оптимізаційних методів.....	73
A.6.1. Одновимірна оптимізація методом золотого перетину.....	73
A.6.2. Метод покоординатного спуску.....	74
ДОДАТОК Б.....	75
B.1. Апроксимація функцій.....	75
B.1.1. Лінійна, квадратична і сплайн-інтерполяція.....	75
B.1.2. Глобальна апроксимація.....	76
B.1.3. Метод найменших квадратів.....	77
B.2. Диференціювання і інтегрування.....	79
B.2.1. Обчислення похідної.....	79
B.2.2. Обчислення похідної від складної функції.....	79
B.2.3. Обчислення похідної від кусково-заданої функції.....	79

Б.2.4. Обчислення інтегралу від кусково-заданої функції	81
Б.2.5. Приклади обчислення інтегралів	82
Б.2.6. Обчислення кратних інтегралів	83
Б.4. Системи лінійних рівнянь	89
Б.4.1. Системи лінійних рівнянь з дійсними коефіцієнтами	89
Б.4.2. Системи лінійних рівнянь з комплексними коефіцієнтами	89
Б.4.3. Розв'язок системи трьох лінійних рівнянь в загальному вигляді ..	90
Б.5. Нелінійні рівняння	92
Б.5.1. Метод дихотомії	92
Б.5.2. Метод Ньютона	93
Б.5.3. Розв'язок нелінійних рівнянь системою MAPLE.....	94
Б.6. Диференційні рівняння	99
Б.6.1. Розв'язок диференційних рівнянь в загальному вигляді.....	99
Б.6.2. Вирішення диференційних рівнянь в чисельному вигляді.....	101
Б.6.3. Розв'язання системи диференційних рівнянь в чисельному вигляді.....	103
Б.6.4. Розв'язання нелінійних диференційних рівнянь в чисельному вигляді	107
Б.6.5. Розв'язання диференційних рівнянь в частинних похідних	109
Б.7. Визначення екстремуму функції	115
Б.7.1. Визначення екстремуму функції однієї змінної	115
Б.7.2. Визначення екстремуму функції двох змінних	116
Б.7.3. Визначення екстремуму функції методом Монте-Карло	116

ВСТУП

На даний час ЕОМ використовуються практично у всіх сферах людської діяльності, а їх застосування набуло масового характеру. У зв'язку з цим виникає потреба в спеціальних знаннях, що дозволяють вирішувати науково-технічні завдання, використовуючи обчислювальні машини. Однією з таких областей знань є обчислювальна математика, яка вивчає методи побудови і дослідження чисельних методів розв'язку математичних задач, що моделюють різні фізичні процеси.

Даний навчальний посібник спрямований на вивчення чисельних методів, що дозволяють вирішити конкретні завдання, які виникають при розрахунку і проектуванні електротехнічних пристроїв. Особливістю цього посібника є те, що основний акцент зроблений на описі самих методів і алгоритмів їх застосування, а не на доведеннях єдності рішення, стійкості, збіжності і т.д., які наводяться у відповідній літературі.

Розглядаючи чисельні методи вирішення завдань, необхідно звернути увагу на те, що в останні роки з'явилися системи так званої «комп'ютерної алгебри», зокрема MAPLE. Володіючи потужною мовою програмування та маючи у своєму арсеналі понад три тисячі різних функцій, система MAPLE дозволяє вирішити практично будь-які завдання, як в аналітичному, так і в числовому вигляді. В даному навчальному посібнику розглянуті основи цієї системи і наведені конкретні приклади вирішення електротехнічних задач. Даний посібник не є підручником по системі MAPLE, тому він розрахований на читача, який володіє елементарними знаннями в програмуванні і здатного самостійно освоїти інтерфейс системи.

1. ЕТАПИ РОЗВ'ЯЗАННЯ ЗАДАЧ НА ЕОМ

При вирішенні будь-якої задачі за допомогою ЕОМ основна роль належить людині, а машина лише виконує його завдання за розробленою програмою. Для з'ясування ролі машини і людини процес вирішення конкретного завдання необхідно розділити на певні етапи.

1. *Постановка завдання.* Полягає у визначенні вихідних параметрів і кінцевих цілей вирішення задачі.

2. *Побудова математичної моделі (математичне формулювання завдання).* Модель повинна адекватно описувати основні фізичні явища. Побудова такої математичної моделі вимагає розуміння фізичних процесів і проблеми дослідження.

3. *Вибір чисельного методу.* Як показує досвід, чисельні методи є основним інструментом вирішення складних математичних задач. Даний етап дозволяє звести задачу розрахунку до деякого обчислювального алгоритму. Користувачу необхідно обрати той метод, який найбільш придатний в даному конкретному випадку.

Багато з чисельних методів розроблені відносно давно, проте їх реалізація стала можливою тільки завдяки розвитку ЕОМ. Чисельні методи повинні забезпечувати отримання результату за прийнятний час, не вносячи в обчислювальний процес значних похибок.

4. *Розробка алгоритму та побудова блок-схеми.* Дозволяє записати процес вирішення задачі у вигляді послідовності елементарних арифметичних і логічних операцій, що призводять до вирішення задачі. Алгоритм можна зобразити у вигляді блок-схеми.

5. *Програмування.* Етап пов'язаний з написанням алгоритму розв'язку задачі на мові, зрозумілій ЕОМ.

6. *Налагодження програми.* Полягає в усуненні синтаксичних і логічних помилок. Для отримання впевненості в достовірності роботи програми необхідно вирішити тестові задачі.

7. *Проведення розрахунків.* Етап включає в себе підготовку вихідних даних і проведення розрахунків.

8. *Аналіз результатів.* Аналізуються отримані результати, за якими робляться певні висновки.

2. ОСНОВИ РОБОТИ З СИСТЕМОЮ MAPLE

MAPLE - система комп'ютерної математики, яка здатна швидко і правильно виконувати як символічні, так і чисельні розрахунки в поєднанні з потужними засобами графічної візуалізації результатів. Система може бути використана для обчислень будь-якої складності - від найпростіших до найскладніших. При цьому необхідно зазначити, що система не може замінити математичні знання, необхідні для правильної постановки задачі, вибору математичної моделі і адекватної оцінки результатів математичного експерименту. MAPLE має потужну довідкову систему з сотнями прикладів, тому вона широко використовується при вирішенні різного роду завдань різними категоріями користувачів, починаючи зі студентів і закінчуючи науковими співробітниками.

2.1. Алфавіт мови. Поняття про оператори і функції

Алфавіт мови містить двадцять шість латинських великих і двадцять шість малих літер, десять арабських цифр і близько тридцяти двох спеціальних символів: арифметичні оператори (наприклад, знак зведення в степінь - \wedge); різні дужки та ін. До спеціальних типів належать наступні символи: `%` - системна змінна, що зберігає попередній результат; `:` - фіксатор виразу без виведення результатів; `;` - фіксатор виразу з виведенням результатів; `:=` - оператор присвоювання; `#` - оператор програмного коментарю.

Синтаксис мови збігається з основними правилами відомих мов програмування: заборонено давати імена змінних зарезервованим словам (*and*, *do*, *od*, *if*, *fi* та ін.); круглі дужки змінюють пріоритет математичних операцій.

Найбільш поширеним оператором є оператор присвоювання, наприклад: `x:=y; z:=2;` Існують також оператори логічних умов (`=`, `<`, `>`, `<>`, `=<`, `>=`); оператори вказівки областей змінних (`i=1..5`).

Система має величезну кількість вбудованих функцій, які на звернення до них повертають значення цієї функції.

Приклад. `z:=2*sin(1.0);` $\rightarrow z:=1.68$; але `z:=2*sin(1);` $\rightarrow z:=2*sin(1)$.

Користувач має можливість створювати свої власні функції.

Приклад. $y:=x \rightarrow x^2$ або $y:=unapply(x^2,x)$; $z:=(x,y) \rightarrow x^2+y^2$; або $z:=unapply(x^2+y^2,x,y)$;

У відповідь на звернення до такої функції вона автоматично повертає в програму своє значення.

2.2. Типи даних

Основними типами даних, з якими працює система MAPLE, є:

1. Цілі; дійсні з десятковою крапкою; раціональні у вигляді дроби; дійсні з мантисою і порядком. Причому якщо користувач вказав число як ціле, то і результат буде у вигляді такого ж числа, наприклад: $z := 1/3$; $\rightarrow z := 1/3$. Перетворити цей вираз в дійсне число можна функцією *evalf*(z).

2. Комплексні числа і функції комплексних чисел. Уявна одиниця при цьому позначається як *I*.

Приклад. $Z:=1.25+I*Pi$;

3. Списки виразів. Являють собою упорядкований набір символів.

Приклад. $X:=[1,2,3,4,5,6]$;

4. Масиви, вектори і матриці. Представляють упорядкований набір символів, доступ до яких здійснюється зазначенням їх номеру в найменуванні набору.

Приклад. $A:=array(1..3,[])$; – порожній одновимірний масив;

$A:=array(1..3,[a1,a2,a3])$; – заповнений одновимірний масив;

$B:=array(1..3,1..5[])$; – двовимірний масив;

$V:=vector(3,[Vx,Vy,Vz])$; – вектор з відповідними проекціями;

$M:=matrix(2,2[])$; – порожня матриця.

Звернення до елемента набору здійснюється однаково: $A[1]$; $M[1,2]$;

2.3. Найменування змінних

Формування імен змінних здійснюється користувачем і не відрізняється від загальноприйнятих в програмуванні правил. Причому змінна може мати як невизначене значення, так і чисельне значення. Для скидання всіх попередніх значень змінних настійно рекомендується починати програму з ключового слова *restart*; . Скидання значення змінної в тілі програми можна здійснити оператором ' '.

Приклад. $x:='x'$;

2.4. Деякі математичні вбудовані функції мови MAPLE

Система включає наступний набір вбудованих функцій:

1. Тригонометричні функції – *sin, cos, tan, csc*.
2. Зворотні тригонометричні функції – *arcsin, arccot*.
3. Гіперболічні функції – *sinh, tanh*.
4. Зворотні гіперболічні функції – *arcsinh, arctanh*.
5. Статечні і логарифмічні функції – *exp, ln, log10, sqrt* – квадратний корінь та ін.
6. Функції комплексного аргументу: *abs* – модуль; *Re* – виділення дійсної частини; *Im* – виділення уявної частини; *argument* – аргумент; *conjugate* – комплексно-поєднане число та ін. Для більш звичного уявлення про результати операцій з комплексними числами може використовуватись функція *evalc*. *Приклад. r:=evalc(Re(z));*
7. Спеціальні математичні функції: *BesselI, BesselJ* та ін. – функції Беселя; *EllipticE, EllipticK* та ін. – еліптичні інтеграли; *erf* – функція помилок та багато інших.

2.5. Деякі математичні операції з векторами і матрицями

У прикладах показано використання функції *evalm* для виконання операцій з векторами і матрицями.

Приклад. V:=vector(3,[Vx,Vy,Vz]);
evalm(V+2); означає $[Vx+2, Vy+2, Vz+2]$;
*evalm(2*V);* означає $[2*Vx, 2*Vy, 2*Vz]$;
*evalm(a*V);* означає $[a*Vx, a*Vy, a*Vz]$;
M:=matrix(2,2,[1,2,3,4]);

$$M := \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix},$$

*evalm(2*M);*

$$\begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix},$$

evalm(2+M);

$$\begin{bmatrix} 3 & 2 \\ 3 & 6 \end{bmatrix},$$

evalm(M^2);

Оператори *by*, *while* є необов'язковими. Більш докладні відомості та конкретні приклади можна отримати з системи допомоги.

2.6.3. Функції користувача

Були розглянуті вище.

Приклад. $z := (x, y) \rightarrow x^2 + y^2;$

$$z := (x, y) \rightarrow x^2 + y^2$$

$z(2,2);$

8.

2.6.4. Найпростіші процедури

Процедура – це модуль програми, що має самостійне значення. Найпростіша форма завдання процедури:

Ім'я:=proc(параметри)

Тіло процедури

end;

Зазвичай процедура повертає значення останнього оператора. За допомогою оператора **RETURN** цей порядок може бути змінений.

Висновки

Розглянуті в розділі 2 основи дають невелике уявлення про систему. Подальші можливості будуть розглянуті на конкретних прикладах. Більш детально з системою можна ознайомитись у відповідній літературі [5] і довідковій системі.

3. АПРОКСИМАЦІЯ ФУНКЦІЙ

Апроксимація (наближення) – це задача по знаходженню функції $\varphi(x)$, яка дозволяє наближено замінити функцію $y = f(x)$ на певному інтервалі. Апроксимація застосовується у випадках, якщо невідомий взаємозв'язок між значеннями y і x (залежність задана у вигляді таблиці), або якщо залежність $y = f(x)$ громіздка і містить вирази, які важко обчислити.

На практиці важливий випадок, коли апроксимуюча функція представлена у вигляді поліному порядку m :

$$\varphi(x) = a_0 + a_1 \cdot x + \dots + a_m \cdot x^m. \quad (3.1)$$

Коефіцієнти поліному (3.1) підбираються таким чином, щоб досягти найменшого відхилення значення функції $\varphi(x)$ від реальної функції $f(x)$.

Апроксимація може бути локальною, якщо $\varphi(x)$ визначається для окремих частин розглянутого інтервалу $[a, b]$ змінної x , або глобальною, якщо апроксимуючий багаточлен $\varphi(x)$, визначається на всьому інтервалі $[a, b]$ зміни аргументу x .

3.1. Локальна апроксимація

3.1.1. Точкова апроксимація

Якщо наближення будується на заданій дискретній множині точок $\{y_i, x_i\}$, то апроксимація називається точковою. До неї відносять інтерполяцію, середньоквадратичне наближення та ін. Використовується у тому випадку, коли апроксимуюча функція $y = f(x)$ задана у вигляді табличної залежності $\{y_i, x_i\}$.

Основним типом точкової апроксимації є інтерполяція (рис. 3.1), яка полягає в тому, що значення інтерполяційного багаточлену $\varphi(x)$ збігаються зі значеннями функції у вузлах інтерполяції $\{x_i, y_i\} \in [a, b]$.

Як правило, така інтерполяція використовується для визначення значення y в проміжних точках $y_{i-1} < y < y_i$, що належать інтервалу $x_{i-1} < x < x_i$.

3.1.2 Лінійна та квадратична інтерполяція

Лінійна інтерполяція полягає в тому, що задані таблицею точки $\{x_i, y_i\}$ при $i=1 \dots n$ з'єднуються прямими відрізками з вершинами в заданих точках $\{x_i, y_i\}$ (рівняння ламаної лінії).

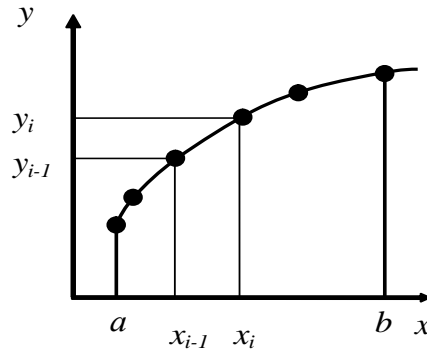


Рисунок 3.1 – Приклад інтерполяції

Оскільки є $n-1$ інтервалів, то для кожної ділянки $x_{i-1} < x < x_i$ використовується рівняння прямої лінії у вигляді

$$\frac{y - y_{i-1}}{y_i - y_{i-1}} = \frac{x - x_{i-1}}{x_i - x_{i-1}}. \quad (3.2)$$

Звідси

$$y = a_i \cdot x + b_i, \quad x_{i-1} < x < x_i,$$

де $a_i = \frac{y_i - y_{i-1}}{x_i - x_{i-1}}$, $b_i = y_{i-1} - a_i \cdot x_{i-1}$.

Для використання лінійної інтерполяції необхідно спочатку визначити інтервал, в який потрапляє аргумент x , а потім шукати наближене значення y , обчисливши попередньо коефіцієнти a_i , b_i .

Квадратична інтерполяція полягає в тому, що в якості інтерполяційної функції на ділянці $x_{i-1} < x < x_i$ виступає квадратичний трьохчлен. Таку інтерполяцію називають також параболічною:

$$y = a_i \cdot x^2 + b_i \cdot x + c_i. \quad (3.3)$$

Рівняння (3.3) містить три невідомих (a_i , b_i , c_i), для визначення яких необхідно записати три рівняння, що визначають умови проходження параболи через три точки ($x_{i-1}, y_{i-1}; x_i, y_i; x_{i+1}, y_{i+1}$):

$$\begin{cases} a_i \cdot x_{i-1}^2 + b_i \cdot x_{i-1} + c_i = y_{i-1}, \\ a_i \cdot x_i^2 + b_i \cdot x_i + c_i = y_i, \\ a_i \cdot x_{i+1}^2 + b_i \cdot x_{i+1} + c_i = y_{i+1}. \end{cases} \quad (3.4)$$

Розв'язок системи:

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

$$b_i = \frac{-x_i^2 y_{i-1} + x_i^2 y_{i+1} + x_{i+1}^2 y_{i-1} - y_i x_{i+1}^2 + y_i x_{i-1}^2 - y_{i+1} x_{i-1}^2}{-x_{i+1} x_{i-1} + x_{i+1} x_i + x_i x_{i-1} - x_{i-1} x_i + x_{i-1} x_{i+1} - x_i x_{i+1}} ;$$

$$c_i = \frac{x_i^2 x_{i+1} y_{i-1} - x_i^2 x_{i-1} y_{i+1} - x_{i+1}^2 x_i y_{i-1} + y_{i+1} x_i x_{i-1}^2 - y_i x_{i+1} x_{i-1}^2 + x_{i+1}^2 x_{i-1} y_i}{-x_{i+1} x_{i-1} + x_{i+1} x_i + x_i x_{i-1} - x_{i-1} x_i + x_{i-1} x_{i+1} - x_i x_{i+1}} .$$

Алгоритм обчислення наближеного значення у такий самий, як і для випадку лінійної інтерполяції з урахуванням зміни блоку обчислення коефіцієнтів (a_i, b_i, c_i).

Приклад. Знайти у при значенні $x = 0.32$, якщо задана таблиця даних (табл. 1.1).

Таблиця 1.1 – Табличне задання функції

x_i	0.15	0.3	0.4	0.55
y_i	2.17	3.63	5.07	7.78

Значення знайдених коефіцієнтів для лінійної інтерполяції: $a_i=14.4$, $b_i= - 0.69$, $y=3.92$; для квадратичної інтерполяції: $a_i=18.67$, $b_i=1.33$, $c_i=1.55$, $y=3.89$.

3.1.3. Сплайни

Сплайн – це локальна інтерполяція функції багаточленом 3-го порядку, механічним аналогом якого є модель тонкого гнучкого стрижня, закріпленого в сусідніх вузлах. В цьому випадку стрижень приймає форму, яка відповідає мінімуму його потенційної енергії. Як і у випадках лінійної і квадратичної інтерполяції, сплайн є кусковою функцією, яка апроксимує вихідну на локальному інтервалі всередині глобального інтервалу апроксимації.

3.2. Глобальна апроксимація

Глобальна апроксимація (рис. 3.2) полягає у визначенні параметрів апроксимуючої функції $\varphi(x)$ на всьому інтервалі зміни $[a, b]$ аргументу x . Задача про глобальну апроксимацію ставиться в разі необхідності отримання

аналітичної залежності табличних даних $\{x_i, y_i\}$. Зазвичай ці дані отримують експериментальним шляхом. Вони як правило, містять помилки виміру, а глобальна апроксимація дозволяє ці помилки згладити. Розглянемо два основні методи глобальної апроксимації.

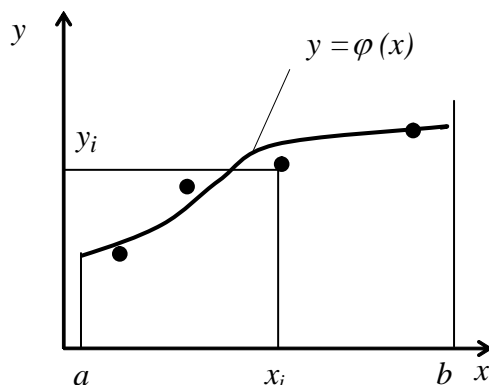


Рисунок 3.2 – Глобальна апроксимація

3.2.1. Метод обраних точок

Припустимо, що вигляд апроксимуючої функції заданий у вигляді поліному

$$y = \varphi(x, a_0, a_1 \dots a_m), \quad (3.5)$$

де m – вища степінь апроксимуючого полінома; a_m – невідомі параметри, для визначення яких вирішується система рівнянь

$$y_i = \varphi(x_i, a_0, a_1 \dots a_m), \quad i = 0, 1 \dots n.$$

Кількість точок n повинно бути не менше ніж $m+1$.

Прикладом може служити визначення коефіцієнтів системи (3.4), отриманих шляхом розв'язку системи лінійних рівнянь.

3.2.2. Метод найменших квадратів

Точність, з якою апроксимуюча функція відображає задані таблицею значення, залежить від обраного способу наближення апроксимуючої функції до табличних значень. Для даних, отриманих експериментальним шляхом, найкращим наближенням вважається середньоквадратичне відхилення, яке полягає у визначенні мінімуму виразу квадрату різниці між апроксимуючою функцією і заданими табличними значеннями:

$$S = \sum_{i=0}^n [(\varphi(x_i, a_0, a_1 \dots a_m) - y_i)]^2, \quad (3.6)$$

де x_i, y_i – задані таблицею значення; $\varphi(x_i, a_0, a_1 \dots a_m)$ – апріорі задана

функція; n – кількість табличних значень, не менше ніж $m+1$.

Оскільки параметри $a_0 .. a_m$ виступають у ролі незалежних змінних, то мінімум S буде за умови $\frac{\partial S}{\partial a_i} = 0$.

Нехай $\varphi(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + .. + a_m \cdot x^m$, тоді

$$\begin{aligned} \frac{\partial S}{\partial a_0} &= 2 \cdot \sum_{i=0}^n (a_0 + a_1 \cdot x_i + a_2 \cdot x_i^2 + .. + a_m \cdot x_i^m - y_i), \\ \frac{\partial S}{\partial a_1} &= 2 \cdot \sum_{i=0}^n (a_0 + a_1 \cdot x_i + a_2 \cdot x_i^2 + .. + a_m \cdot x_i^m - y_i) \cdot x_i, \\ &..... \\ \frac{\partial S}{\partial a_m} &= 2 \cdot \sum_{i=0}^n (a_0 + a_1 \cdot x_i + a_2 \cdot x_i^2 + .. + a_m \cdot x_i^m - y_i) \cdot x_i^m. \end{aligned} \tag{3.7}$$

Провівши деякі перетворення, маємо

$$\begin{aligned} (n+1) \cdot a_0 + a_1 \cdot \sum_{i=0}^n x_i + a_2 \cdot \sum_{i=0}^n x_i^2 + .. + a_m \cdot \sum_{i=0}^n x_i^m &= \sum_{i=0}^n y_i, \\ a_0 \cdot \sum_{i=0}^n x_i + a_1 \cdot \sum_{i=0}^n x_i^2 + a_2 \cdot \sum_{i=0}^n x_i^3 + .. + a_m \cdot \sum_{i=0}^n x_i^{m+1} &= \sum_{i=0}^n y_i \cdot x_i, \\ &..... \\ a_0 \cdot \sum_{i=0}^n x_i^m + a_1 \cdot \sum_{i=0}^n x_i^{m+1} + a_2 \cdot \sum_{i=0}^n x_i^{m+2} + .. + a_m \cdot \sum_{i=0}^n x_i^{2 \cdot m} &= \sum_{i=0}^n y_i \cdot x_i^m. \end{aligned} \tag{3.8}$$

В результаті отримуємо систему лінійних алгебраїчних рівнянь відносно невідомих коефіцієнтів $a_0..a_m$.

Приклад. Задана таблиця значень (табл. 1.2). Апроксимувати значення квадратичним поліномом.

Таблиця 1.2 – Вихідні дані для глобальної апроксимації

x	0.75	1.5	2.25	3.75
y	2.5	1.2	1.12	4.48

Вигляд апроксимуючого поліному

$$y \approx \varphi(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2; m = 2; n = 4,$$

де m – степінь поліному; n – кількість контрольних точок.

Обчислені коефіцієнти повинні мати значення – $a_0=4.97, a_1=-4.09$;

$a_2=1.05$.

Необхідно відзначити, що підвищення степені апроксимуючого поліному знижує точність наближення. Більш кращою у більшості випадків є локальна апроксимація.

У додатку А показано блок-схему алгоритму. В даному алгоритмі необхідно, щоб значення x не виходило за інтервал $x_0 \dots x_n$. В іншому випадку результат інтерполяції буде непередбачуваний.

У додатку Б наведено приклади програми, написаної на мові MAPLE з побудовою відповідних графіків.

З наведених прикладів випливає, що найбільш точною є лінійна і сплайн-інтерполяція. Однак більш кращою є сплайн-інтерполяція, що забезпечує не тільки точність, але і неперервність функції і її похідних.

Найбільш поширеною задачею в електромеханіці є апроксимація основної кривої намагнічування феромагнетиків, з чим зазначені методи успішно справляються. Однак необхідно мати на увазі наступне:

1. Необхідно проявляти обережність під час розрахунків при завданні діапазону зміни апроксимуючої функції, що виходить за межі табличних значень (функція може вести себе неадекватно).

2. Досить добра апроксимація значень функції не означає адекватну апроксимацію і похідну. Так, функція, що апроксимує основну криву намагнічування, часто дає неправильні значення динамічної магнітної проникності.

4. ДИФЕРЕНЦІЮВАННЯ І ІНТЕГРУВАННЯ

4.1. Диференціювання

4.1.1. Апроксимація похідних

За визначенням, похідна обчислюється як приріст функції до приросту аргументу при прагненні останнього до нуля:

$$y' = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}; \quad (4.1)$$

У числових розрахунках використовувати таку формулу не завжди зручно і можливо. Тому на практиці значення Δx вважають рівним кінцевому значенню. Тоді значення похідної визначається як

$$y' \approx \frac{\Delta y}{\Delta x}. \quad (4.2)$$

Це співвідношення називається апроксимацією похідною за допомогою кінцево-різницевого співвідношення. Розглянемо функцію $y = f(x)$, задану в табличній формі. Нехай $h = x_i - x_{i-1}$ – різниця між сусідніми значеннями $x_i \dots x_{i-1}$, $i=1..n$. Запишемо значення похідної в точці (x_i, y_i) .

Тоді

$$y'_i \approx \frac{y_i - y_{i-1}}{h} \text{ – ліва різницева апроксимація;}$$

$$y'_i \approx \frac{y_{i+1} - y_i}{h} \text{ – права різницева апроксимація;}$$

$$y'_i \approx \frac{y_{i+1} - y_{i-1}}{2 \cdot h} \text{ – центральна різницева апроксимація.}$$

Для старших похідних маємо

$$\begin{aligned} y''_i &\approx \frac{y_{i+1} - 2 \cdot y_i + y_{i-1}}{h^2}; \\ y'''_i &= \frac{y_{i+2} - 2 \cdot y_{i+1} + 2 \cdot y_{i-1} - y_{i-2}}{2 \cdot h^3}; \\ y''''_i &= \frac{y_{i+2} - 4 \cdot y_{i+1} + 6 \cdot y_i - 4 \cdot y_{i-1} + y_{i-2}}{h^4}. \end{aligned} \quad (4.3)$$

4.1.2. Похибки апроксимації

Апроксимуємо функцію $y = f(x)$ деякою функцією $\varphi(x)$, представивши результат у вигляді

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

де $R(x)$ – помилка апроксимації, яка залежить від кроку h , причому $h < 1$.

Оцінити похибку апроксимації можна, розклавши функцію $f(x)$ в ряд Тейлора в околиці точки Δx :

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

Використовуючи це співвідношення, отримуємо

$$y'_i = \frac{y_i - y_{i-1}}{h} + O(h); \quad y'_i = \frac{y_{i+1} - y_{i-1}}{2 \cdot h} + O(h^2); \quad y''_i = \frac{y_{i+1} - 2 \cdot y_i + y_{i-1}}{h^2} + O(h^2),$$

де $R(x) = O(h)$.

Похибки при чисельному диференціюванні виникають також у зв'язку з неточним табличним завданням функції у вузлах і при проведенні розрахунків на ЕОМ (помилки округлення).

У додатку Б наведено приклади обчислення меж і символічне обчислення похідної програмою MAPLE.

4.2. Інтегрування

Нехай на інтервалі $[a, b]$ задана деяка функція $y = f(x)$. Тоді з геометричної точки зору інтеграл $S = \int_a^b f(x) \cdot dx$ дорівнює площі фігури, обмеженої лініями: $y = 0$, $y = f(x)$; $x = a$, $x = b$. Для обчислення інтегралу за допомогою точок $x_0 \dots x_n$ розіб'ємо відрізок $[a, b]$ на n елементарних відрізків, причому $x_0 = a$, а $x_n = b$. На кожному з довільних відрізків $[x_{i-1}, x_i]$ виберемо точку ξ_i ($x_{i-1} \leq \xi_i \leq x_i$) всередині зазначеного інтервалу і запишемо площу елементарної ділянки:

$$S_i = f(\xi_i) \cdot \Delta x, \quad (4.4)$$

де $\Delta x = x_i - x_{i-1}$.

Тоді

$$S_n = \sum_{i=1}^n f(\xi_i) \cdot \Delta x \approx \int_a^b f(x) \cdot dx, \quad (4.5)$$

де S_n – інтегральна сума, яка прагне до значення інтегралу при $n \rightarrow \infty$.

4.2.1. Метод прямокутників

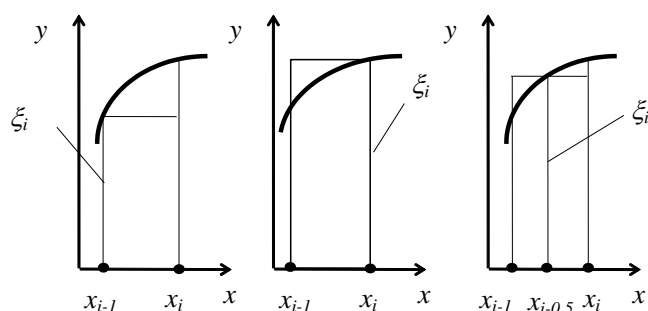


Рисунок. 4.1 – Ілюстрація обчислення інтеграла методом прямокутників

Даний метод безпосередньо використовує формулу (4.5). В якості точок ξ_i можуть вибиратися ліві ($\xi_i=x_{i-1}$), праві ($\xi_i=x_i$) або центральні ($\xi_i=x_{i-0,5}$) точки інтервалу $x_{i-1}-x_i$ (рис. 4.1)

Надалі під формулою прямокутників будемо розуміти останній випадок:

$$S_n = \sum_{i=1}^n f\left(x_{i-1} + \frac{\Delta x}{2}\right) \cdot \Delta x = \sum_{i=1}^n f(x_{i-0,5}) \cdot \Delta x, \quad (4.6)$$

$$\text{де } x_{i-0,5} = \frac{2 \cdot i - 1}{2} \cdot \Delta x + a, \quad i = 1..n, \Delta x = \frac{b-a}{n}.$$

Ілюстрація методу в системі MAPLE наведена в додатку Б.

4.2.2. Метод трапецій

Даний метод використовує лінійну інтерполяцію на інтервалі $[x_{i-1}, x_i]$:

$$S_n = \frac{y_{i-1} + y_i}{2} \cdot \Delta x, \quad (4.7)$$

$$\text{де } i = 1, 2, \dots, n, \Delta x = \frac{b-a}{n}.$$

Тоді

$$\int_a^b f(x) \cdot dx = \frac{1}{2} \cdot \sum_{i=1}^n (y_{i-1} + y_i) \cdot \Delta x = \Delta x \cdot \left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i \right), \quad (4.8)$$

$$\text{де } y_0 = f(a), \quad y_n = f(b).$$

4.2.3. Метод Сімпсона

Розіб'ємо інтервал інтегрування $[a, b]$ на парне число інтервалів n з кроком $\Delta x = \frac{b-a}{n}$. На кожному відрізку замінимо підінтегральну функцію

$y = f(x)$ інтерполяційним багаточленом 2-го порядку, (рис. 4.2):

$$y = f(x) \approx \varphi_i(x) = a_i \cdot x_i^2 + b_i \cdot x_i + c_i,$$

де $x_{i-1} \leq x \leq x_i$.

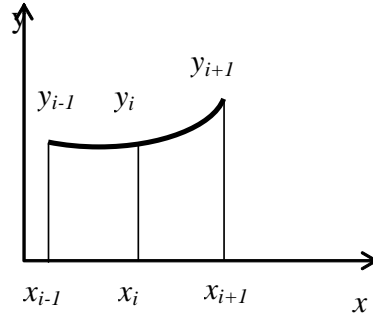


Рисунок 4.2 – Метод Сімпсона

В цьому випадку для $\varphi_i(x)$ можна записати:

$$\begin{aligned} \varphi_i(x) = & \frac{(x - x_i) \cdot (x - x_{i+1})}{(x_{i-1} - x_i) \cdot (x_{i-1} - x_{i+1})} \cdot y_{i-1} + \\ & + \frac{(x - x_{i-1}) \cdot (x - x_{i+1})}{(x_i - x_{i-1}) \cdot (x_i - x_{i+1})} \cdot y_i + \frac{(x - x_{i-1}) \cdot (x - x_i)}{(x_{i+1} - x_{i-1}) \cdot (x_{i+1} - x_i)} \cdot y_{i+1} \end{aligned}$$

Враховуючи, що $S_i = \int_{x_{i-1}}^{x_{i+1}} \varphi_i(x) \cdot dx = \frac{\Delta x}{3} \cdot (y_{i-1} + 4 \cdot y_i + y_{i+1})$, запишемо

остаточний варіант формули Сімпсона при постійному кроці Δx :

$$\int_a^b f(x) \cdot dx = \frac{\Delta x}{3} \cdot [y_0 + 4 \cdot (y_1 + y_3 + \dots y_{n-1}) + 2 \cdot (y_2 + y_4 + \dots y_{n-2}) + y_n], \quad (4.9)$$

де $\Delta x = \frac{b-a}{n}$.

Загальний алгоритм обчислення інтегралу зводиться до наступного:

1. Вибираємо кількість розбиття інтервалу інтегрування і визначаємо крок розбиття.

2. Обчислюємо площу елементарної фігури.

3. Знаходимо інтеграл як суму елементарних площ.

Блок-схема обчислення інтеграла методом Сімпсона з автоматичним вибором кількості розбиттів інтервалу інтегрування в залежності від заданої точності обчислення інтегралу наведена в додатку А.

Приклад. Обчислити інтеграл

$$I = \int_0^1 \frac{1}{1+x^2} \cdot dx = \arctan(x) \Big|_0^1 = \frac{\pi}{4} = 0,785398.$$

Результати обчислення інтегралу різними методами для $n = 10$ наведено в табл. 4.1.

Таблиця 4.1 – Точність обчислення заданого інтегралу різними методами

Метод	Прямо- кутників	Трапецій	Лівих, правих прямокутників	Сімпсона
Помилка	$\varepsilon=0.027 \%$	$\varepsilon=0.054 \%$	$\varepsilon=3.0 \%$	$\varepsilon=0.0 \%$

У додатку Б показані приклади символічного і чисельного обчислення інтегралу системою MAPLE. У разі неможливості обчислення інтегралу в символічному вигляді система завжди зможе обчислити числове значення певного інтегралу.

4.3. Кратні інтеграли

Розглянемо інтегрування подвійного інтегралу виду $I = \iint_G f(x, y) \cdot dx \cdot dy$. З геометричної точки зору даний інтеграл являє собою об'єм тіла, обмеженого зверху поверхнею $f(x, y)$, а знизу площиною xOy . З боків об'єм обмежений площинами, проекції яких на площину xOy представляють область G .

4.3.1. Метод прямокутників

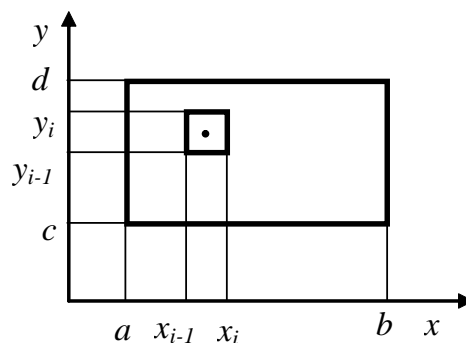


Рисунок 4.3 – Площа елементарної комірки

Одним з найпростіших методів обчислення такого інтегралу є метод комірок. Нехай G – прямокутна область $a \leq x \leq b$, $c \leq y \leq d$, площа якої $S=(b-a) \cdot (d-c)$ (рис. 4.3). Розіб'ємо область G на елементарні площі. Тоді, відповідно до теореми про середнє, об'єм елементарної фігури, побудованої на прямокутнику $[x_{i-1}, x_i]$, $[y_{i-1}, y_i]$, буде визначатися як

$$I_i = S_i \cdot f(\bar{x}, \bar{y}),$$

де \bar{x}, \bar{y} – координати центру комірки.

Нехай $\Delta x = x_i - x_{i-1}$; $\Delta y = y_i - y_{i-1}$, тоді

$$I_i = \Delta x \cdot \Delta y \cdot f(x_{i-0,5}, y_{i-0,5}).$$

Остаточна формула обчислення подвійного інтегралу набирає вигляду

$$I = \iint_G f(x, y) \cdot dx \cdot dy \approx \sum_{i=1}^M \sum_{j=1}^N f(x_{i-0,5}, y_{j-0,5}) \cdot \Delta x_i \cdot \Delta y_j. \quad (4.10)$$

де M, N – кількість розбиття інтервалів інтегрування по осі x, y ;

$$x_{i-0,5} = \frac{2 \cdot i - 1}{2} \cdot \Delta x + a, y_{j-0,5} = \frac{2 \cdot j - 1}{2} \cdot \Delta y + c; \Delta x = \frac{b - a}{M}, \Delta y = \frac{d - c}{N}.$$

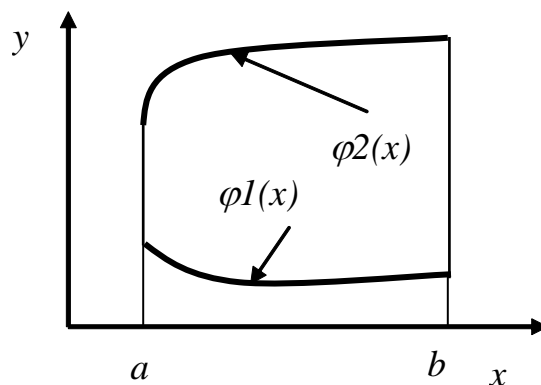


Рисунок 4.4 – Подвійний інтеграл зі змінними межами по осі y

Точність обчислення інтегралу залежить від значень $\Delta x, \Delta y$ і від форми елементарної комірки. Бажано, щоб $\Delta x \approx \Delta y$. Розглянемо випадок, коли межі інтегрування по осі y є функціями $\varphi_1(x)$ і $\varphi_2(x)$ (рис. 4.4).

У цьому випадку кількість підсумовування по напрямку y буде змінною величиною, яка залежить від координати x . Блок-схема обчислення інтеграла

наведена в додатку А.

4.3.2. Обчислення кратного інтегралу методом трапецій

Нехай задана прямокутна область інтегрування (див. рис. 4.3). Необхідно обчислити подвійний інтеграл $I = \iint_G f(x, y) \cdot dx \cdot dy$.

Застосувавши до підінтегрального виразу двічі формулу для обчислення площі елементарних трапецій, отримаємо

$$I = \iint_G f(x, y) \cdot dx \cdot dy \approx \frac{1}{4} \cdot \Delta x \cdot \Delta y \cdot \sum_{i=1}^M \sum_{j=1}^N \{f(x_i, y_j) + f(x_i, y_{j+1}) + f(x_{i+1}, y_j) + f(x_{i+1}, y_{j+1})\}. \quad (4.11)$$

4.3.3. Метод Монте-Карло

Розглянемо деяку випадкову величину ξ , математичне очікування якої

$$M\xi = \bar{\xi}. \quad (4.12)$$

Якщо була проведена серія дослідів, що складається з n незалежних випробувань в результаті яких отримано n випадкових величин ξ_i , то

$$\bar{\xi} = \frac{1}{n} \cdot \sum_{i=1}^n \xi_i. \quad (4.13)$$

Нехай x – випадкова величина з рівномірно розподіленою щільністю ймовірності $p(x)$ на інтервалі $0 \leq x \leq 1$, тоді

$$p(x) = \begin{cases} 0; & x < 0, \\ 1; & 0 \leq x \leq 1, \\ 0; & x > 1. \end{cases}$$

З курсу теорії ймовірності відомо, що математичне очікування деякої випадкової функції $\xi=f(x)$ визначається як

$$M\xi = \int_{-\infty}^{\infty} p(x) \cdot f(x) \cdot dx = \int_0^1 f(x) \cdot dx. \quad (4.14)$$

Отже,

$$\int_0^1 f(x) \cdot dx \approx \frac{1}{n} \cdot \sum_{i=1}^n f(x_i), \quad (4.15)$$

де x_i – випадкова величина, рівномірно розподілена на інтервалі $[0, 1]$.

У разі обчислення кратного інтегралу, якщо область інтегрування G являє собою одиничний квадрат, формула обчислення інтегралу

перетвориться до вигляду

$$\iint_{G=1} f(x, y) \cdot dx \cdot dy \approx \frac{1}{n} \cdot \sum_{i=1}^n f(x_i, y_i), \quad (2.16)$$

де x_i, y_i – вибірка незалежних випадкових величин, рівномірно розподілених в області інтегрування.

Для використання методу Монте-Карло необхідно виробляти послідовність випадкових чисел, рівномірно розподілених в заданій області. Якщо межі інтегрування відмінні від розглянутих меж, то необхідно виконати перетворення області інтегрування.

Як приклад розглянемо перетворення координат для подвійного

інтегралу $J = \int_a^b \int_{\varphi_1(x)}^{\varphi_2(x)} f(x, y) \cdot dx \cdot dy$.

Перетворимо область інтегрування в одиничний квадрат шляхом заміни змінних:

$$x = a + (b - a) \cdot u; \quad dx = (b - a) \cdot du;$$

$$y = \varphi_1(x) + (\varphi_2(x) - \varphi_1(x)) \cdot v; \quad dy = (\varphi_2(x) - \varphi_1(x)) \cdot dv$$

де $0 \leq u \leq 1; 0 \leq v \leq 1$.

В результаті отримаємо

$$J = \int_a^b \int_{\varphi_1(x)}^{\varphi_2(x)} f(x, y) \cdot dx \cdot dy = \int_0^1 \int_0^1 f(u, v) \cdot (\varphi_2(x(u)) - \varphi_1(x(u))) \cdot (b - a) \cdot du \cdot dv$$

$$= \int_0^1 \int_0^1 F(u, v) \cdot (b - a) \cdot du \cdot dv.$$

Приклад. Знайти об'єм верхньої півсфери в системі координат Декарта (рис. 4.5) за допомогою методу Монте-Карло. Рівняння для верхньої півсфери і об'єму будуть наступними:

$$x^2 + y^2 + z^2 = R^2 \Rightarrow z = \sqrt{R^2 - x^2 - y^2},$$

$$V = \int_{-R}^R \int_{-\sqrt{R^2-x^2}}^{\sqrt{R^2-x^2}} \sqrt{R^2 - x^2 - y^2} \cdot dx \cdot dy.$$

Перетворивши область інтегрування, отримаємо розрахункову формулу

$$\begin{aligned}
x &= a + (b - a) \cdot u = -R + 2 \cdot R \cdot u; \quad dx = 2 \cdot R \cdot du; \\
y &= -\sqrt{R^2 - x^2} + 2 \cdot \sqrt{R^2 - x^2} \cdot v; \quad dy = 2 \cdot \sqrt{R^2 - x^2} \cdot dv; \\
V &= 4 \cdot R \cdot \int_0^1 \int_0^1 \sqrt{R^2 - y^2(u, v) - x^2(u)} \cdot \sqrt{R^2 - x^2(u)} \cdot du \cdot dv \\
&= 4 \cdot R \cdot \frac{1}{n} \cdot \sum_{i=1}^n \sqrt{R^2 - y^2(u_i, v_i) - x^2(u_i)} \cdot \sqrt{R^2 - x^2(u_i)}.
\end{aligned}$$

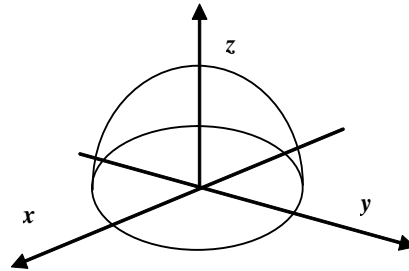


Рисунок 4.5 – Обчислювальний об'єм півсфери

Блок-схема обчислення інтегралів наведена в додатку А. В додатку Б показано обчислення інтегралів в системі MAPLE.

$$\begin{cases} a_1 \cdot x_1 + b_1 \cdot x_2 = c_1, \\ a_2 \cdot x_1 + b_2 \cdot x_2 = c_2. \end{cases}$$

Якщо $\frac{a_1}{a_2} \neq \frac{b_1}{b_2}$, то розв'язком даної системи будуть координати точки

перетину двох прямих (рис. 5.2, а). Якщо $\frac{a_1}{a_2} = \frac{b_1}{b_2} \neq \frac{c_1}{c_2}$, то розв'язок не існує,

а прямі лінії паралельні (рис. 5.1, б). Якщо всі коефіцієнти пропорційні $\frac{a_1}{a_2} = \frac{b_1}{b_2} = \frac{c_1}{c_2}$, то прямі збігаються і система має безліч рішень. При $D \approx 0$

(випадок «майже» паралельних прямих) рішення можуть бути погано зумовленими – малі зміни вхідних параметрів призводять до істотних похибок при розв'язанні (рис. 5.1, в).

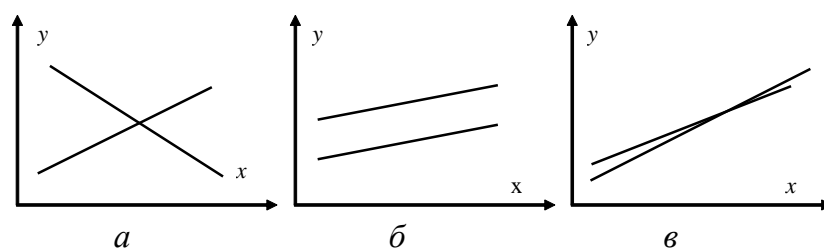


Рисунок 5.2 – Геометрична інтерпретація розв'язку системи двох лінійних рівнянь

Методи розв'язку систем лінійних рівнянь діляться на дві групи: прямі та ітераційні.

Прямі методи використовують кінцеві співвідношення (формули) для обчислення невідомих і дають рішення після виконання відомого числа операцій. Дані методи прості та універсальні, але мають ряд недоліків:

1) вимагають зберігання в оперативній пам'яті ЕОМ відразу всієї матриці, яка при розв'язку рівнянь в частинних похідних може бути досить громіздкою, що нераціонально для вироджених матриць;

2) відбувається накопичення похибки обчислень в зв'язку з використанням на поточному етапі результатів попереднього етапу. Тому застосування таких методів при $n > 200$ і для систем, у яких $D \approx 0$, важко.

Ітераційні методи – методи послідовних наближень. Характерною особливістю цих методів є завдання початкового значення невідомих, яке

уточнюється шляхом подальших ітерацій. У деяких випадках ітераційні методи кращі, тому що вони позбавлені недоліків прямих методів, проте мають свої недоліки:

- 1) алгоритми розв'язання задачі зазвичай складніші;
- 2) обсяг обчислення заздалегідь визначити важко;
- 3) збіжність методу може бути дуже повільною.

Існують також змішані алгоритми, в яких ітераційні методи використовуються в якості уточнюючих після розв'язку системи лінійних рівнянь прямими методами.

5.2. Прямі методи

5.2.1. Метод визначників

Це один з можливих методів розв'язку систем рівнянь з n невідомими. Однак число операцій, в результаті яких буде отримано рішення, надзвичайно велике: $N \approx n \cdot n!$.

5.2.2. Метод Гауса

Метод заснований на приведенні матриці до верхнього трикутного вигляду (рис. 5.1, б) шляхом послідовного виключення невідомих. Ця операція триває до тих пір, поки в лівій частині останнього n -го рівняння залишиться лише один член з невідомим x_n (прямий хід методу Гауса). Зворотний хід методу Гауса полягає в послідовному обчисленні шуканих невідомих, починаючи з останнього рівняння.

Приклад. Вирішити систему рівнянь (5.2) методом Гауса:

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + a_{13} \cdot x_3 = b_1, \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + a_{23} \cdot x_3 = b_2, \\ a_{31} \cdot x_1 + a_{32} \cdot x_2 + a_{33} \cdot x_3 = b_3. \end{cases} \quad (3.2)$$

Для виключення x_1 з другого рівняння додамо до нього перше, помножене на коефіцієнт $-\frac{a_{21}}{a_{11}}$, а потім, помноживши перше рівняння на

дріб мінус $\frac{a_{31}}{a_{11}}$, додамо його до третього.

В результаті отримаємо

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + a_{13} \cdot x_3 = b_1, \\ a'_{22} \cdot x_2 + a'_{23} \cdot x_3 = b'_2, \\ a'_{32} \cdot x_2 + a'_{33} \cdot x_3 = b'_3, \end{cases} \quad (5.3)$$

$$\text{де } a'_{22} = a_{22} - a_{12} \cdot \frac{a_{21}}{a_{11}}; \quad a'_{23} = a_{23} - a_{31} \cdot \frac{a_{21}}{a_{11}}; \quad b'_2 = b_2 - b_1 \cdot \frac{a_{21}}{a_{11}};$$

$$a'_{32} = a_{32} - a_{12} \cdot \frac{a_{31}}{a_{11}}; \quad a'_{33} = a_{33} - a_{13} \cdot \frac{a_{31}}{a_{11}}; \quad b'_3 = b_3 - b_1 \cdot \frac{a_{31}}{a_{11}}.$$

Для виключення x_2 з третього рівняння помножимо друге на дріб мінус $\frac{a'_{32}}{a'_{22}}$ і складемо з третім рівнянням, тоді

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + a_{13} \cdot x_3 = b_1, \\ a'_{22} \cdot x_2 + a'_{23} \cdot x_3 = b'_2, \\ a''_{33} \cdot x_3 = b''_3, \end{cases} \quad (5.4)$$

$$\text{де } a''_{33} = a'_{33} - a'_{23} \cdot \frac{a'_{32}}{a'_{22}}; \quad b''_3 = b'_3 - b'_2 \cdot \frac{a'_{32}}{a'_{22}}.$$

Прямий хід методу Гауса на цьому закінчується. На даному етапі виконувався поділ на коефіцієнти, які стоять на головній діагоналі, тому вони повинні бути відмінні від нуля. В іншому випадку необхідна перестановка рівнянь.

Зворотний хід методу Гауса починається з розв'язку останнього рівняння системи. Потім, на підставі знайденого значення x_3 , послідовно визначаються інші невідомі:

$$\begin{cases} x_3 = \frac{b''_3}{a''_{33}}, \\ x_2 = \frac{b'_2 - a'_{23} \cdot x_3}{a'_{22}}, \\ x_1 = \frac{b_1 - a_{12} \cdot x_2 + a_{13} \cdot x_3}{a_{11}}. \end{cases}$$

У додатку А наведено алгоритм розв'язку системи рівнянь методом Гауса.

На блок-схемі є такі позначення: i – номер рівняння, з якого

З отриманого співвідношення знаходимо x_2 :

$$x_2 = \frac{-c_2 \cdot x_3 + d_2 - a_2 \cdot B_1}{a_2 \cdot A_1 + b_2}.$$

Але, з іншої сторони, якщо $x_2 = A_2 \cdot x_3 + B_2$, тоді $A_2 = \frac{-c_2}{e_2}$;

$$B_2 = \frac{d_2 - a_2 \cdot B_1}{e_2},$$

де $e_2 = a_2 \cdot A_1 + b_2$.

Отже, для будь-якого $i = 2, 3, \dots, n-1$ маємо

$$A_i = \frac{-c_i}{e_i}, B_i = \frac{d_i - a_i \cdot B_{i-1}}{e_i}, e_i = a_i \cdot A_{i-1} + b_i. \quad (5.8)$$

Зворотній прогін полягає в послідовному обчисленні невідомих x_i по знайденим прогоночним коефіцієнтам. Спочатку необхідно визначити x_n . Для цього використовуємо значення x_{n-1} і останнє рівняння системи (5.5):

$$\begin{cases} x_{n-1} = A_{n-1} \cdot x_n + B_{n-1}, \\ a_n \cdot x_{n-1} + b_n \cdot x_n = d_n. \end{cases}$$

Тоді

$$x_n = \frac{d_n - a_n \cdot B_{n-1}}{b_n + a_n \cdot A_{n-1}}. \quad (5.9)$$

Використовуючи знайдені значення коефіцієнтів A_i , B_i , знайдемо x_i , починаючи з невідомого x_{n-1} . Виконання умови переважання діагональних коефіцієнтів в даній схемі виконується ($|b_i| \geq |a_i| + |c_i|$) і рішення стійке щодо похибок округлення, що дає можливість успішно застосовувати метод прогонки для систем рівнянь високого порядку. Блок-схема методу прогонки наведена в додатку А.

5.3. Ітераційні методи

Найбільш поширеним ітераційним методом вирішення систем лінійних рівнянь, що відрізняється своєю простотою, є метод Гауса-Зейделя.

В якості ілюстрації цього методу розглянемо систему лінійних рівнянь:

$$\begin{cases} a_{11} \cdot x_1 + a_{12} \cdot x_2 + a_{13} \cdot x_3 = b_1, \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + a_{23} \cdot x_3 = b_2, \\ a_{31} \cdot x_1 + a_{32} \cdot x_2 + a_{33} \cdot x_3 = b_3, \end{cases} \quad (5.10)$$

при умові, що $a_{11} \neq 0, a_{22} \neq 0, a_{33} \neq 0$; в іншому випадку необхідна перестановка коефіцієнтів.

Виразимо невідомі x_1, x_2, x_3 з системи (5.10):

$$\begin{cases} x_1 = \frac{1}{a_{11}} \cdot (b_1 - a_{12} \cdot x_2 - a_{13} \cdot x_3), \\ x_2 = \frac{1}{a_{22}} (b_2 - a_{21} \cdot x_1 - a_{23} \cdot x_3), \\ x_3 = \frac{1}{a_{33}} \cdot (b_3 - a_{31} \cdot x_1 - a_{32} \cdot x_2). \end{cases} \quad (5.11)$$

Задамо початкові наближення значень невідомих:

$$x_1 = x_1^0; x_2 = x_2^0; x_3 = x_3^0.$$

Тоді

$$\begin{cases} x_1^1 = \frac{1}{a_{11}} \cdot (b_1 - a_{12} \cdot x_2^0 - a_{13} \cdot x_3^0), \\ x_2^1 = \frac{1}{a_{22}} (b_2 - a_{21} \cdot x_1^1 - a_{23} \cdot x_3^0), \\ x_3^1 = \frac{1}{a_{33}} \cdot (b_3 - a_{31} \cdot x_1^1 - a_{32} \cdot x_2^1) \end{cases} \quad (5.12)$$

Використовуючи знайдені значення невідомих, можна побудувати схему ітераційного процесу:

$$\begin{cases} x_1^k = \frac{1}{a_{11}} \cdot (b_1 - a_{12} \cdot x_2^{k-1} - a_{13} \cdot x_3^{k-1}), \\ x_2^k = \frac{1}{a_{22}} (b_2 - a_{21} \cdot x_1^k - a_{23} \cdot x_3^{k-1}), \\ x_3^k = \frac{1}{a_{33}} \cdot (b_3 - a_{31} \cdot x_1^k - a_{32} \cdot x_2^k) \end{cases} \quad (5.13)$$

Ітераційний процес продовжується поки не буде виконано одну з умов:

$$\delta = \max_{i=1..n} |x_i^k - x_i^{k-1}| < \varepsilon$$

або

$$\Delta = \max_{i=1..n} \left| \frac{x_i^k - x_i^{k-1}}{x_i^k} \right| < \varepsilon,$$

де ε – заздалегідь задане мале число.

Для збіжності ітераційного процесу досить, щоб модулі діагональних коефіцієнтів становили не менше суми всіх інших коефіцієнтів:

$$|a_{i,i}| \geq \sum_{i \neq j} |a_{i,j}|.$$

Приклад. Вирішити систему рівнянь методом Гауса:

$$\begin{cases} 10x_1 - 7x_2 & = 7, \\ -3x_1 + 2x_2 + 6x_3 & = 4, \\ 5x_1 - x_2 + 5x_3 & = 6. \end{cases}$$

Виключимо невідомі x_2 з другого і третього рівнянь:

$$\begin{cases} 10x_1 - 7x_2 & = 7, \\ -0,1x_2 + 6x_3 & = 6,1, \\ 2,5x_2 + 5x_3 & = 2,5, \end{cases}$$

а потім x_3 з третього:

$$\begin{cases} 10x_1 - 7x_2 & = 7, \\ -0,1x_2 + 6x_3 & = 6,1, \\ 155x_3 & = 155. \end{cases}$$

Знайдемо розв'язок системи: $x_3 = 1$; $x_2 = 1$; $x_1 = 0$.

Приклад. Вирішити систему рівнянь методом Гауса-Зейделя:

$$\begin{cases} 4x_1 - x_2 + x_3 = 4, \\ 2x_1 + 6x_2 - x_3 = 7, \\ x_1 + 2x_2 - 3x_3 = 0. \end{cases}$$

Задамо початкові наближення невідомих: $x_1^0 = 0, x_2^0 = 0, x_3^0 = 0$.

Тоді

$$\begin{cases} x_1^1 = \frac{1}{4}(4 + x_2^0 - x_3^0) = 1, \\ x_2^1 = \frac{1}{6}(7 - 2x_1^1 + x_3^0) = \frac{5}{6}, \\ x_3^1 = \frac{1}{3}(x_1^1 + 2x_2^1) = \frac{8}{9}. \end{cases}$$

На наступному ітераційному циклі отримаємо

$$\begin{cases} x_1^2 = \frac{1}{4}(4 + x_2^1 - x_3^1) = \frac{71}{72}, \\ x_2^2 = \frac{1}{6}(7 - 2x_1^2 + x_3^1) = \frac{71}{72}, \\ x_3^1 = \frac{1}{3}(x_1^2 + 2x_2^2) = \frac{71}{72}. \end{cases}$$

Точний розв'язок системи рівнянь $x_3 = 1$; $x_2 = 1$; $x_1 = 1$.

Блок-схема методу наведена в додатку А. У додатку Б наведено приклад розв'язку системи лінійних рівнянь за допомогою функції *linsolve* бібліотеки *linalg* в разі, якщо матриця коефіцієнтів і права частина є дійсними або комплексними числами, а також розв'язок системи рівнянь в загальному вигляді. Будь-яке знайдене невідоме може бути отримане при зверненні до імені розв'язку системи як до елемента масиву, наприклад $X[1]$.

На закінчення необхідно відзначити, що існує досить велика кількість методів розв'язку систем лінійних рівнянь. Всі ці методи детально описані в спеціальній літературі.

6. НЕЛІНІЙНІ РІВНЯННЯ

6.1. Рівняння з одним невідомим

Рівняння даного типу в найзагальнішому вигляді можна записати $F(x)=0$, умовно розділивши його на два класи: алгебраїчні і трансцендентні (тригонометричні, показникові, логарифмічні). Рішенням рівняння є така величина x , яка при підстановці її в рівняння перетворює його в тотожність. Методи розв'язку нелінійних рівнянь діляться на прямі (застосовуються в обмеженому числі випадків) і ітераційні, що складаються у визначенні значення кореня шляхом послідовних наближень. Кожен крок послідовного наближення називається ітерацією. Якщо значення кореня, що визначається, сходиться до істинного значення, то процес називається збіжним.

6.1.1. Метод ділення відрізка навпіл (метод дихотомії)

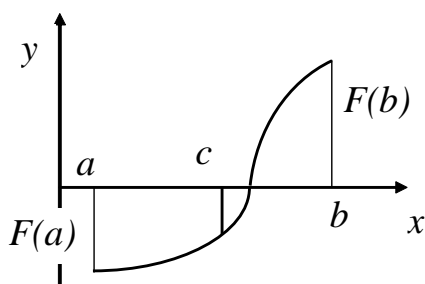


Рисунок 6.1 – Метод дихотомії

Нехай $[a, b]$ – інтервал локалізації кореня (рис. 6.1). В якості початкового наближення кореня рівняння виберемо величину $c = \frac{a+b}{2}$ і досліджуємо функцію на кінцях відрізків $[a, c]$ і $[c, b]$. Відрізок, на кінцях якого $F(x)$ приймає значення різних знаків, містить шуканий корінь; інший відрізок відкидаємо. В якості наступного наближення приймаємо середину нового інтервалу. Ітераційний процес продовжується до того, поки $|F(c)|$ не стане менше деякого заздалегідь заданого значення ε . Якщо правильно обраний інтервал локалізації кореня, то його значення буде знайдено (метод має властивість абсолютної збіжності). У додатку А наведена блок-схема методу дихотомії.

6.1.2. Метод хорд

Нехай $[a, b]$ – відрізок локалізації кореня (рис. 6.2). Тоді ітераційний

процес полягає в послідовному визначенні точок перетину хорди з віссю абсцис. Рівняння хорди має вигляд $\frac{y - F(a)}{F(b) - F(a)} = \frac{x - a}{b - a}$. Якщо $x=c$, то $y=0$,

тоді $c = a - \frac{b - a}{F(b) - F(a)} \cdot F(a)$. Порівнюючи значення функції на інтервалах

$[a, c]$ і $[c, b]$, вибираємо той інтервал, на якому значення функції має протилежні знаки.

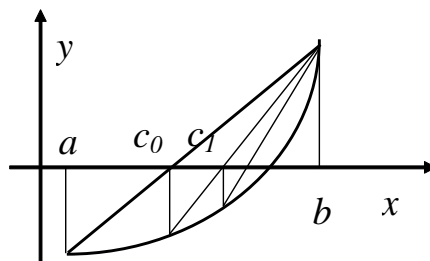


Рисунок 6.2– Метод хорд

Блок-схема методу аналогічна блок-схемі методу дихотомії з урахуванням різниці в обчисленні значення c . Необхідно змінити оператори обчислення функції на границях нових відрізків. Метод має властивість абсолютної збіжності (сходиться до значення кореня при правильно обраному інтервалі пошуку кореня).

6.1.3. Метод Ньютона

Даний метод полягає в знаходженні перетину дотичної до кривої, проведеної з деякої точки, з віссю абсцис (рис. 6.3). Рівняння дотичної, проведеної з точки c_0 має вигляд

$$y - F(c_0) = F'(c_0) \cdot (x - c_0).$$

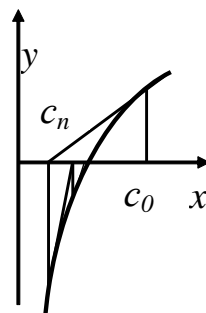


Рисунок 6.3 – Метод Ньютона

7. ЧИСЕЛЬНІ МЕТОДИ РОЗВ'ЯЗКУ ДИФЕРЕНЦІЙНИХ РІВНЯНЬ

Ряд завдань, пов'язаних з тепловими, електромагнітними і багатьма іншими процесами, зводиться до розв'язку диференційних рівнянь. Із загального курсу математичного аналізу відомо, що диференційні рівняння можна розділити на два типи:

1. Звичайні диференційні рівняння (ЗДР), що містять одну незалежну змінну.
2. Диференційні рівняння в частинних похідних, що містять кілька незалежних змінних.

7.1. Звичайні диференційні рівняння

У загальному вигляді звичайне диференційне рівняння може бути записане як

$$F(x, y, y', y'', \dots, y^n) = 0,$$

де x – незалежна змінна; y – шукана функція і її похідна.

Найвищий порядок величини n , що входить в рівняння похідної, називається порядком диференційного рівняння. Диференційне рівняння може бути записане у формі, розв'язаній відносно похідної вищого порядку:

$$y^n = F(x, y, y', y'', \dots, y^{n-1}).$$

Лінійними ДР називаються рівняння, які є лінійними щодо похідних і шуканої функції, наприклад: $y' - x^2 \cdot y = \sin(x)$.

Розв'язком ДР називається функція $y = f(x)$, яка при підстановці в ДР перетворює його в тотожність. Загальний розв'язок ДР n -го порядку містить n констант $y = f(x, C_1, \dots, C_n)$ для визначення яких необхідно задати n додаткових умов. Якщо ці умови задаються в одній точці, то така задача називається задачею Коші:

$$m \cdot x''(t) = F(t, x) - c \cdot x(t),$$

при $t = 0, x = x_0, x' = x'_0$.

Якщо умови задаються в декількох точках, то така задача називається крайовою.

Найбільш універсальним методом вирішення ДР є метод кінцевих різниць, що полягає в дискретизації області зміни неперервного аргументу безліччю точок. Такі точки називаються вузлами і утворюють різницеву сітку. Шукана функція y в цьому випадку називається сітковою функцією. Рівняння, записане в дискретних вузлах сітки, називається кінцево-різницевою схемою. Під стійкістю схеми розуміють неперервну залежність її розв'язку від вхідних даних. Під коректністю розуміють існування єдиного рішення за будь-яких вхідних даних.

Постановка задачі Коші може бути сформульована в наступному вигляді. Знайти функцію $y = f(x)$ що задовольняє диференційному рівнянню $\frac{dy}{dx} = f(x, y)$ з додатковими (початковими або граничними) умовами $x = x_0; y = y_0$.

Методи вирішення такої задачі поширюються на рівняння вищих порядків і систем ДР.

Приклад. Рівняння $z'' = \varphi(x, z, z')$ можна представити у вигляді

$$\begin{cases} y' = \varphi(x, z, y), \\ z' = y. \end{cases}$$

Для вирішення задачі Коші виділимо послідовність точок x_i , розташованих на відстані одна від одної: $h_i = x_i - x_{i-1}$. Нехай значенню x_i відповідає значення y_i . Тоді безліч точок $\{x_i, y_i\}$ є сітковою функцією. Замінюючи значення похідної встановленням кінцевих різниць, можна записати:

$$y_{i+1} = f(x_i, h_i, y_i, y_{i+1}, y_{i-(k-1)}),$$

де $i=1, 2, \dots, y_{i=0}=y_0$.

Якщо в правій частині рівняння відсутнє значення y_{i+1} , тоді схема називається явною.

При $k=1$ отримаємо однокроковий метод розв'язку:

$$y_{i+1} = f(x_i, h_i, y_i, y_{i+1}, y_i).$$

При $k=2$ – двохкроковий метод розв'язку: $y_{i+1} = f(x_i, h_i, y_i, y_{i+1}, y_{i-1})$.

7.2. Однокрокові методи

7.2.1. Метод Ейлера

Метод заснований на розкладанні функції в ряд Тейлора в околиці

вузлів $x=x_i$. Вважаємо, що вузли розташовані на однаковій відстані h один від одного. Тоді диференційне рівняння $\frac{dy}{dx} = f(x, y)$ можна записати так:

$$y_{i+1} = y_i + h \cdot f(x_i, y_i), \quad (7.1)$$

при початкових умовах $x = x_0$, $y = y_0$, $i = 0, 1, \dots, n-1$.

Якщо всі результати розрахунку необхідно зберігати в пам'яті, то x_i , y_i задаються у вигляді масиву.

Блок-схема розв'язку диференційного рівняння методом Ейлера наведена в додатку А. Геометрична інтерпретація розв'язку диференційного рівняння методом Ейлера показана на рис. 7.1.

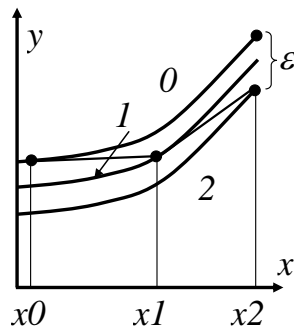


Рисунок 7.1. – Графічна ілюстрація методу Ейлера

Інтегральні криві 0... 2 є розв'язком диференційного рівняння методом Ейлера при різних початкових умовах. Нульова лінія відповідає точному розв'язку. Точки на лініях 1 і 2 отримані в результаті наближеного чисельного розв'язку і характеризують похибку методу ε , рівною $O(h)$ (див. рис. 7.1).

7.2.2. Розрахункові схеми другого порядку точності. Модифікований метод Ейлера

Припустимо, що знайдено досить точне значення сіткової функції на кроці $i + 1$, яке визначено як

$$\hat{y}_{i+1} = y_i + h \cdot f(x_i, y_i).$$

Тоді уточнене значення y_{i+1} можна записати у вигляді

$$y_{i+1} = y_i + \frac{h}{2} \cdot [f(x_i, y_i) + f(x_{i+1}, \hat{y}_{i+1})], \quad (7.2)$$

$$y_{i+1} = y_i + \frac{h}{2} \cdot [f(x_i, y_i) + f(x_{i+1}, y_i + h \cdot f(x_i, y_i))],$$

де h – крок інтегрування.

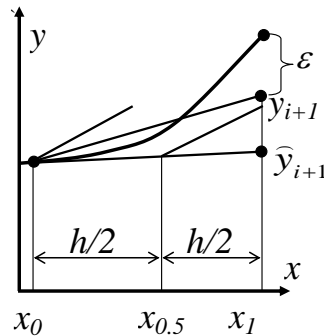


Рисунок 7.2 – Модифікований метод Ейлера

Геометрична інтерпретація модифікованого методу Ейлера представлена на рис. 7.2. За допомогою даного методу можна побудувати схему з контролем точності одержуваних результатів шляхом порівняння y_{i+1} і \hat{y}_{i+1} . Якщо $|y_{i+1} - \hat{y}_{i+1}| > \varepsilon$, то крок h необхідно зменшити.

Приклад. Вирішити ЗДР $y'' = 2 \cdot (x^2 + y)$ при початкових умовах $x = 0$, $y = 0$. Число кроків інтегрування дорівнює десяти. Порівняти отримані розв'язки різними методами.

Розв'язок методом Ейлера дає значення $y = 0.8555$ при $x = 1$. Розв'язок модифікованим методом Ейлера дає значення $y = 1.1810$ при $x = 1$. Точне значення становить $y = 1.19452771$ при $x = 1$. Отже, помилка методу Ейлера залежить від кроку інтегрування і від виду шуканої функції і може бути досить велика.

7.2.3. Розрахункові схеми другого порядку точності (неявні схеми). Ітераційні методи

Розрахункова формула методу має вигляд

$$y_{i+1} = y_i + \frac{h}{2} \cdot [f(x_i, y_i) + f(x_{i+1}, y_{i+1})] \quad (7.3)$$

або

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

Схеми (7.3) мають другий порядок точності, однак присутність у правій частині шуканої величини y_{i+1} робить дану схему неявною і тому для її вирішення необхідні ітераційні методи.

7.2.4. Метод Рунге-Кутта

Даний метод є найбільш поширеним методом інтегрування диференціальних рівнянь і систем ДР. Помилка методу $O(h^4)$. Значення функції на кроці $i+1$ визначається за такими формулами:

$$y_{i+1} = y_i + \frac{h}{6} \cdot (k_0 + 2 \cdot k_1 + 2 \cdot k_2 + k_3), \quad (7.4)$$

де $k_0 = f(x_i, y_i)$; $k_1 = f\left(x_i + \frac{h}{2}, y_i + h \cdot \frac{k_0}{2}\right)$; $k_2 = f\left(x_i + \frac{h}{2}, y_i + h \cdot \frac{k_1}{2}\right)$;
 $k_3 = f(x_i + h, y_i + h \cdot k_2)$.

Дана схема вимагає чотирьохкратного обчислення функції в кожній точці. Однак для досягнення заданої точності вона допускає багато більший крок в порівнянні з іншими кінцево-різницевиими схемами.

Приклад. Вирішити ОДР методом Рунге-Кутта $y'' = 2 \cdot (x^2 + y)$ при таких початкових умовах: $x=0, y=0$. Кількість кроків інтегрування дорівнює десяти. Розв'язок дає результат $y=1.1945$ при $x=1$, що співпадає з точним розв'язком до четвертого знаку після коми.

Точність розв'язку диференціального рівняння різними методами представлена в табл. 7.1.

Таблиця 7.1 – Точність розв'язку ДР різними методами.

x_i	Метод Ейлера	Модифікований метод Ейлера	Метод Рунге-Кутта	Точний розв'язок
1.0	0.8555	1.1810	1.1945	1.1945

Метод Ейлера при даному кроці інтегрування дає помилку приблизно 28%. При зменшенні кроку помилка зменшується, але при цьому зростає час розрахунку. Крім того, значне зменшення кроку може позначитись на точності результату за рахунок помилок округлення чисел.

Розглянуті вище методи можуть застосовуватись для вирішення систем рівнянь, до яких призводить розв'язок диференціальних рівнянь високого

порядку. Як приклад, розглянемо рішення системи з двох диференціальних рівнянь методом Рунге-Кутта:

$$\begin{cases} \frac{dy}{dx} = \varphi(x, y, z), \\ \frac{dz}{dx} = \psi(x, y, z), \end{cases} \quad (7.5)$$

при початкових умовах $y(x_0) = y_0$, $z(x_0) = z_0$.

Тоді

$$\begin{cases} y_{i+1} = y_i + \frac{h}{6} \cdot (k_0 + 2k_1 + 2k_2 + k_3), \\ z_{i+1} = z_i + \frac{h}{6} \cdot (l_0 + 2l_1 + 2l_2 + l_3), \end{cases}$$

де $k_0 = \varphi(x_i, y_i, z_i)$, $l_0 = \psi(x_i, y_i, z_i)$;

$$k_1 = \varphi\left(x_i + \frac{h}{2}, y_i + h \cdot \frac{k_0}{2}, z_i + h \cdot \frac{l_0}{2}\right), l_1 = \psi\left(x_i + \frac{h}{2}, y_i + h \cdot \frac{k_0}{2}, z_i + h \cdot \frac{l_0}{2}\right);$$

$$k_2 = \varphi\left(x_i + \frac{h}{2}, y_i + h \cdot \frac{k_1}{2}, z_i + h \cdot \frac{l_1}{2}\right), l_2 = \psi\left(x_i + \frac{h}{2}, y_i + h \cdot \frac{k_1}{2}, z_i + h \cdot \frac{l_1}{2}\right);$$

$$k_3 = \varphi(x_i + h, y_i + h \cdot k_2, z_i + h \cdot l_2), l_3 = \psi(x_i + h, y_i + h \cdot k_2, z_i + h \cdot l_2);$$

$$i = 0 \dots n.$$

7.3. Крайові задачі

На практиці часто доводиться вирішувати завдання, коли додаткові умови вирішення диференційного рівняння задаються на кінцях відрізків незалежної змінної. Такі завдання називаються крайовими. Вирішення зазначених завдань полягає у визначенні на інтервалі $[a, b]$ функції $y=f(x)$, що задовольняє граничним умовам $y(a)=A$, $y(b)=B$ та перетворює у тотожність диференційне рівняння

$$y'' + p(x) \cdot y' + q(x) \cdot y = f(x). \quad (7.6)$$

У найзагальнішому вигляді граничні умови можна записати в вигляді

$$\begin{cases} \alpha_1 \cdot y(a) + \beta_1 \cdot y'(a) = A, \\ \alpha_2 \cdot y(b) + \beta_2 \cdot y'(b) = B. \end{cases}$$

Розділимо інтервал $[a, b]$ на деяку кількість рівних відрізків h та будемо шукати розв'язок диференційного рівняння (7.6) у вузлах отриманої сітки $x_i = a + i \cdot h$, $i = 0 \dots n$, $a \leq x_i \leq b$.

У кінцево-різницевій формі рівняння (7.6) запишемо для внутрішніх вузлів сітки:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p(x_i) \cdot \frac{y_{i+1} - y_{i-1}}{2h} + q(x_i) \cdot y_i = f(x_i). \quad (7.7)$$

В результаті отримаємо систему рівнянь щодо невідомих значень y для внутрішніх вузлів:

$$F(x_i, y_{i-1}, y_i, y_{i+1}) = 0; i = 2..n-1.$$

Для граничних вузлів використовуються умови

$$\begin{cases} \alpha_1 \cdot y(0) + \beta_1 \cdot y'(0) = A, \\ \alpha_2 \cdot y(1) + \beta_2 \cdot y'(1) = B. \end{cases}$$

Найпростіша кінцево-різницева апроксимація буде виглядати наступним чином

$$\begin{cases} \alpha_1 \cdot y_0 + \beta_1 \cdot \frac{y_1 - y_0}{h} = A, \\ \alpha_2 \cdot y_n + \beta_2 \cdot \frac{y_n - y_{n-1}}{h} = B. \end{cases} \quad (7.8)$$

Отримана система рівнянь має трьох діагональну матрицю і вирішується методом прогонки.

7.4. Розв'язок нелінійних звичайних диференціальних рівнянь другого порядку

У загальному вигляді нелінійне диференціальне рівняння другого порядку може бути записане $y'' = f(x, y)$, або в кінцево-різницевій формі як:

$$y_{i-1} - 2y_i + y_{i+1} = h^2 \cdot f(x_i, y_i), \quad (7.9)$$

де $0 \leq x \leq 1$, $y(0) = A$; $y(1) = B$.

Диференціальне рівняння (7.9) розв'язується ітераційними методами за допомогою лінеаризації, тобто зведення розв'язку нелінійного диференціального рівняння до розв'язку послідовності систем лінійних алгебраїчних рівнянь.

Нехай знайдено розв'язок рівняння на k -й ітерації; тоді, підставляючи знайдене значення $y^{(k)}$ в праву частину рівняння (7.9), отримаємо

$$y^{(k+1)}_{i-1} - 2y^{(k+1)}_i + y^{(k+1)}_{i+1} = h^2 \cdot f(x_i, y^{(k)}_i). \quad (7.10)$$

Вирішення (7.10) продовжується, поки не буде виконана умова

$$\left| y_i^{(k+1)} - y_i^{(k)} \right| \leq \varepsilon, \text{ при } i=1..n,$$

де ε – деяка мала величина.

Матриця системи (7.10) трьох діагональна та розв'язок може бути отримано методом прогонки. Збіжність такого ітераційного методу достатньо повільна.

В додатку наведені приклади розв'язку звичайних диференційних рівнянь та систем диференційних рівнянь в загальному вигляді та чисельним методом в системі MAPLE.

7.5. Диференційні рівняння в частинних похідних

У багатьох практичних завданнях шукані функції залежать від декількох змінних. Диференційні рівняння, що описують такі завдання, містять часткові похідні по шуканим функціям. Ці рівняння називаються рівняннями в частинних похідних (рівняння потенціалу, поширення тепла, дифузії). Для їх вирішення необхідно задати додаткові умови. Це можуть бути граничні (крайові умови) для просторових змінних, якщо рішення шукається в обмеженій області. Такі завдання називаються крайовими. Якщо однією з незалежних змінних є час, то задається деяка умова в початковий момент часу, яка називається початковою умовою. Задачі з початковими і граничними умовами називаються змішаними.

Найбільш широко поширеними методами розв'язку таких рівнянь є кінцево-різницеві методи, засновані на введенні просторово-часової або просторової сітки в даній області розв'язку задачі.

В даному курсі розглянуті рівняння, що описують процеси дифузії, поширення тепла і рівняння Лапласа:

$$\frac{\partial U}{\partial t} = a^2 \cdot \sum_l \frac{\partial^2 U}{\partial x_l^2}; \quad a > 0; \quad (7.11)$$

$$\sum_l \frac{\partial^2 U}{\partial x_l^2} = 0.$$

Розв'язок диференційних рівнянь в частинних похідних заснований на апроксимації рівняння (7.11) його кінцево-різницевим аналогом, побудованим на просторово-часовій або просторовій сітці (рис. 7.3). Значення шуканої функції U визначається у вузлових точках сітки.

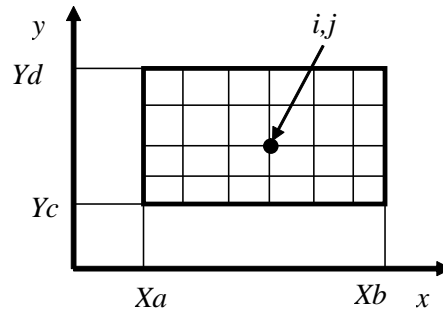


Рисунок 7.3 –
Координатна сітка

Розглянемо деяку область $G(x,y)$ і введемо на ній рівномірну прямокутну сітку з кроком h_1 і h_2 , утворену перетином прямих ліній $x_i = Xa + h_1 \cdot i$; $y_j = Yc + h_2 \cdot j$, які утворюють вузли з номерами i, j . Аналогічним чином вводиться сітка і для тривимірних областей.

В даному курсі розглядаються тільки прямокутні сітки, хоча є сітки трикутні і шестикутні. Вузли, що лежать на границях даної області рішення, називаються граничними. Всі інші вузли називаються внутрішніми.

7.5.1. Одновимірне рівняння теплопровідності

Рівняння вигляду

$$\frac{\partial U}{\partial t} = a^2 \cdot \frac{\partial^2 U}{\partial x^2} \quad \text{при } 0 \leq x \leq 1; t > 0, \quad (7.12)$$

з початковою умовою $U(x,0) = \phi(x)$ та граничними умовами

$$\begin{cases} U(0,t) = \psi_1(t) \\ U(1,t) = \psi_2(t) \end{cases}$$

описує одновимірний процес поширення тепла.

Введемо в області вирішення завдання прямокутну просторово-часову сітку:

$$x_i = i \cdot h; t_j = j \cdot \tau,$$

де τ – крок по часовій координаті; h – крок по просторовій координаті.

Значення шуканої функції у вузлах сітки позначимо як U_i^j і запишемо вихідні рівняння на двох шаблонах (рис. 7.4).

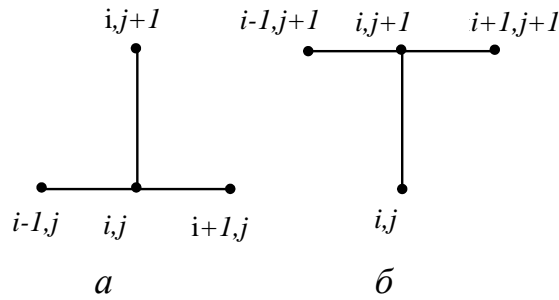


Рисунок 7.4. – Розрахункові схеми розв’язку рівнянь теплопровідності:

a – явна схема; *б* – неявна схема

Рівняння 7.12 для шаблонів *a*, *б* (див. рис. 7.4) запишуться наступним чином:

$$\frac{U_i^{j+1} - U_i^j}{\tau} = a^2 \cdot \frac{U_{i+1}^j - 2U_i^j + U_{i-1}^j}{h^2}; \quad (7.13)$$

$$\frac{U_i^{j+1} - U_i^j}{\tau} = a^2 \cdot \frac{U_{i+1}^{j+1} - 2U_i^{j+1} + U_{i-1}^{j+1}}{h^2},$$

де $i = 1..N - 1; j = 0, 1, \dots$

Для явної схеми (див. рис. 7.4 *a*) значення функції на $j+1$ часовому шарі визначається з формули

$$U_i^{j+1} = \lambda \cdot U_{i+1}^j + (1 - 2 \cdot \lambda) \cdot U_i^j + \lambda \cdot U_{i-1}^j, \quad (7.14)$$

де $\lambda = \frac{a^2 \cdot \tau}{h^2}; i = 1, 2..N - 1$.

Дана схема сходиться до вирішення вихідної задачі зі швидкістю $O(h^2 + \tau)$. Вона стійка при виконанні умови $\lambda = \frac{a^2 \cdot \tau}{h^2} \leq \frac{1}{2}$, що накладає обмеження на просторово-часовий крок.

Для неявної схеми (див. рис. 7.4 *б*) визначення функції на $j+1$ часовому кроці призводить до необхідності розв’язку системи лінійних алгебраїчних рівнянь:

$$\lambda \cdot U_{i-1}^{j+1} - (1 + 2 \cdot \lambda) \cdot U_i^{j+1} + \lambda \cdot U_{i+1}^{j+1} = -U_i^j, \quad (7.15)$$

де $i = 1..N - 1; j = 0, 1, \dots$

Дана схема має трьох діагональну матрицю і може бути вирішена щодо функції на $j+1$ часовому шарі методом прогонки. Неявна схема безумовно стійка і не накладає обмежень на просторово-часовий крок, значення якого вибирається тільки з міркування точності одержуваного розв'язку. Схеми 7.14 і 7.15 необхідно доповнити початковими і граничними умовами

$$U_i^0 = \varphi(x_i), U_0^j = \psi_1(t_j), U_N^j = \psi_2(t_j). \quad (7.16)$$

В граничні умови (7.16) може також входити і значення похідної шуканої функції.

7.5.2. Багатовимірні рівняння теплопровідності

Багатовимірними рівняннями теплопровідності називаються рівняння, що залежать більше ніж від однієї просторової змінної:

$$\frac{\partial U}{\partial t} = a^2 \cdot \left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} \right), \quad (7.17)$$

де $U(x, y, 0) = \varphi(x, y)$, а граничні умови мають вигляд

$$\begin{cases} U(0, y, t) = \psi_1(t), U(1, y, t) = \psi_2(t), \\ U(x, 0, t) = \psi_3(t), U(x, 1, t) = \psi_4(t). \end{cases}$$

Розв'язок таких рівнянь може бути здійснено різними методами, проте найбільш універсальними є локально-одновимірні схеми. Побудова локально-одновимірних схем заснована на введенні на кожному часовому кроці проміжного етапу. Кількість етапів залежить від кількості просторових координат. На кожному етапі здійснюється одновимірна апроксимація диференційного рівняння в частинних похідних по одній з просторових змінних. Отже, багатовимірна задача розщеплюється на послідовність одновимірних задач по кожній з просторових координат. Такі схеми мають властивість сумарної апроксимації, тобто отримані значення функції на проміжному етапі не є розв'язком рівняння.

Для двовимірного рівняння теплопровідності кінцево-різницева локально-одновимірна схема може бути записана так:

$$\begin{cases} \frac{U_{i,j}^{k+0,5} - U_{i,j}^k}{a^2 \cdot \tau} = \frac{U_{i+1,j}^{k+0,5} - 2 \cdot U_{i,j}^{k+0,5} + U_{i-1,j}^{k+0,5}}{h^2}, \\ \frac{U_{i,j}^{k+1} - U_{i,j}^{k+0,5}}{a^2 \cdot \tau} = \frac{U_{i,j+1}^{k+1} - 2 \cdot U_{i,j}^{k+1} + U_{i,j-1}^{k+1}}{h^2}, \end{cases} \quad (7.18)$$

де h – крок по просторовим змінним.

Рівняння (7.18) необхідно доповнити початковими і граничними умовами. Дана схема являє собою неявну схему по кожному з напрямків. Вирішується система рівнянь (7.18) методом прогонки на кожному часовому шарі.

7.5.2. Рівняння Лапласа та Пуассона

Багато стаціонарних фізичних завдань (задачі теплопровідності, дифузії, потенціалу і т. ін.) зводяться до вирішення рівняння Лапласа:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0. \quad (7.19)$$

Розв'язок цього рівняння будемо шукати для деякої області G , обмеженої замкнутою лінією L . Тоді для повного формулювання завдання Лапласа необхідно задати умову на границі:

$$U(x, y)_L = \varphi(x, y). \quad (7.20)$$

Граничні умови можуть бути задані і в тому випадку, коли межа L буде являти собою сукупність окремих примітивів, на яких задаються як значення шуканої функції, так і значення її похідної.

Для вирішення рівняння (7.19) з умовою (7.20) апроксимуємо вихідне рівняння на прямокутній кінцево-різницевої сітці, взявши для простоти однаковий крок h і замінивши значення функції у вузлах сітки значеннями сіткової функції $U_{i,j}$ на шаблоні рис. 7.5.

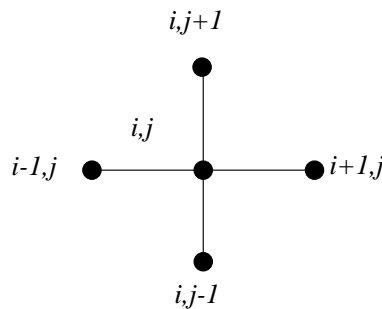


Рисунок 7.5 – Шаблон рівняння Лапласа

Тоді отримаємо різницеве рівняння

$$\frac{U_{i+1,j} - 2 \cdot U_{i,j} + U_{i-1,j}}{h^2} + \frac{U_{i,j+1} - 2 \cdot U_{i,j} + U_{i,j-1}}{h^2} = 0, \quad (7.21)$$

де $i = 1, 2, \dots, I - 1$; $j = 1, 2, \dots, J - 1$.

Значення сіткової функції в граничних вузлах може бути знайдене з граничних умов

$$U_{0,j} = \varphi(x_0, y_j), U_{I,j} = \varphi(x_I, y_j), j = 0, 1, \dots, J;$$

$$U_{i,0} = \varphi(x_i, y_0), U_{i,J} = \varphi(x_i, y_J), i = 0, 1, \dots, I.$$

Одним з найбільш поширених методів розв'язку рівняння (7.21) є ітераційний метод. Для цього перепишемо рівняння (7.21) у вигляді

$$U_{i,j} = \frac{1}{4} \cdot (U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1}).$$

Поставивши собі за початковим наближенням $U_{i,j}$. Будемо уточнювати його в процесі вирішення.

Рівняння Лапласа (7.19) є окремим випадком рівняння Пуансона:

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = f(x, y). \quad (7.22)$$

Розрахункова формула для розв'язку рівняння (7.22) ітераційним методом має вигляд

$$U_{i,j} = \frac{1}{4} \cdot (U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1} - h^2 \cdot f(x_i, y_j)).$$

У додатку А наведена блок-схема розв'язку рівняння Лапласа. Збіжність ітераційного процесу контролюється змінною M для двох послідовних ітерацій. Схема може бути використана і для вирішення рівняння (7.22).

У додатку Б наведено розв'язок рівняння теплопровідності системою MAPLE в чисельному вигляді. У прикладі вирішується одновимірне рівняння теплопровідності з внутрішніми джерелами тепла при адіабатичному режимі нагріву структури, що складається з трьох різних тіл. Кожне тіло має своє значення теплопровідності, теплоємності і своє джерело тепла (завдання різних властивостей забезпечує функція *piecewise*). Щільність струму j задана і є функцією часу. Розв'язок наводиться в просторовій, просторово-часовій та анімаційній (можливо тільки в прикладній програмі) формах.

У додатку Б наведено рішення завдання на прикладі теорії пружності – удару по одному з кінців довгого металевго стрижня.

Розглянуті програми ілюструють можливості системи MAPLE стосовно розв'язку диференціальних рівнянь в частинних похідних.

обчислюється тільки нова точка x_3 . Оскільки $f(x_3) > f(x_1)$, то мінімум знаходиться на відрізку $[x_3, b_1]$. Позначимо цей відрізок як $[a_2, b_2]$ і знову вибираємо точку в середині інтервалу.

3. Процес оптимізації повторюється до тих пір, поки довжина інтервалу невизначеності $[a_n, b_n]$ не стане менше заданої величини ε .

Найбільш важливим елементом в алгоритмі є розміщення внутрішніх точок на кожному відрізку $[a_k, b_k]$. Можна показати, що оптимальними будуть наступні значення координат точок y, z поділу відрізка $[a_k, b_k]$ на $k+1$ кроці оптимізації ($y < z$):

$$\begin{cases} y = 0,618 \cdot a_k + 0,382 \cdot b_k, \\ z = 0,382 \cdot a_k + 0,618 \cdot b_k. \end{cases} \quad (8.4)$$

Довжина інтервалу невизначеності буде $d_k = b_k - a_k = 0,618^k \cdot d_0$, де d_0 – початкова довжина інтервалу невизначеності.

Процес оптимізації закінчується при виконанні умови $d_k < \varepsilon$. При цьому проектний параметр оптимізації можна прийняти наступним:

$$x = \frac{a_k + b_k}{2}.$$

В додатку А наведена блок-схема одновимірної оптимізації методом золотого перетину, а в додатку Б – прикладна програма пошуку мінімуму функції в системі MAPLE.

8.2. Багатовимірна оптимізація

Як правило, функція цілі нелінійно залежить від оптимізаційних параметрів. Тому для пошуку екстремуму необхідно використовувати методи нелінійного програмування. Розглянемо найбільш розповсюджені методи пошуку екстремуму для аналітично заданої цільової функції.

1. Класичний метод пошуку екстремуму багатовимірної функції. Припустимо, що є цільова функція, яка задана в наступному вигляді: $U = f(x_1, \dots, x_n)$. У цьому випадку задача зводиться до необхідності вирішення системи з n нелінійних рівнянь: $\frac{\partial U(x_1, \dots, x_n)}{\partial x_i} = 0$. Такий метод не

завжди зручний, так як часто отримане значення параметрів неможливо реалізувати. Крім того, розв'язок нелінійної системи рівнянь високого

порядку може викликати великі труднощі.

2. Метод перебору (сканування). Полягає у визначеності значення функції в деяких визначених вузлових точках. З множини отриманих значень цільової функції вибирається те, в якому значення функції має екстремум. Даний метод найбільш простий, проте для отримання прийнятної точності кількість обчислень може бути дуже великою.

3. Метод покоординатного спуску. Полягає в послідовній оптимізації цільової функції по кожній з координат при фіксації інших. Нехай необхідно визначити екстремум функції $U = f(x_1, \dots, x_n)$. В якості початкового наближення виберемо точку $M_0(x_1^0, \dots, x_n^0)$. Зафіксуємо всі координати, крім першої, і вирішимо задачу одновимірної оптимізації по x_1 для функції $U = f(x_1, \dots, x_n^0)$. Знайдемо значення екстремуму по напрямку x_1 в x_1^1 (точка $M_1(x_1^1, \dots, x_n^0)$). Зафіксуємо всі координати, крім x_2 і продовжимо пошук екстремуму для функції $U = f(x_1^1, x_2, \dots, x_n^0)$ по параметру x_2 . В результаті отримаємо значення x_2^1 (точка $M_2(x_1^1, x_2^1, \dots, x_n^0)$). Завершивши оптимізацію по x_n , вертаємось до початку процесу. В результаті отримаємо ряд точок M_0, \dots, M_n , в яких функція спадає. На будь-якому k -ому кроці процес можна перервати. Тоді $U = f(M_k)$ приймається за екстремум. Даний метод дозволяє звести задачу багатовимірної оптимізації до послідовності задач одновимірної оптимізації.

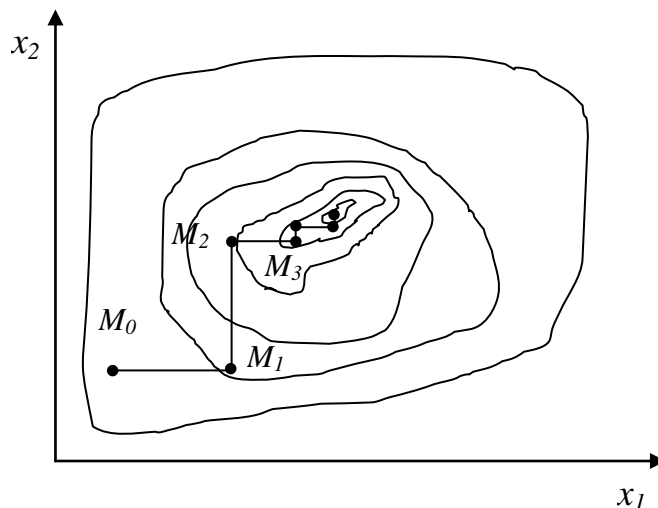


Рисунок 8.1 – Метод покоординатного спуску

На рис. 8.1 показана геометрична ілюстрація методу для двох параметрів. Блок-схема методу наведена в додатку А.

5. Метод градієнтного спуску. Він полягає в тому, що з деякої початкової точки робиться крок в напрямі, який є протилежним градієнту. Далі обчислюється нове значення градієнта, і процес повторюється. Процес продовжується до отримання найменшого значення цільової функції.

Одним з різновидів градієнтного методу є метод найскорішого спуску. Він полягає у тому, що в напрямі, протилежному градієнту, робиться не один, а декілька кроків, поки функція спадає. При невиконанні цієї умови обчислюється градієнт і новий напрям пошуку екстремуму.

Градієнтний метод і метод найскорішого спуску дозволяють вирішити задачу оптимізації за мінімальний час. Проте ці методи не позбавлені недоліків. У випадку належності у функції цілі декількох екстремумів, такі методи не гарантують знаходження глобального екстремуму. Тому для гарантованого визначення глобального екстремуму задачу оптимізації вирішують декілька разів з різними значеннями початкових параметрів оптимізації. Найменше (найбільше) отримане значення приймається за екстремальне значення. Відповідно, впевнитись в наявності глобального екстремуму можна тільки у випадку інформації по всій поверхні функції відклику в досліджуваній області.

6. Метод Монте-Карло. Пошук глобального екстремуму може бути здійснений методом Монте-Карло. Алгоритм визначення глобального екстремуму функції полягає в наступному:

1. По кожному напрямі в області пошуку вирішення $(a_i \dots b_i)$ в n -вимірному просторі рівномірно розподіляється за допомогою генератора випадкових величин деяке число точок.

2. З усієї множини точок вибираються ті, при яких цільова функція має екстремальне значення.

3. Отримана точка вибирається в якості екстремуму цільової функції.

4. У випадку, якщо точність отриманого рішення задовільна, процес закінчується, а якщо ні, то число випадкових точок збільшується, або звужуються границі пошуку екстремуму (рис. 8.2).

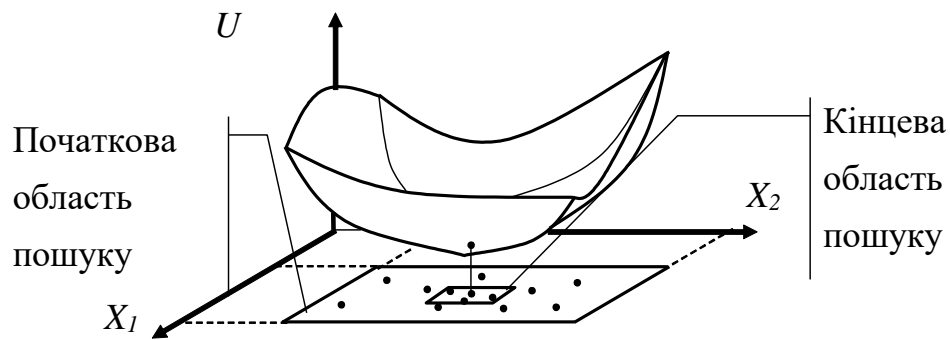


Рисунок 8.2 – Ілюстрація методу пошуку мінімуму цільової функції

Для використання стандартних програм по генерації випадкової величини змінні x_i повинні бути перетворені у відповідності з формулою

$$\begin{aligned} x_i &= a_i + (b_i - a_i) \cdot \eta_i, \\ U &= f(\eta_1, \eta_2, \dots, \eta_n), \end{aligned} \quad (8.5)$$

де η_i – незалежні випадкові величини, які рівномірно розподілені на інтервалі 0-1.

Перевагами такого алгоритму є:

- простота реалізації;
- отримання інформації про всю поверхню цільової функції;
- визначення глобального екстремуму в області пошуку рішень.

В додатку А наведені алгоритми деяких оптимізаційних методів, а в додатку Б надане вирішення задачі пошуку екстремуму для функції Розенброка системою MAPLE і визначення екстремуму методом Монте-Карло.

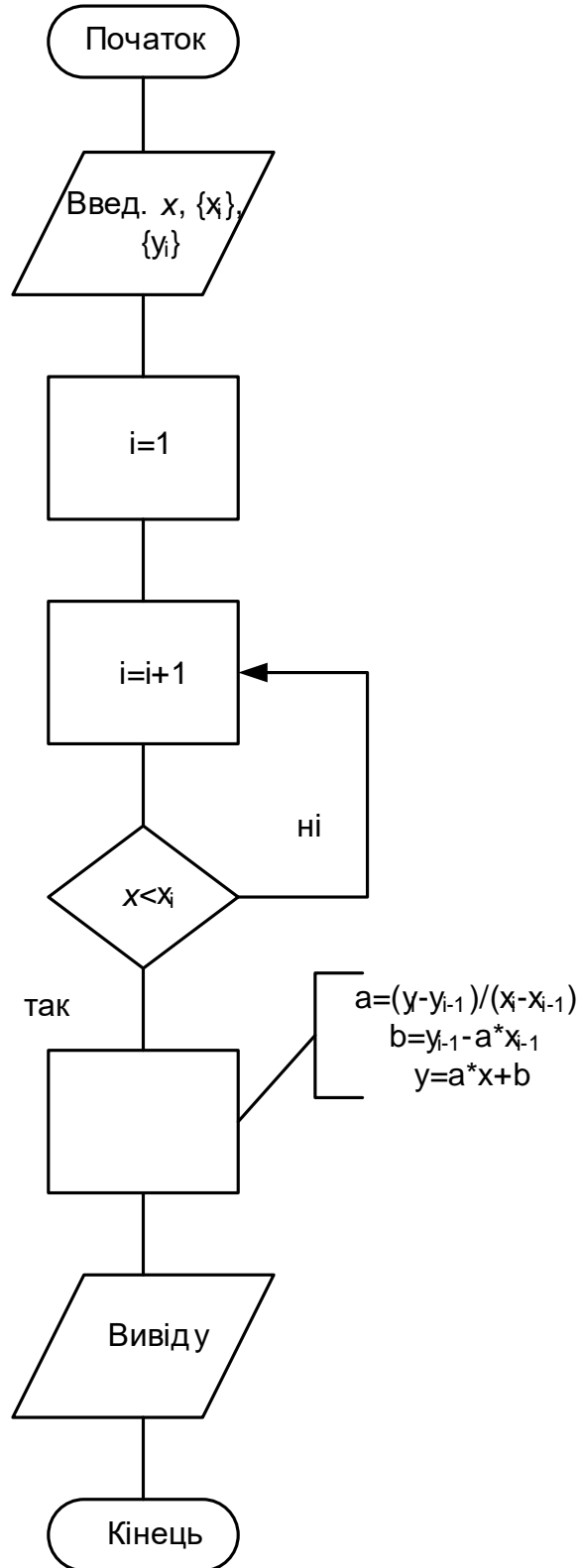
СПИСОК ЛИТЕРАТУРИ

1. Турчак Л. И. Основы численных методов / Л. И. Турчак – М.: Наука, 1987. – 320 с.
2. Самарский А. А. Численные методы / А. А. Самарский, Г. И. Гулин – М. : Наука, 1987. – 320 с.
3. Соболев И. М. Метод Монте-Карло / И. М. Соболев – М.: Наука, 1987. – 80 с.
4. Фиакко А. Нелинейное программирование. Методы последовательной безусловной минимизации / А. Фиакко, Г. Мак-Кормик – М. : Мир, 1972. – 192 с.
5. Дьяконов В. Maple 6 : учебный курс / В. Дьяконов – СПб. : Питер, 2001. – 608 с.

ДОДАТОК А

А.1. Лінійна інтерполяція

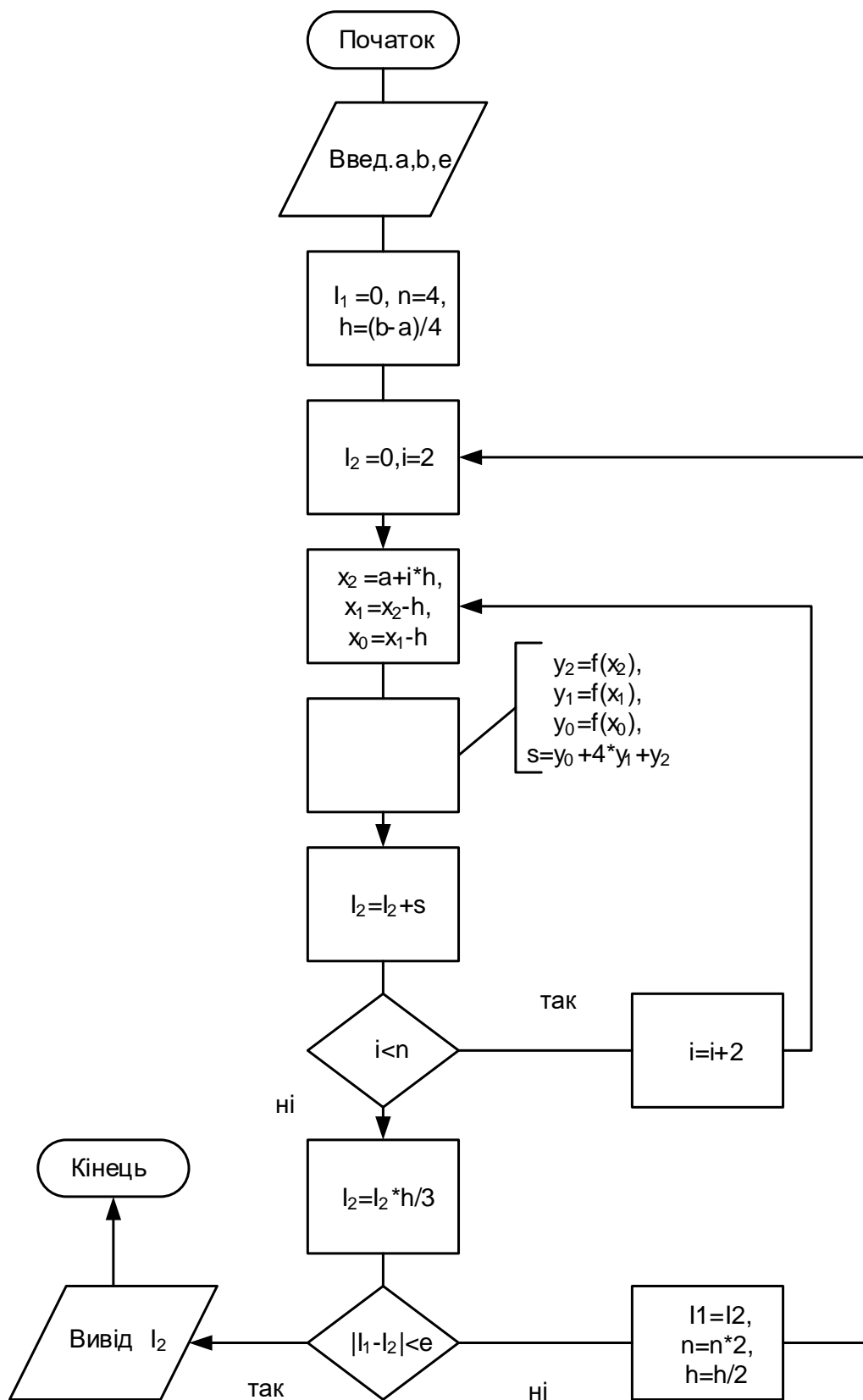
Блок-схема лінійної інтерполяції



А.2. Диференціали і інтеграли

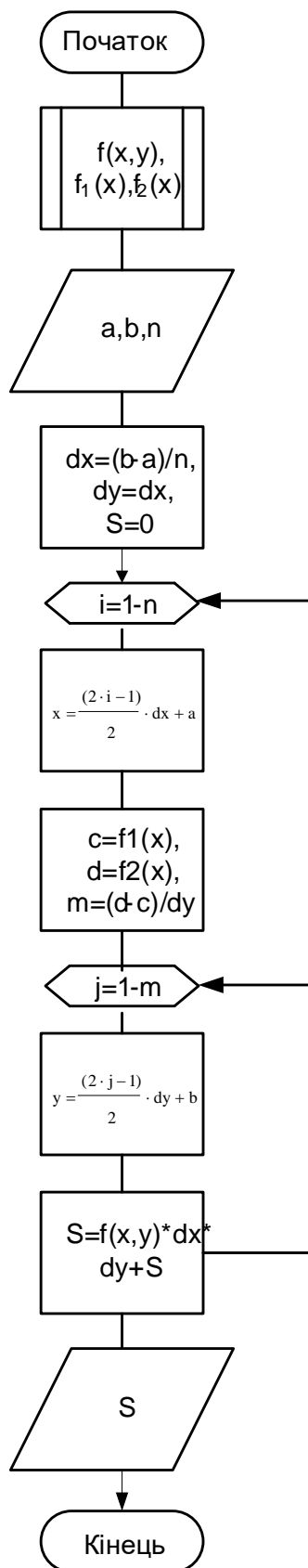
А.2.1. Метод Сімпсона

Блок-схема методу Сімпсона



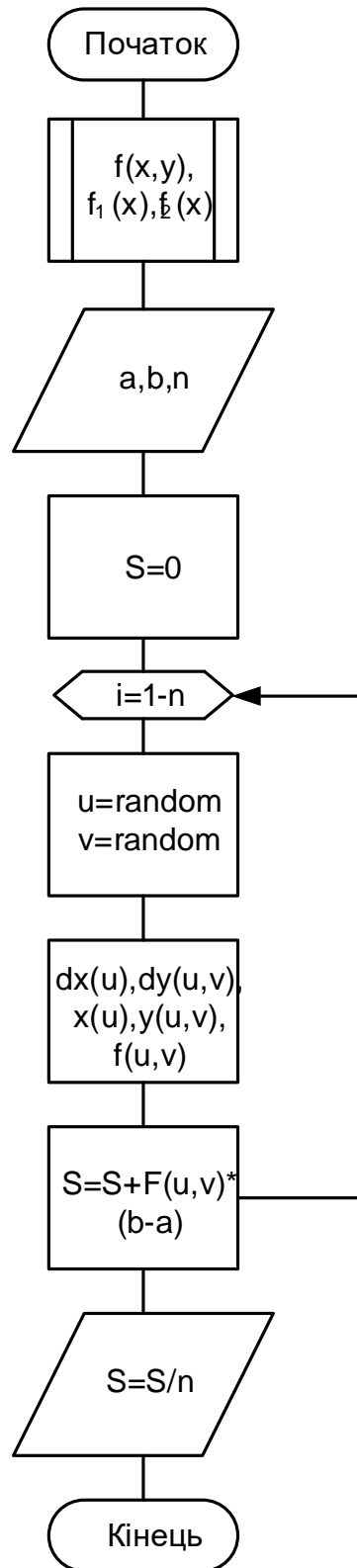
A.2.2. Обчислення подвійного інтегралу

Обчислення подвійного інтегралу зі змінними границями інтегрування



A.2.3. Метод Монте-Карло

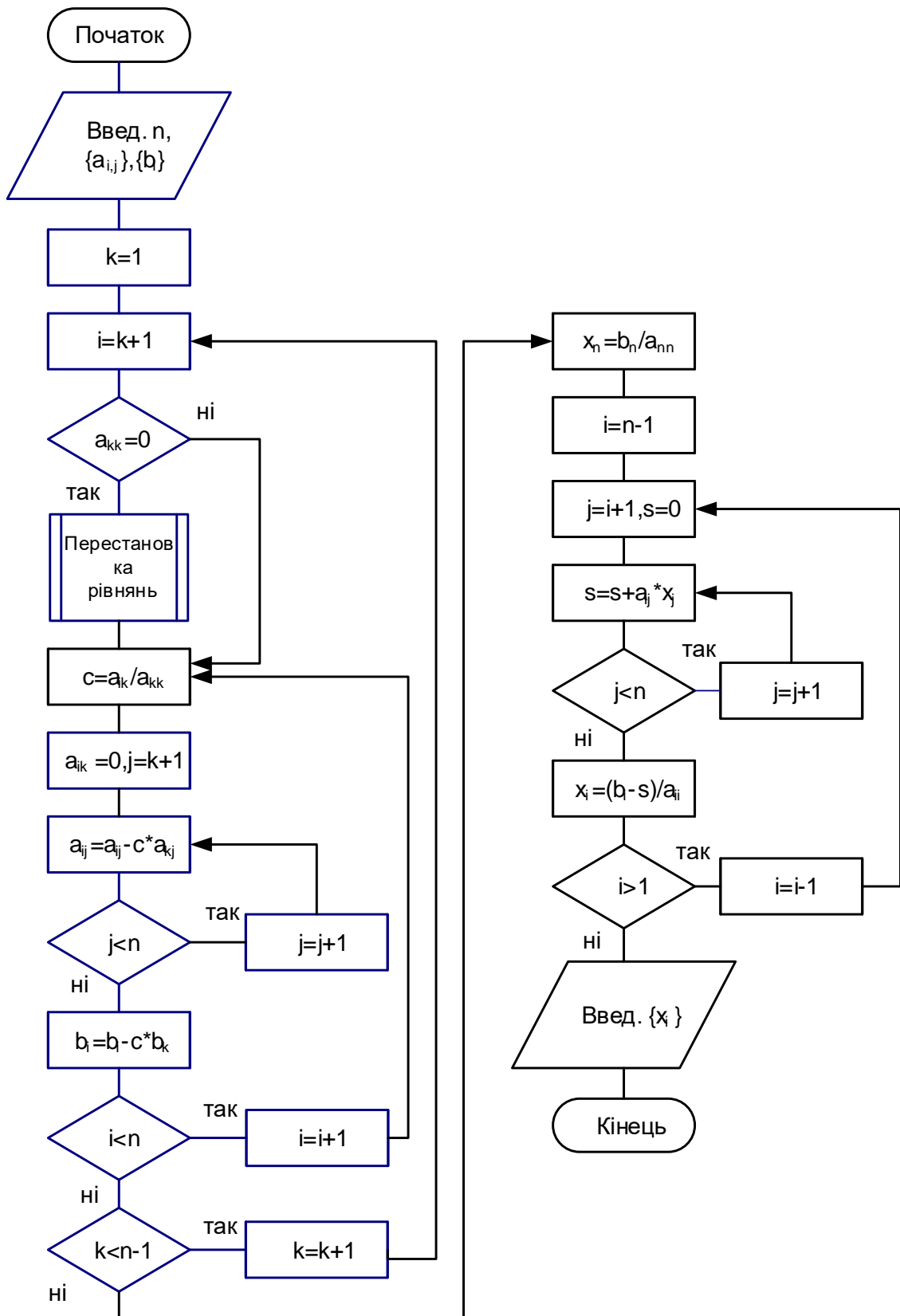
Блок-схема обчислення інтегралу методом Монте - Карло



А.3. Системи лінійних рівнянь.

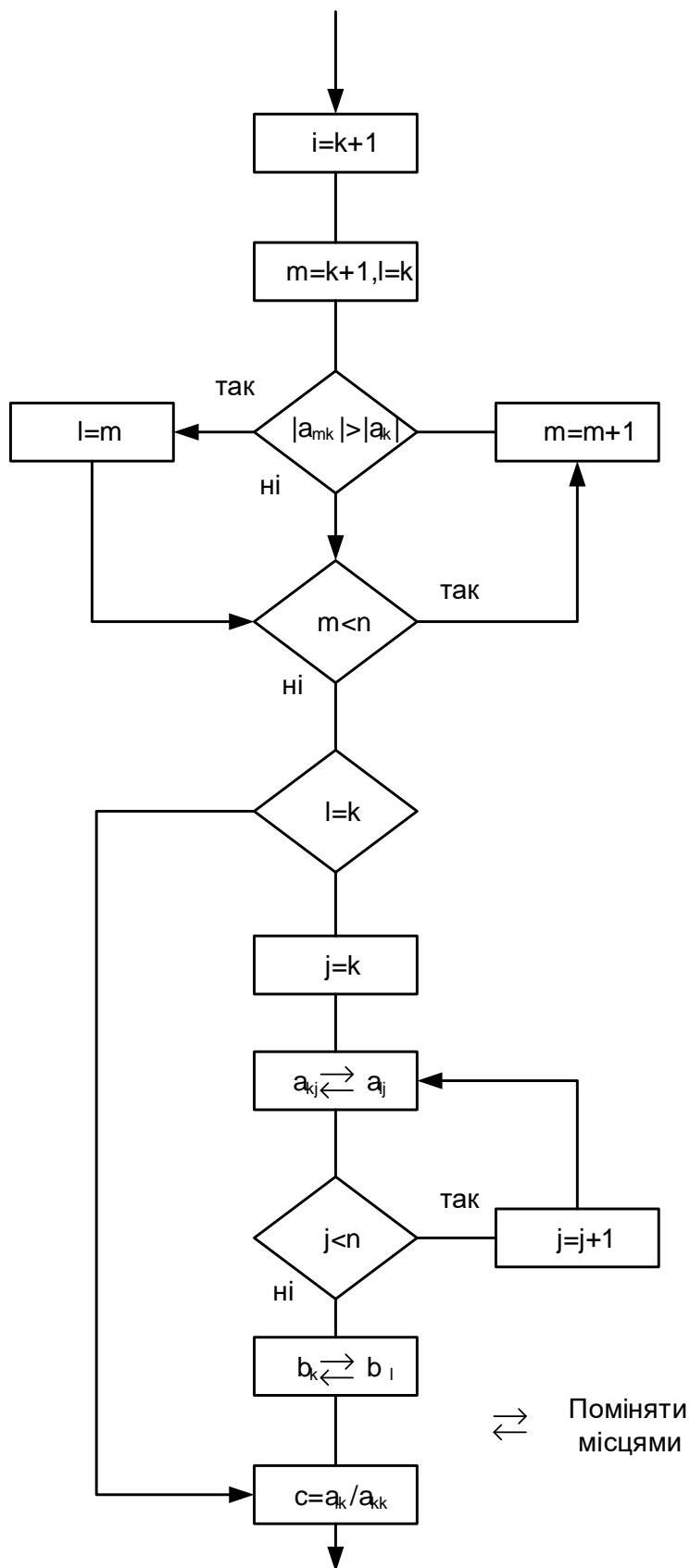
А.3.1. Метод Гауса

Блок-схема розв'язку системи рівнянь методом Гауса



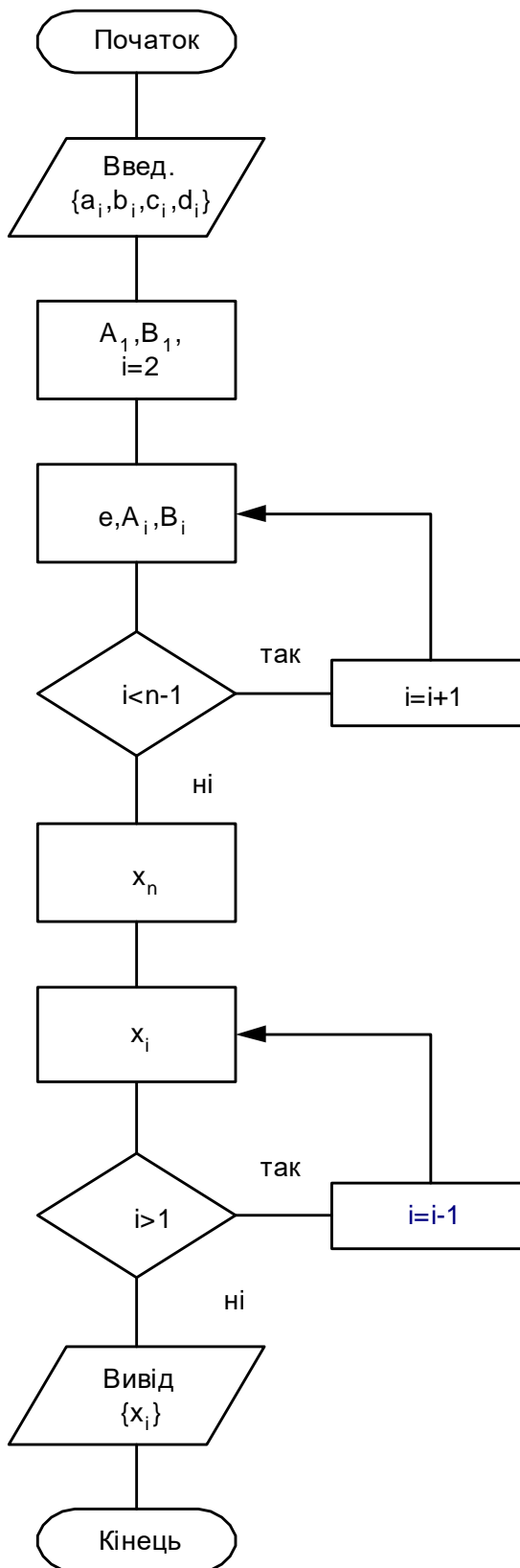
А.3.2. Вибір головного елемента

Блок-схема вибору головного елемента



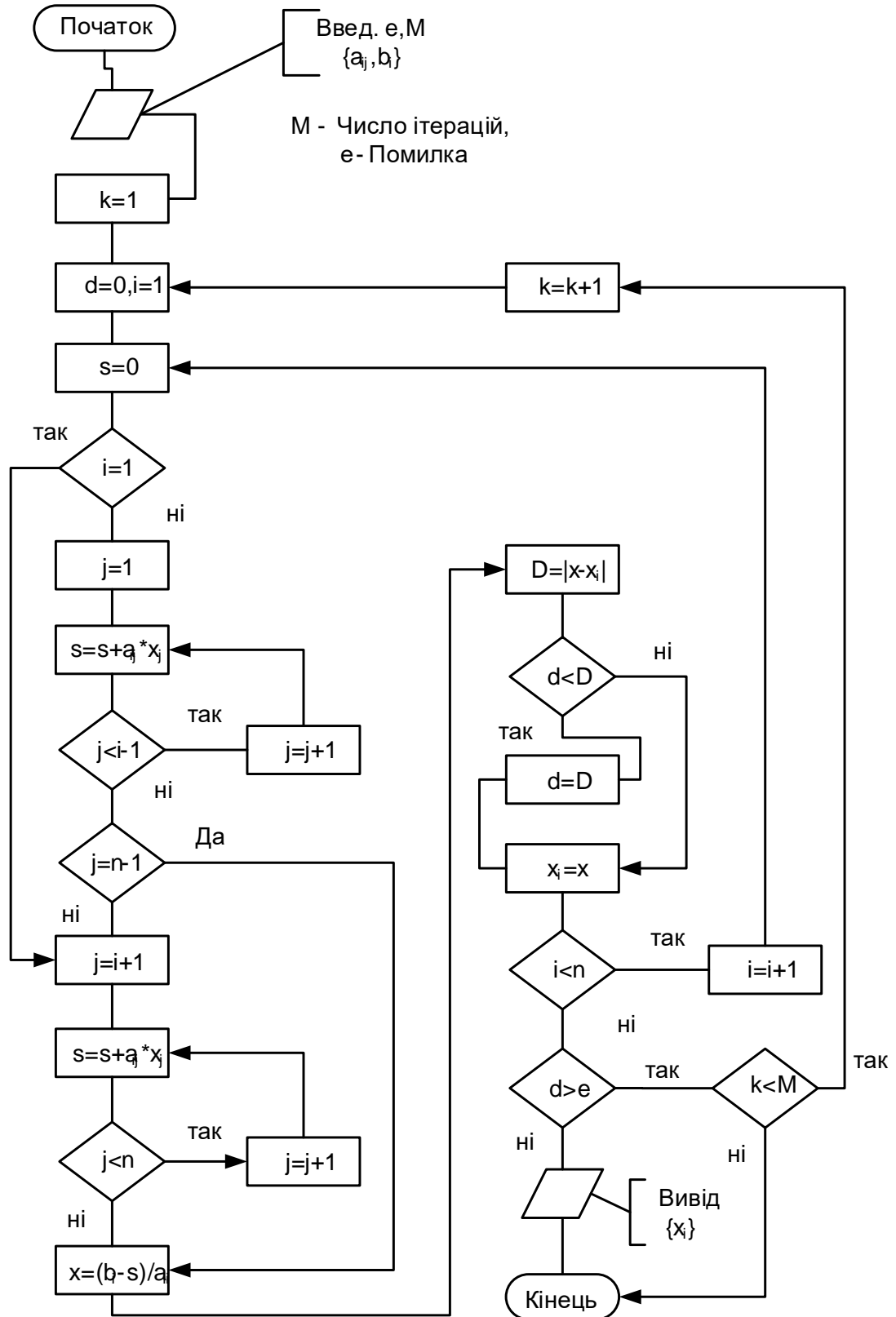
А.3.3. Метод прогонки

Блок-схема методу прогонки



А.3.4. Метод Гауса-Зейделя

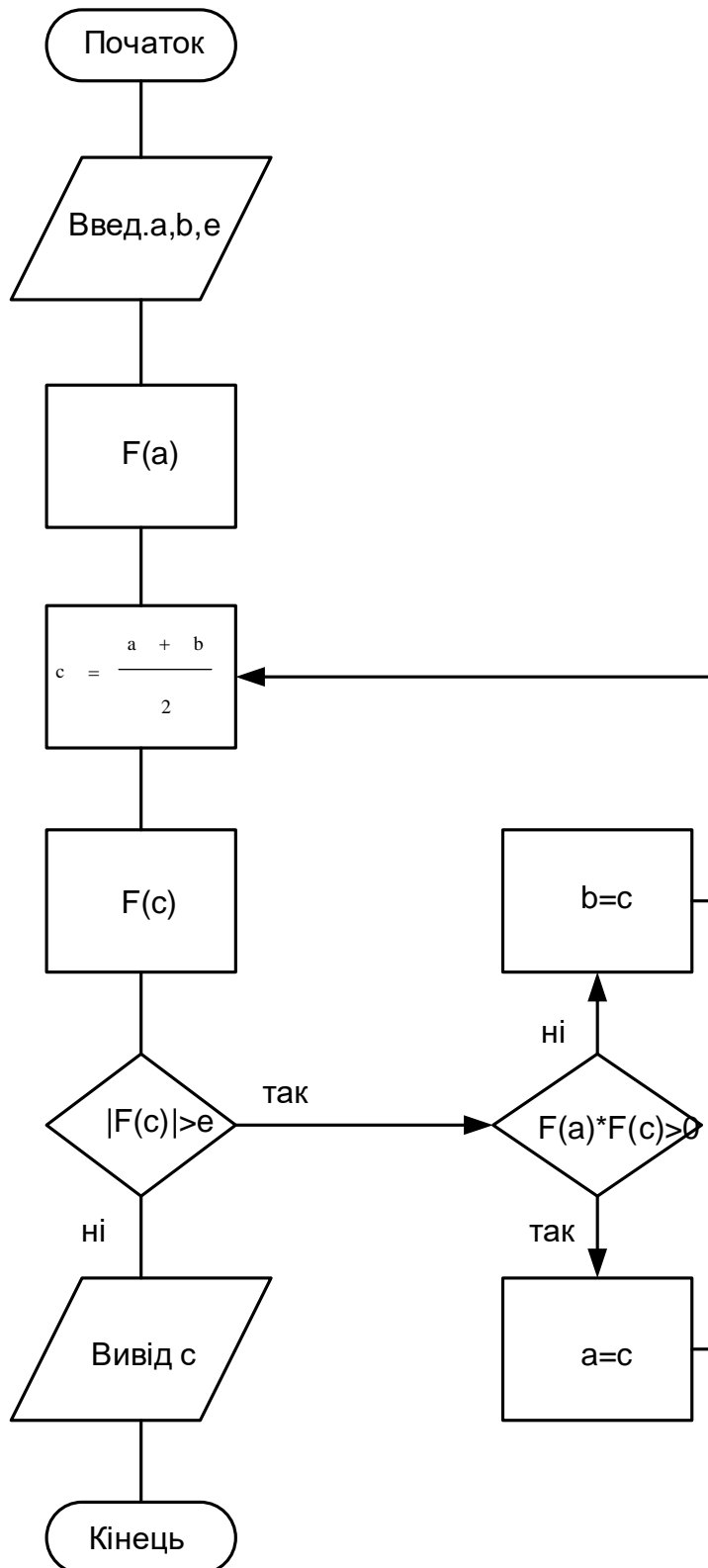
Блок-схема методу Гауса-Зейделя



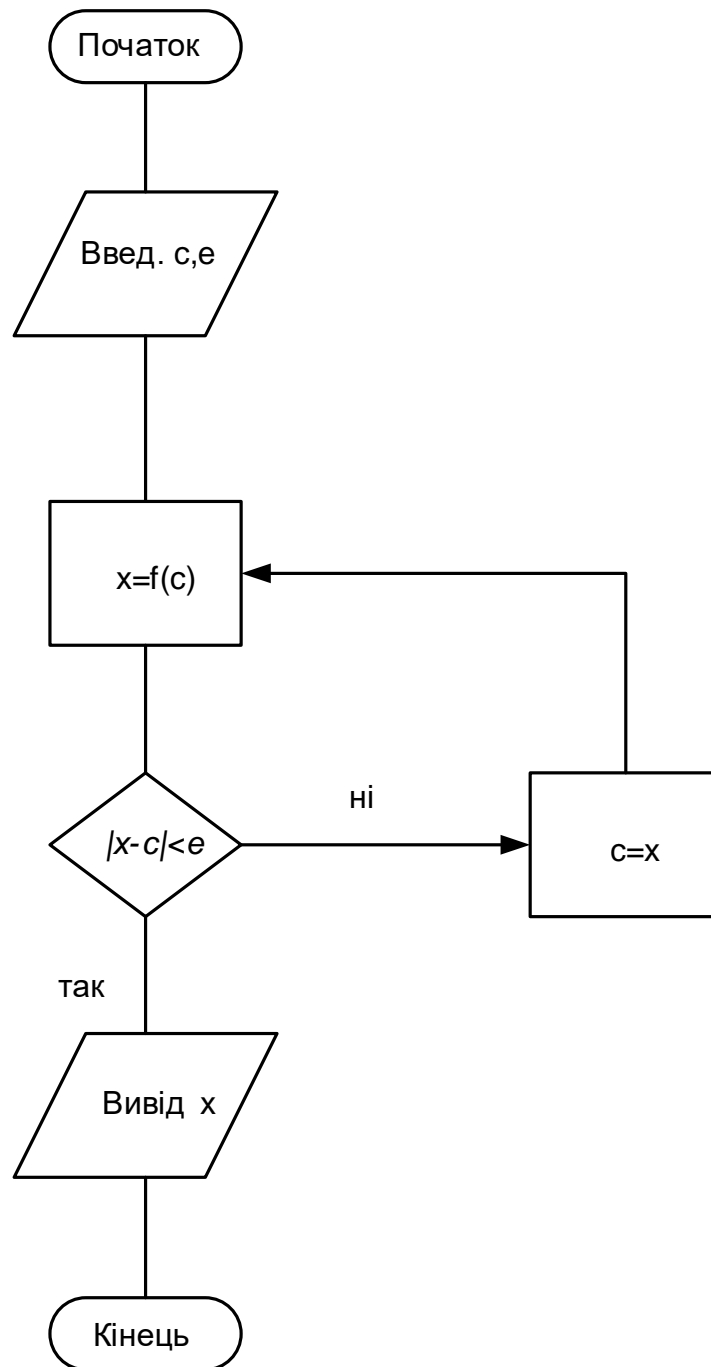
А.4. Нелінійні рівняння

А.4.1. Метод дихотомії

Блок-схема методу дихотомії

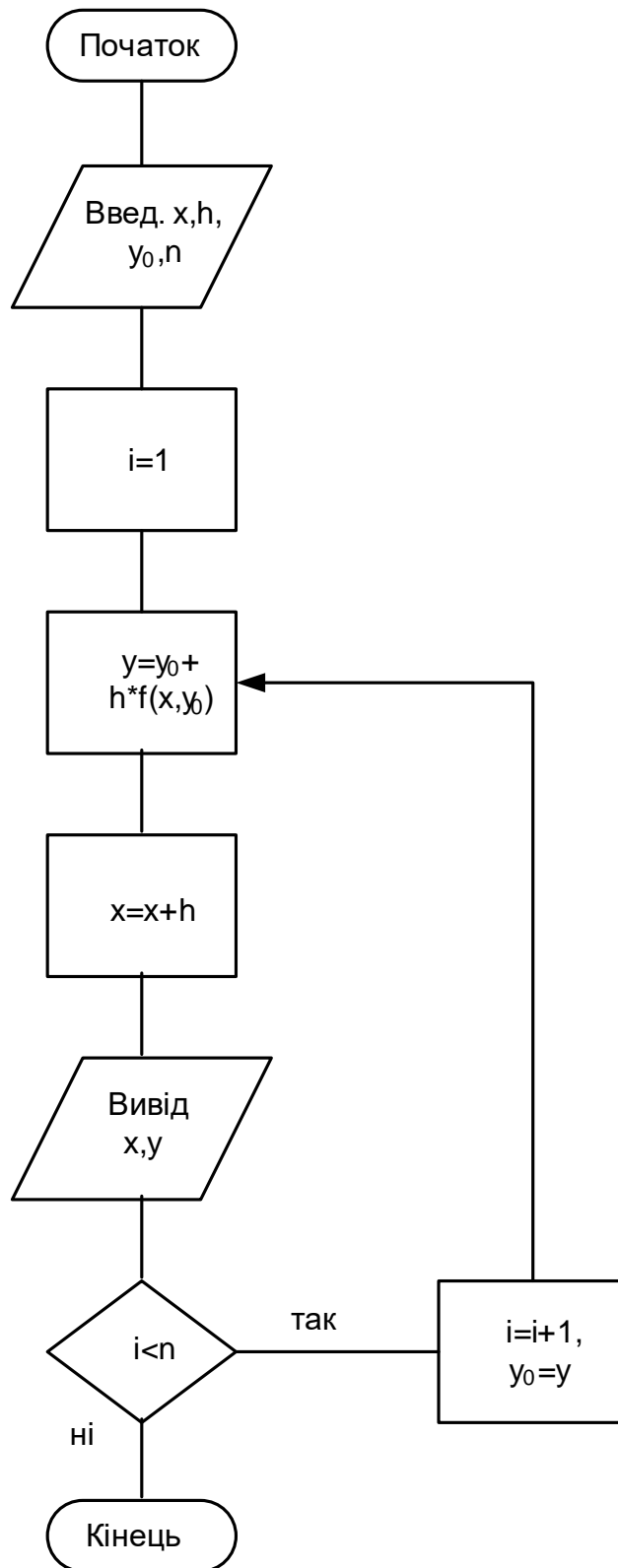


А.4.2. Метод простої ітерації
Блок-схема методу простої ітерації



А.5. Вирішення ОДР. Метод Ейлера

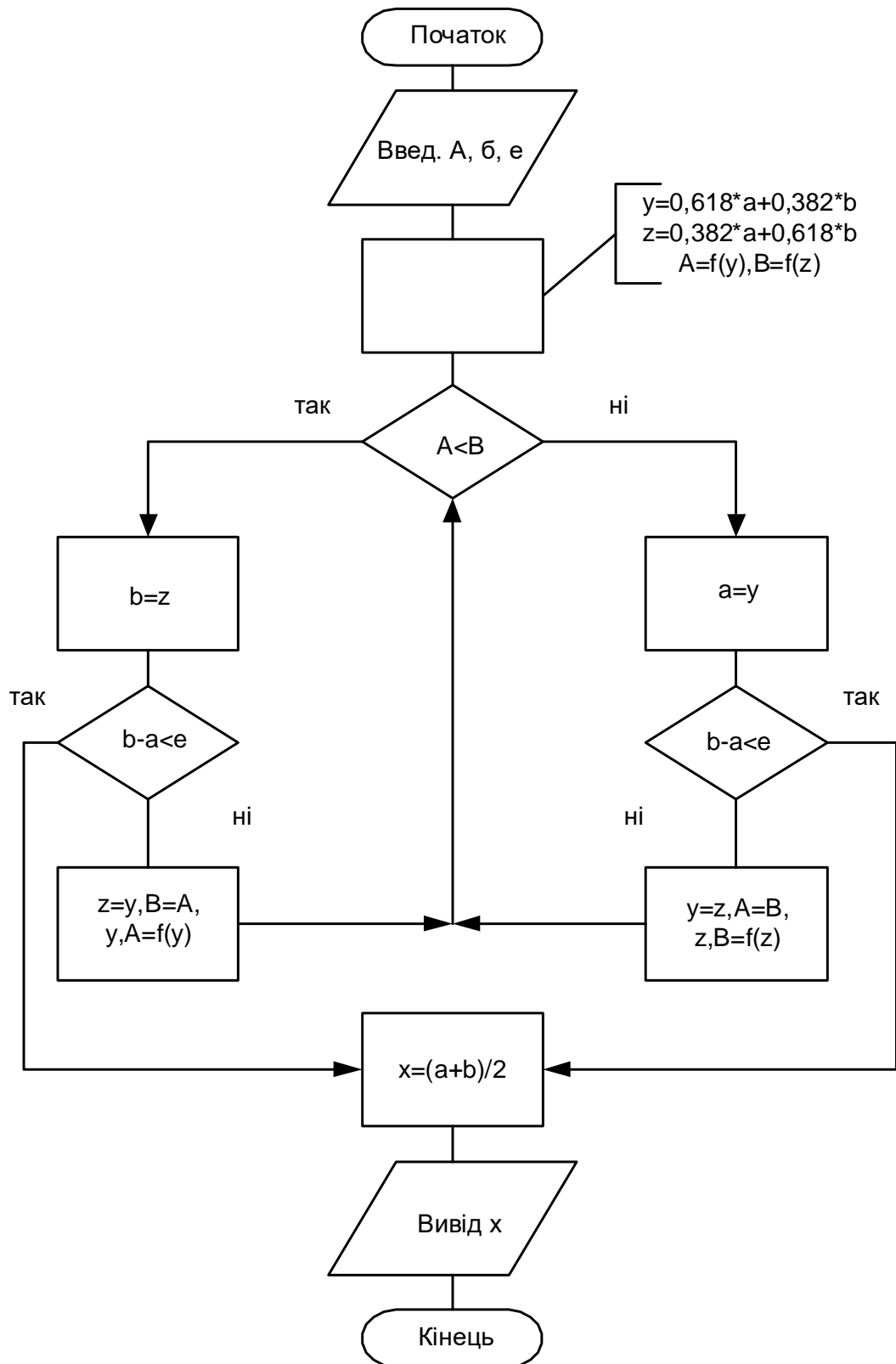
Блок-схема розв'язку ОДР методом Ейлера



А.6. Алгоритми оптимізаційних методів

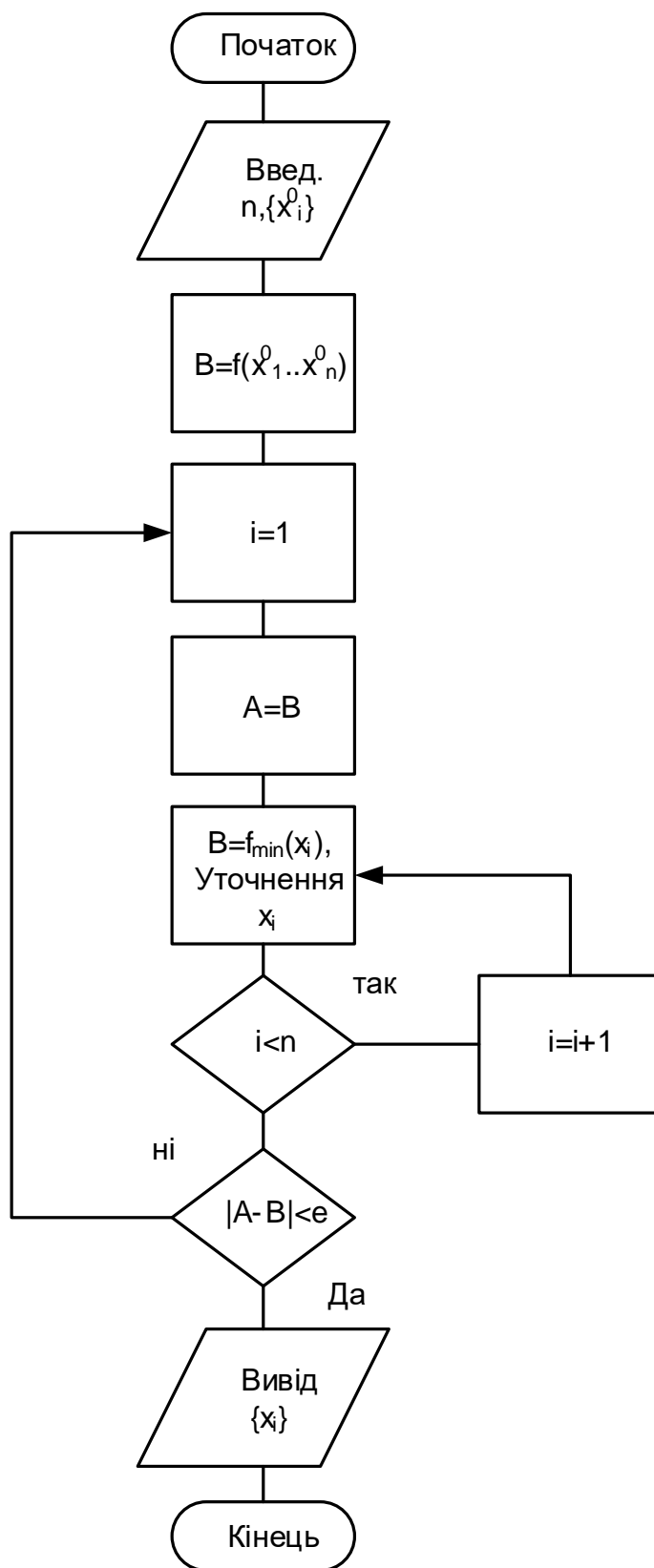
А.6.1. Одновимірна оптимізація методом золотого перетину

Блок-схема одновимірної оптимізації методом
Золотого перетину



А.6.2. Метод покоординатного спуску

Блок-схема покоординатного спуску



ДОДАТОК Б

Б.1. Апроксимація функцій

Б.1.1. Лінійна, квадратична і сплайн-інтерполяція

> restart:

> X:=[0.15,0.3,0.4,0.55];#Список табличних значень аргументу

$$X := [0.15, 0.3, 0.4, 0.55]$$

> Y:=[2.17,3.63,5.07,7.78];#Список табличних значень функції

$$Y := [2.17, 3.63, 5.07, 7.78]$$

> pare:=(X,Y)->[X,Y]; #Функція відповідності значень аргументу значенням функції

$$\text{pare} := (X, Y) \rightarrow [X, Y]$$

> y0:=zip(pare,X,Y); #Перетворення парного списку у функціональну залежність

$$y0 := [[0.15, 2.17], [0.3, 3.63], [0.4, 5.07], [0.55, 7.78]]$$

> g0:=plot(y0,x=0.15..0.56,style=point,thickness=100,color=black); #Побудова графічного об'єкту табличної функції

$$g0 := PLOT(...)$$

> y1:=spline(X,Y,x,linear); #Лінійна інтерполяція

$$y1 := \begin{cases} 0.7100000000 + 9.733333333x & x < 0.3 \\ -0.6900000000 + 14.40000000x & x < 0.4 \\ -2.156666667 + 18.06666667x & \text{otherwise} \end{cases}$$

> y2:=spline(X,Y,x,quadratic); #Квадратична інтерполяція

$$y2 := \begin{cases} 2.170000000 + 86.444444459999997x - 0.15)^2 & x < 0.22500000000000000 \\ -0.270000000 + 13. x + 0.22222220000000000(x - 0.3)^2 & x < 0.34999999999999997 \\ -2.352222224 + 18.555555599999984x + 55.333333439999997(x - 0.4)^2 & x < 0.47499999999999997 \\ 7.780000000 - 179.03703719999998(x - 0.55)^2 & \text{otherwise} \end{cases}$$

> y3:=spline(X,Y,x,cubic); #Сплайн-інтерполяція

$$y_3 := \begin{cases} 0.894375000 + 8.5041666659999997k & x < 0.3 \\ + 54.6296296299999996(x - 0.15)^3 & \\ -0.027500001 + 12.1916666700000000k & x < 0.4 \\ + 24.583333329166667(x - 0.3)^2 & \\ - 24.99999993000000014(x - 0.3)^3 & \\ -1.473333336 + 16.35833334000000015x & \text{otherwise} \\ + 17.083333354166669(x - 0.4)^2 & \\ - 37.962963000000002(x - 0.4)^3 & \end{cases}$$

> with(plots);#Підключення бібліотеки графіки

> display(g0, g1, g2, g3); # Побудова ряду графічних об'єктів

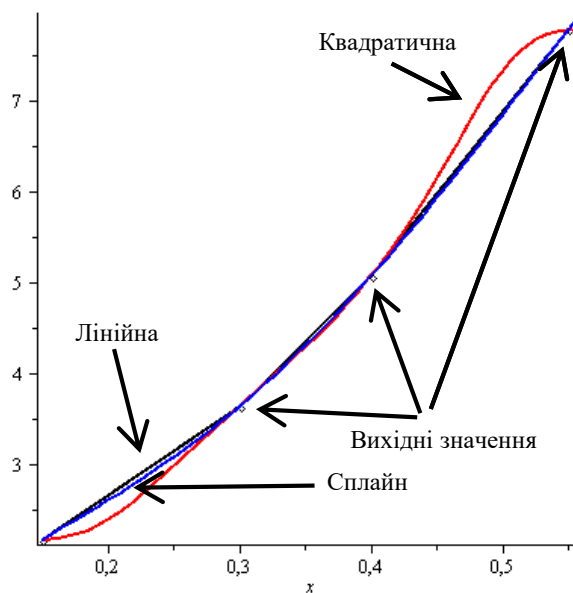


Рисунок Б.1 – Види інтерполяції

Б.1.2. Глобальна апроксимація

> restart:

> X:=[0.5,1,2,3,4,5];Y:=[0,5,-2,1,0,4];#Список табличних значень

X := [0.5, 1, 2, 3, 4, 5]

Y := [0, 5, -2, 1, 0, 4]

> y:=interp(X,Y, x); #Функція глобальної апроксимації поліномом степені m=n-1, де n – кількість парних елементів у списку – 6 значень.

$$y := 0.7759259245x^5 - 10.68055554x^4 + 54.03703697x^3 - 122.0416665x^2 + 117.0203703x - 34.11111108$$

> g2:=plot(y,x=0.4..5): #Графічний об'єкт

> pare:=(X,Y)->[X,Y]; #Функція відповідності значень аргументу значенням функції

$$pare := (X, Y) \rightarrow [X, Y]$$

> CoordXY:=zip(pare,X,Y,2);

$$CoordXY := [[0.5, 0], [1, 5], [2, -2], [3, 1], [4, 0], [5, 4]]$$

> g1:=plot(CoordXY,x=0.5..5,style=point,color=black):

> with(plots):

> display({g1,g2});

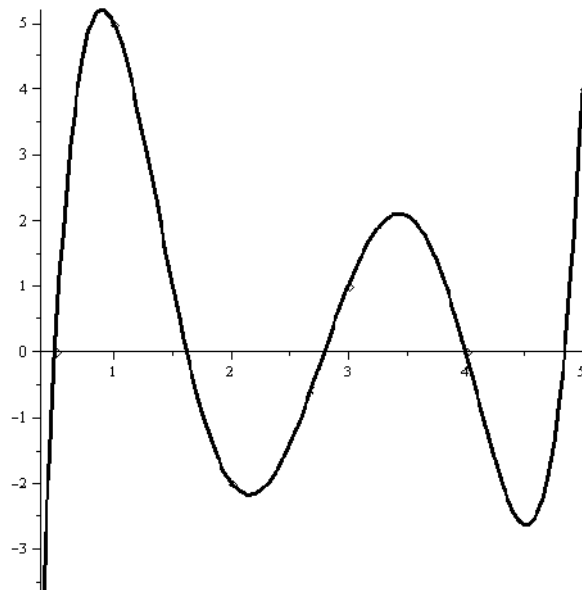


Рисунок Б.2 – Глобальна апроксимація

Б.1.3. Метод найменших квадратів

> restart:

> X:=array(1..4,[0.75,1.5,2.25,3.75]); #Масив значень аргументу

$$X := [0.75 \ 1.5 \ 2.25 \ 3.75]$$

> Y:=array(1..4,[2.5,1.2,1.12,4.48]); #Масив значень функції

$$Y := [2.5 \ 1.2 \ 1.12 \ 4.48]$$

> pare:=(X,Y)->[X,Y];

> y0:=zip(pare,X,Y);

$$pare := (X, Y) \rightarrow [X, Y]$$

$$y0 := [[0.75, 2.5] \ [1.5, 1.2] \ [2.25, 1.12] \ [3.75, 4.48]]$$

> y:=a0+a1*X[i]+a2*X[i]^2; #Вигляд апроксимованої функції, яка задається користувачем

$$y := a0 + a1 X_i + a2 X_i^2$$

```
> S:=sum((y-Y[i])^2,i=1..4);
```

$$S := (a_0 + 0.75 a_1 + 0.5625 a_2 - 2.5)^2 + (a_0 + 1.5 a_1 + 2.25 a_2 - 1.2)^2 + (a_0 + 2.25 a_1 + 5.0625 a_2 - 1.12)^2 + (a_0 + 3.75 a_1 + 14.0625 a_2 - 4.48)^2$$

```
> sys:=diff(S,a0)=0,diff(S,a1)=0,diff(S,a2)=0; #Обчислення похідних по невідомим значенням коефіцієнтів
```

$$\begin{aligned} \text{sys} := & 8 a_0 + 16.50 a_1 + 43.8750 a_2 - 18.60 = 0, 16.50 a_0 \\ & + 43.8750 a_1 + 135.843750 a_2 - 45.9900 = 0, 43.8750 a_0 \\ & + 135.843750 a_1 + 457.5234374 a_2 - 145.552500 = 0 \end{aligned}$$

```
> sols:=solve({sys},{a0,a1,a2});assign(sols); #Розв'язок системи рівнянь і привласнення числових значень коефіцієнтам a0...a2
```

$$\text{sols} := \{a_2 = 1.056161629, a_1 = -4.091030364, a_0 = 4.970363690\}$$

```
> f:=unapply(a0+a1*x+a2*x^2,x); #Задання функціональної залежності апроксимованого поліному
```

$$f := x \rightarrow 4.970363690 - 4.091030364x + 1.056161629x^2$$

```
> g1:=plot(f(x),x=0.74..3.76,color=red,style=line,thickness=2):
```

```
> g2:=plot(y0,x=0.74..3.76,style=point,color=black,thickness=2):
```

```
> with(plots);
```

```
> display(g1,g2);
```

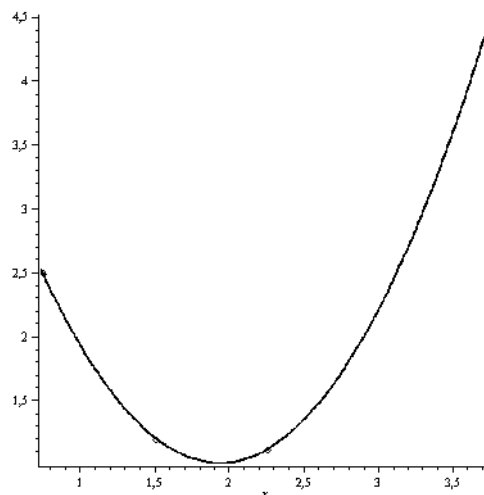


Рисунок Б.3 – Метод найменших квадратів

Б.2. Диференціювання і інтегрування

Б.2.1. Обчислення похідної

> restart:

> diff(a*x^n,x);# Вирахування похідної

$$\frac{a x^n n}{x}$$

> df:=diff(a*sin(x)+b*cos(x),x); # Вирахування похідної від суми

$$df := a \cos(x) - b \sin(x)$$

> diff(a*x^n,x,x); # Вирахування другої похідної

$$\frac{a x^n n^2}{x^2} - \frac{a x^n n}{x^2}$$

> f:=(x,y)->cos(x)*y^3; #Задання функцій двох змінних

$$f := (x, y) \rightarrow \cos(x) y^3$$

> z_x:=diff(f(x,y),x); #Вирахування частинної похідної по x

$$z_x := -\sin(x) y^3$$

> z_y:=diff(f(x,y),y); # Вирахування частинної похідної по y

$$z_y := 3 \cos(x) y^2$$

Б.2.2. Обчислення похідної від складної функції

>> restart :

> y := x→3 x + 5x²; #Задання y, як функції x

$$y := x \rightarrow 3 x + 5 x^2$$

> z := y→y(x)³ #Задання z, як функції y

$$z := y \rightarrow y(x)^3$$

> z(y);

$$(3 x + 5 x^2)^3$$

> diff(z(y),x)#Похідна від z, по x

$$3 (3 x + 5 x^2)^2 (3 + 10 x)$$

Б.2.3. Обчислення похідної від кусково-заданої функції

> restart:

> X:=[0.15,0.3,0.4,0.55]; #Задання списку значень X

```

X := [0.15, 0.3, 0.4, 0.55]
> Y := [2.17, 3.63, 5.07, 7.78]; #Задання списку значень Y
Y := [2.17, 3.63, 5.07, 7.78]
>
> y := spline(X, Y, x); # Апроксимація списку сплайном

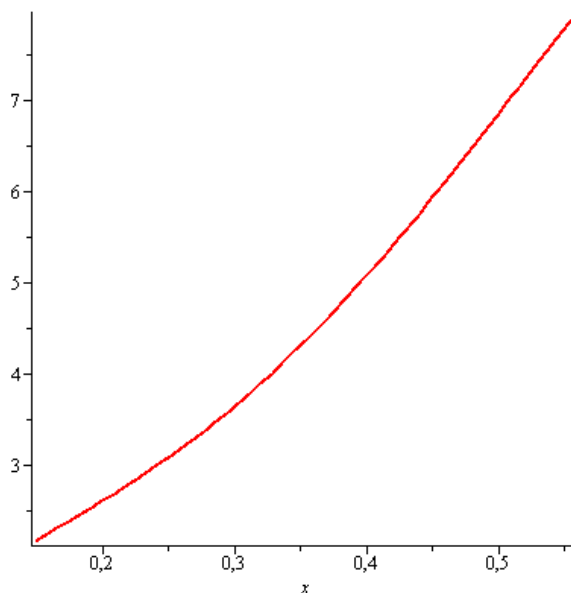
```

$$y := \begin{cases} 0.894375000 & x < 0.3 \\ + 8.50416666599999971 x & \\ + 54.6296296299999966 (x & \\ - 0.15)^3 & \\ -0.027500001 & x < 0.4 \\ + 12.19166667000000001 x & \\ + 24.5833333291666670 (x & \\ - 0.3)^2 & \\ - 24.99999993000000014 (x & \\ - 0.3)^3 & \\ -1.473333336 & \text{otherwise} \\ + 16.35833334000000015 x & \\ + 17.0833333541666690 (x & \\ - 0.4)^2 & \\ - 37.96296300000000020 (x & \\ - 0.4)^3 & \end{cases}$$

```

> plot(y, x=0.15..0.56, thickness=2); #Графік функції

```



```

> dy := diff(y, x) #Вирахування похідної

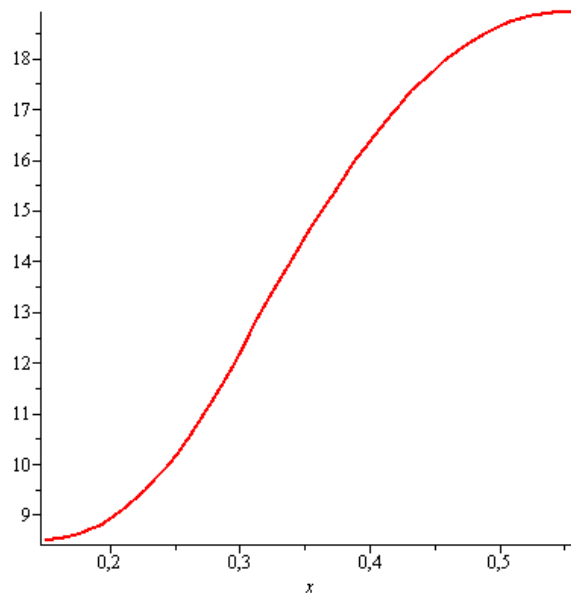
```

```

dy := {
  8.504166667      x < 0.3000000000
  + 163.8888889 (x
  - 0.1500000000)2
  Float(undefined) x = 0.3000000000
  -2.558333333    x < 0.4000000000
  + 49.16666667 x
  - 74.99999979 (x
  - 0.3000000000)2
  Float(undefined) x = 0.4000000000
  2.691666650    0.4000000000 < x
  + 34.16666671 x
  - 113.8888889 (x
  - 0.4000000000)2

```

> **plot(dy, x = 0.15..0.56, thickness = 2);** #Графік похідної



Б.2.4. Обчислення інтегралу від кусково-заданої функції

> restart:

> X:=[0.15,0.3,0.4,0.55];

X:= [0.15, 0.3, 0.4, 0.55]

> Y:=[2.17,3.63,5.07,7.78];

Y:= [2.17, 3.63, 5.07, 7.78]

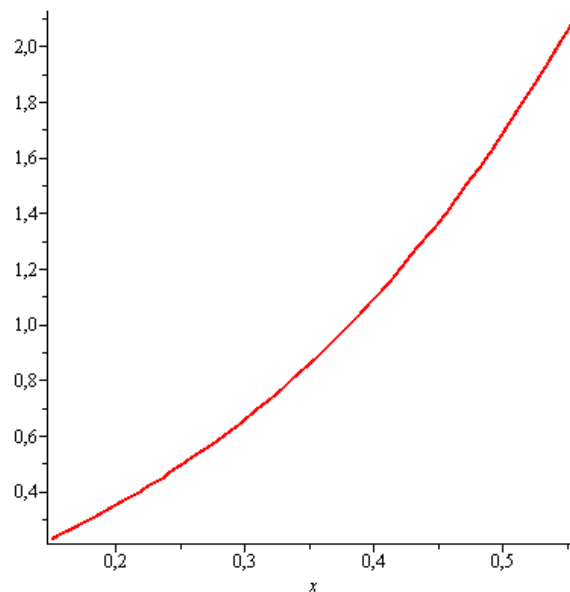
>

> y:=spline(X,Y,x):

> inty:=int(y,x) #Вирахування невизначеного інтегралу

$$\text{inty} := \begin{cases} 0.8943750000 x & x \leq 0.3000000000 \\ + 4.252083333 x^2 \\ + 13.65740741 (x \\ - 0.1500000000)^4 \\ -0.02750000100 x & x \leq 0.4000000000 \\ + 6.095833335 x^2 \\ + 8.194444443 (x \\ - 0.3000000000)^3 \\ - 6.249999982 (x \\ - 0.3000000000)^4 \\ + 0.1175390626 \\ -1.473333336 x & 0.4000000000 < x \\ + 8.179166670 x^2 \\ + 5.694444451 (x \\ - 0.4000000000)^3 \\ - 9.490740750 (x \\ - 0.4000000000)^4 \\ + 0.3701085075 \end{cases}$$

> **plot(inty, x = 0.15 ..0.56, thickness = 2);**



Б.2.5. Приклади обчислення інтегралів

> restart;

> J:=int(a^x,x);

$$J := \frac{a^x}{\ln(a)}$$

> J:=int(a^x,x=a1..b1); #Вирахування інтегралу на заданому інтервалі

$$J := -\frac{a^{a1} - a^{b1}}{\ln(a)}$$

> Int(1/x,x)=int(1/x,x);

$$\int \frac{1}{x} dx = \ln(x)$$

> J:=int(exp(-x^2),x);

$$J := \frac{1}{2} \sqrt{\pi} \operatorname{erf}(x)$$

> J:=int(sin(x)/x,x=0..1); #Вирахування спеціального інтегралу

$$J := \operatorname{Si}(1)$$

> evalf(%);# Вирахування чисельного значення спеціального інтегралу

$$0.9460830704$$

Б.2.6. Обчислення кратних інтегралів

Обчислення об'єму верхньої напівсфери, що розташована на початку координат

> restart:

> z:=sqrt(R^2-y^2-x^2); #Значення висоти верхньої напівсфери

$$z := \sqrt{R^2 - y^2 - x^2}$$

> f1:=-sqrt(R^2-x^2);f2:=sqrt(R^2-x^2); #Границі вимірювання функції по y

$$f1 := -\sqrt{R^2 - x^2}$$

$$f2 := \sqrt{R^2 - x^2}$$

> a:=-R;b:=R; # Границі вимірювання функції по x

$$a := -R$$

$$b := R$$

> J:=int(int(z,y=f1..f2),x=a..b); #Значення інтегралу

$$J := \frac{2}{3} \pi R^3$$

Обчислення площі фігури, яка обмежена кривими $y1 = x^2$ і $y1 = x^{0.5}$

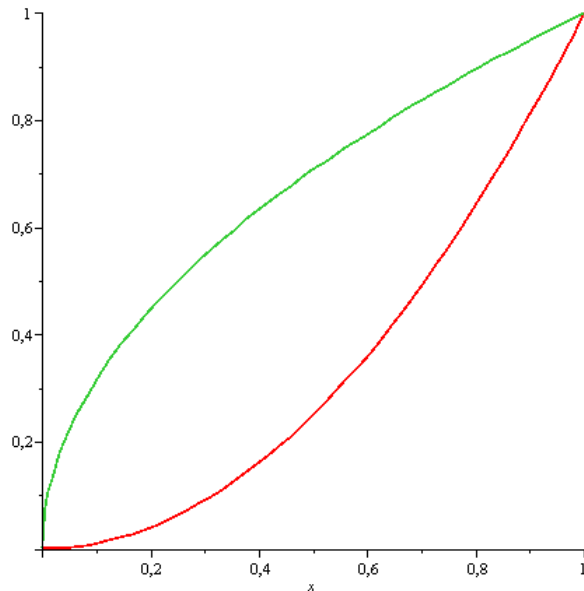
> restart;

> f1:=x^2;f2:=sqrt(x); # Межі вимірювання функції по y

$$f1 := x^2$$

$$f2 := \sqrt{x}$$

> plot([f1,f2],x=0..1,thickness=2); #Вид функцій



> J1:=int(int(1,y=f1..f2),x=0..1); #Вирахування інтегралу на інтервалі 0...1

$$J1 := \frac{1}{3}$$

> evalf(%);#Перетворення отриманого результату

$$0.3333333333$$

Загальний випадок вирахування потрійного інтегралу

> restart :

> x1 := a; x2 := b;

$$x1 := a$$

$$x2 := b$$

> y1 := c; y2 := d;

$$y1 := c$$

$$y2 := d$$

> z1 := 0; z2 := x² + y²;

$$z1 := 0$$

$$z2 := x^2 + y^2$$

> J := int(int(int(1, z = z1 ..z2), y = c ..d), x = a ..b)

$$J := \frac{1}{3} (d - c) (b^3 - a^3) + \frac{1}{3} d^3 (b - a)$$

$$- \frac{1}{3} c^3 (b - a)$$

Вирахування інтегралу чисельним методом. Метод трапецій

> restart :

```
> z := x3 + y3; # Обчислювальний інтеграл
```

```
z := x3 + y3
```

```
> J1 := int(int(z, x = 0 ..1.0), y = 0  
..1);
```

```
J1 := 0.5000000000#Точне значення інтегралу
```

```
> restart :
```

```
> h := 0.1; #Крок інтегрування
```

```
h := 0.1
```

```
#Задання функції, масивів дискретних значень, значення початкового  
інтервалу інтегрування
```

```
>
```

```
z := (x, y) → x3 + y3;
```

```
x := array(1 ..11);
```

```
y := array(1 ..11);
```

```
x[1] := 0; y[1] := 0;
```

```
z := (x, y) → x3 + y3
```

```
x := array(1 ..11, [ ])
```

```
y := array(1 ..11, [ ])
```

```
x1 := 0
```

```
y1 := 0
```

```
#Заповнення масивів аргументів
```

```
for i from 2 to 11 do x[i] := x[i  
- 1] + h :
```

```
od:
```

```
for j from 2 to 11 do y[j] := y[j  
- 1] + h :
```

```
od:
```

```
> S := 0;
```

```
S := 0
```

```

for  $i$  from 1 to 10 do
  for  $j$  from 1 to 10 do
     $S := S + z(x[i], y[j])$ 
       $+ z(x[i], y[j + 1]) + z(x[i$ 
       $+ 1], y[j]) + z(x[i + 1],$ 
       $y[j + 1]);$ 
  od;od;

```

```

>  $J := \frac{S \cdot h^2}{4.0};$ 

```

$J := 0.5050000000$ #Отримане значення інтегралу

Вирахування кратного інтегралу методом Монте-Карло. Об'єм верхньої напівсфери

```

> restart :

```

```

>  $R := 1.0;$  #Радіус сфери

```

$R := 1.0$

```

>  $z := \text{sqrt}(R^2 - x^2 - y^2);$  #Опис верхньої напівсфери

```

$z := \sqrt{1.00 - x^2 - y^2}$

```

>  $x := -R + 2 \cdot R \cdot u;$  #Перетворення змінної  $x$ 

```

$x := -1.0 + 2.0 u$

```

# Перетворення змінної  $y$ 

```

```

>  $y := -\text{sqrt}(R^2 - x^2) + 2 \cdot \text{sqrt}(R^2$ 
   $- x^2) \cdot v;$ 

```

$y := -\sqrt{1.00 - (-1.0 + 2.0 u)^2} + 2 \sqrt{1.00 - (-1.0 + 2.0 u)^2} v$

```

>  $z;$ 

```

$$\left(1.00 - (-1.0 + 2.0 u)^2 - \left(-\sqrt{1.00 - (-1.0 + 2.0 u)^2} + 2 \sqrt{1.00 - (-1.0 + 2.0 u)^2} v \right)^{1/2} \right)$$

> $dx := \text{diff}(x, u)$ #Диференціал

$$dx := 2.0$$

> $dy := \text{diff}(y, v)$

$$dy := 2 \sqrt{1.00 - (-1.0 + 2.0 u)^2}$$

> $J := z \cdot dx \cdot dy$; #Розрахункова формула

$$J :=$$

$$4.0 \left(1.00 - (-1.0 + 2.0 u)^2 - \left(-\sqrt{1.00 - (-1.0 + 2.0 u)^2} + 2 \sqrt{1.00 - (-1.0 + 2.0 u)^2} v \right)^{1/2} \right)$$

> $S := 0.0$;

$$S := 0.$$

> $N := 1000$; #Число випадкових точок

$$N := 1000$$

#Генерування випадкових значень u, v в межах $0 \dots 1$

>

for i **from** 1 **to** N **do**

$u := rand() * 1.0 \cdot 10^{-12} ;$

$v := rand() * 1.0 \cdot 10^{-12} ;$

$S := S + J ;$

od:

> $J := \frac{S}{N} ;$

$J := 2.083614649$ #Вирахуване значення

> $evalf\left(\frac{2 \cdot \pi \cdot R^3}{3}\right) ;$

2.094395103 #Точне значення

Б.4. Системи лінійних рівнянь

Б.4.1. Системи лінійних рівнянь з дійсними коефіцієнтами

Розв'язати систему рівнянь

$$7.1 \cdot x_1 + 6.8 \cdot x_2 + 6.1 \cdot x_3 = 7.0$$

$$5.0 \cdot x_1 + 4.8 \cdot x_2 + 5.3 \cdot x_3 = 6.1$$

$$8.2 \cdot x_1 + 7.8 \cdot x_2 + 7.1 \cdot x_3 = 5.8$$

> restart:

> A:=matrix(3,3,[7.1,6.8,6.1,5.0,4.8,5.3,8.2,7.8,7.1]);#Порядкове заповнення матриці коефіцієнтів рівнянь

$$A := \begin{bmatrix} 7.1 & 6.8 & 6.1 \\ 5.0 & 4.8 & 5.3 \\ 8.2 & 7.8 & 7.1 \end{bmatrix}$$

> B:=array(1..3,[7.0,6.1,5.8]);#Матриця стовпчик правих частин

$$B := \begin{bmatrix} 7.0 & 6.1 & 5.8 \end{bmatrix}$$

> with(linalg):#Підключення бібліотеки лінійної

> X:=linsolve(A,B);#Оператор вирішення системи рівнянь

$$X := \begin{bmatrix} -41.14507716, 43.38082845, \\ 0.678756458 \end{bmatrix}$$

> X[1];X[2];X[3];#Вивід отриманих коренів

$$-41.14507716$$

$$43.38082845$$

$$0.678756458$$

Б.4.2. Системи лінійних рівнянь з комплексними коефіцієнтами

> restart:

> z1:=1+I*2;z2:=0.5+I*0.1;z3:=2-I*1;z4:=I*3;z5:=-I*2;z6:=4+I*1;z7:=0.2;z8:=-I*1;#Значення комплексних опорів електричної схеми

$$z1 := 1 + 2 I$$

$$z2 := 0.5 + 0.1 I$$

$$z3 := 2 - I$$

```

z4 := 3 I
z5 := -2 I
z6 := 4 + I
z7 := 0.2
z8 := -I
> Z:=matrix(4,4,[z1+z2,-z2,0,0,-z2,z2+z4+z5,-z4,0,0,-z4,z4+z3+z6,-z6,0,0,-
z6,z6+z7+z8]); #Матриця коефіцієнтів

```

$$Z := \begin{bmatrix} 1.5 + 2.1I & -0.5 - 0.1I & 0 & 0 \\ -0.5 - 0.1I & 0.5 + 1.1I & -3I & 0 \\ 0 & -3I & 6 + 3I & -4 - I \\ 0 & 0 & -4 - I & 4.2 + 0.1I \end{bmatrix}$$

```

> E1:=10.0;E2:=1+I*0.5;E3:=2+I*1; #Значення електрорушійних сил

```

$$E1 := 10.0$$

$$E2 := 1. + 0.5 I$$

$$E3 := 2 + I$$

```

> E:=array(1..4,[E1,-E2,E2,-E3]); #Матриця-стовпчик прикладених
електрорушійних сил

```

$$E := \begin{bmatrix} 10.0 & -1. - 0.5 I & 1. + 0.5 I & -2 - I \end{bmatrix}$$

```

>

```

```

> with(linalg): #Підключення бібліотеки лінійної алгебри

```

```

> Tok:=linsolve(Z,E); #Розрахункові значення контурних струмів кола

```

$$Tok := [2.193708811 - 3.321450125I, 0.3664060616 - 0.8240545793I, \\ 0.6441245196 - 0.2202212173I, 0.1896950704 - 0.2944667499I]$$

Б.4.3. Розв'язок системи трьох лінійних рівнянь в загальному вигляді

```

> restart:

```

```

> A:=matrix(3,3,[a11,a12,a13,a21,a22,a23,a31,a32,a33]);

```

$$A := \begin{bmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{bmatrix}$$

```

> B:=array(1..3,[b1,b2,b3]);

```

$$B := \begin{bmatrix} b1 & b2 & b3 \end{bmatrix}$$

> with(linalg):

> X:=linsolve(A,B);

$$X := \left[\begin{aligned} & (a_{12} a_{23} b_3 - a_{12} b_2 a_{33} + a_{13} a_{32} b_2 - a_{13} a_{22} b_3 \\ & + b_1 a_{22} a_{33} - b_1 a_{32} a_{23}) / (a_{11} a_{22} a_{33} - a_{11} a_{32} a_{23} \\ & - a_{21} a_{12} a_{33} + a_{32} a_{21} a_{13} - a_{22} a_{31} a_{13} \\ & + a_{31} a_{12} a_{23}), -(a_{11} a_{23} b_3 - a_{11} b_2 a_{33} \\ & - a_{21} a_{13} b_3 - a_{23} a_{31} b_1 + b_2 a_{31} a_{13} + a_{21} b_1 a_{33}) / \\ & (a_{11} a_{22} a_{33} - a_{11} a_{32} a_{23} - a_{21} a_{12} a_{33} + a_{32} a_{21} a_{13} \\ & - a_{22} a_{31} a_{13} + a_{31} a_{12} a_{23}), (a_{32} a_{21} b_1 \\ & - a_{11} a_{32} b_2 + a_{11} a_{22} b_3 - a_{22} a_{31} b_1 - a_{21} a_{12} b_3 \\ & + a_{31} a_{12} b_2) / (a_{11} a_{22} a_{33} - a_{11} a_{32} a_{23} - a_{21} a_{12} a_{33} \\ & + a_{32} a_{21} a_{13} - a_{22} a_{31} a_{13} + a_{31} a_{12} a_{23}) \end{aligned} \right]$$

> X[1];

$$\begin{aligned} & (a_{12} a_{23} b_3 - a_{12} b_2 a_{33} + a_{13} a_{32} b_2 - a_{13} a_{22} b_3 \\ & + b_1 a_{22} a_{33} - b_1 a_{32} a_{23}) / (a_{11} a_{22} a_{33} - a_{11} a_{32} a_{23} \\ & - a_{21} a_{12} a_{33} + a_{32} a_{21} a_{13} - a_{22} a_{31} a_{13} \\ & + a_{31} a_{12} a_{23}) \end{aligned}$$

> X[2];

$$\begin{aligned} & -(a_{11} a_{23} b_3 - a_{11} b_2 a_{33} - a_{21} a_{13} b_3 - a_{23} a_{31} b_1 \\ & + b_2 a_{31} a_{13} + a_{21} b_1 a_{33}) / (a_{11} a_{22} a_{33} - a_{11} a_{32} a_{23} \\ & - a_{21} a_{12} a_{33} + a_{32} a_{21} a_{13} - a_{22} a_{31} a_{13} \\ & + a_{31} a_{12} a_{23}) \end{aligned}$$

> X[3];

$$\begin{aligned} & (a_{32} a_{21} b_1 - a_{11} a_{32} b_2 + a_{11} a_{22} b_3 - a_{22} a_{31} b_1 \\ & - a_{21} a_{12} b_3 + a_{31} a_{12} b_2) / (a_{11} a_{22} a_{33} - a_{11} a_{32} a_{23} \\ & - a_{21} a_{12} a_{33} + a_{32} a_{21} a_{13} - a_{22} a_{31} a_{13} \\ & + a_{31} a_{12} a_{23}) \end{aligned}$$

Б.5. Нелінійні рівняння

Б.5.1. Метод дихотомії

```
> restart :
#Задання функції
> f := x → exp(x) - 5.0;
                                     f := x → ex - 5.0
#Задание інтервалів пошуку кореня і точність
> a := 0.0; b := 5.0; eps := 1.0
   · 10-2;
                                     a := 0.
                                     b := 5.0
                                     eps := 0.010000000000
> Fa := f(a);
                                     Fa := -4.0
>
for i from 1 to 20 do
  c := (a + b) · 0.5;
  Fc := f(c) :
if abs(Fc) < eps then break fi:
  if Fa · Fc > 0 then a := c :
    else b := c :
fi:
od;
                                     c := 2.50
                                     Fc := 7.18249396
                                     c := 1.250
                                     Fc := -1.509657043
                                     c := 1.8750
                                     Fc := 1.520819120
                                     c := 1.56250
                                     Fc := -0.229266818
                                     c := 1.718750
                                     Fc := 0.577552165
```

```

c := 1.6406250
Fc := 0.158392500
c := 1.60156250
Fc := -0.039222413
c := 1.621093750
Fc := 0.058620158
c := 1.611328125
Fc := 0.009460001

```

> c;

1.611328125

Б.5.2. Метод Ньютона

> restart :

>

> f := x → exp(x) - 5.0;

f := x → e^x - 5.0

> a := 0.00; eps := 1.0 · 10⁻²;

a := 0.

eps := 0.010000000000

>

for i from 1 to 20 do

$c := a - \frac{f(a)}{D(f)(a)}$;

if abs(c - a) > eps **then** a := c;

else break

fi:

od;

c := 4.000000000

c := 3.091578194

c := 2.318729195

c := 1.810721953

c := 1.628402097

c := 1.609616601

$$c := 1.609437928$$

Точне значення кореня 1.609437912

Б.5.3. Розв'язок нелінійних рівнянь системою MAPLE

> restart;

> eq1:=a*x^2+b*x+c=0;# Рівняння для розв'язку

$$eq1 := a x^2 + b x + c = 0$$

> sols:=solve(eq1,x);# Застосування функції solve

$$sols := \frac{-b + \sqrt{b^2 - 4 a c}}{2 a}, \frac{-b - \sqrt{b^2 - 4 a c}}{2 a}$$

> x1:=sols[1];

$$x1 := \frac{-b + \sqrt{b^2 - 4 a c}}{2 a}$$

> x2:=sols[2];

$$x2 := \frac{-b - \sqrt{b^2 - 4 a c}}{2 a}$$

> eq2:=x^3-2*x+1=0;

$$eq2 := x^3 - 2 x + 1 = 0$$

> solve(eq2,x); # Комплексні корені

$$1, -\frac{1}{2} + \frac{\sqrt{5}}{2}, -\frac{1}{2} - \frac{\sqrt{5}}{2}$$

> eq3:=2*x^2+x+3=0;

$$eq3 := 2 x^2 + x + 3 = 0$$

> solve(eq3,x);

$$-\frac{1}{4} + \frac{1}{4} I \sqrt{23}, -\frac{1}{4} - \frac{1}{4} I \sqrt{23}$$

> solve(x^4=x-1,x);

$$\text{RootOf}(_Z^4 - _Z + 1, \text{index} = 1), \text{RootOf}(_Z^4 - _Z + 1, \text{index} = 2), \text{RootOf}(_Z^4 - _Z + 1, \text{index} = 3), \text{RootOf}(_Z^4 - _Z + 1, \text{index} = 4)$$

> evalf(%);

```
0.7271360845 + 0.4300142883 I, -0.7271360845
+ 0.9340992895 I, -0.7271360845
- 0.9340992895 I, 0.7271360845
- 0.4300142883 I
```

```
> restart :
```

```
> f := x → exp(x) - 5.0;
```

```
f := x → ex - 5.0
```

```
> x := solve(f(x) = 0, x)
```

```
x := 1.609437912
```

```
> restart :
```

```
> f := x → exp(-x2) - 5.0;
```

```
f := x → e-x2 - 5.0
```

```
> sols := solve(f(x) = 0, x) # Функція визначає всі корені
```

```
sols := -1.268636241 I,
1.268636241 I
```

```
> restart :
```

```
> f := x → exp(-x2) - 5.0;
```

```
f := x → e-x2 - 5.0
```

```
> sols := fsolve(f(x) = 0, x,
complex)
```

```
sols := 1.547778530 10-19
+ 1.268636241 I
```

```
# Знайдений один з коренів
```

```
> restart :
```

```
> f := x → exp(-x2) - 5.0;
```

```
f := x → e-x2 - 5.0
```

```
> sols := fsolve(f(x) = 0, x, -2·I..0, complex) # Вказаний інтервал пошуку
кореня
```

```
sols := -1.268636241 I
```

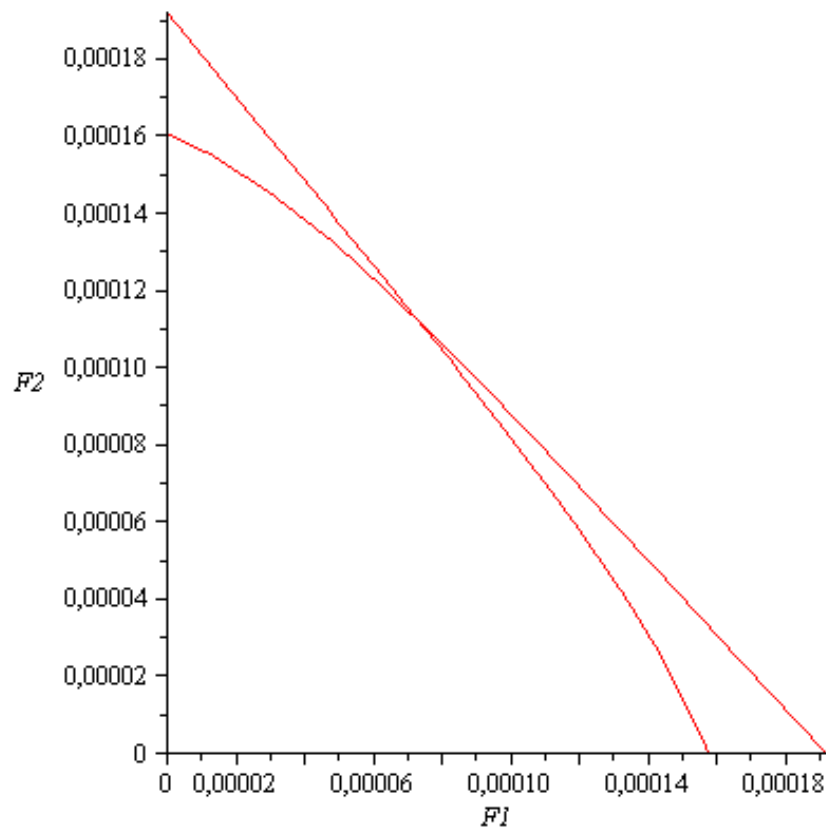
Вирішення системи рівнянь

> *restart* :

> #Графічне вирішення системи

with(plots) :

$$\begin{aligned} > \text{implicitplot} \left(\left[\begin{aligned} &4750.000000 - 2.375000000 \cdot 10^7 F1 \\ &- 2.375000000 \cdot 10^7 F2 = 2.985668790 \cdot 10^6 F1 \\ &+ 9.952229300 \cdot 10^5 F2 \\ &+ \frac{7.961783439 \cdot 10^8 F1}{6000.0 e^{-1.300000000 \cdot 10^8 F1^2} + 1}, \\ &4750.000000 - 2.375000000 \cdot 10^7 F1 \\ &- 2.375000000 \cdot 10^7 F2 = 9.952229300 \cdot 10^5 F1 \\ &+ 1.791401274 \cdot 10^6 F2 \\ &+ \frac{7.961783439 \cdot 10^8 F2}{6000.0 e^{-1.300000000 \cdot 10^8 F2^2} + 1} \end{aligned} \right], F1 = 0 .. 3 \\ &\cdot 10^{-4}, F2 = 0 .. 3 \cdot 10^{-4} \right) \end{aligned}$$



> #Система рівнянь

$$\begin{aligned}
 \text{sys} := & 4750.000000 - 2.375000000 \cdot 10^7 F1 \\
 & - 2.375000000 \cdot 10^7 F2 = 2.985668790 \cdot 10^6 F1 \\
 & + 9.952229300 \cdot 10^5 F2 \\
 & + \frac{7.961783439 \cdot 10^8 F1}{6000.0 e^{-1.300000000 \cdot 10^8 F1^2} + 1}, \\
 & 4750.000000 - 2.375000000 \cdot 10^7 F1 \\
 & - 2.375000000 \cdot 10^7 F2 = 9.952229300 \cdot 10^5 F1 \\
 & + 1.791401274 \cdot 10^6 F2 \\
 & + \frac{7.961783439 \cdot 10^8 F2}{6000.0 e^{-1.300000000 \cdot 10^8 F2^2} + 1}
 \end{aligned}$$

$$\begin{aligned}
sys := & 4750.000000 - 2.375000000 \cdot 10^7 F1 \\
& - 2.375000000 \cdot 10^7 F2 = 2.985668790 \cdot 10^6 F1 \\
& + 9.952229300 \cdot 10^5 F2 \\
& + \frac{7.961783439 \cdot 10^8 F1}{6000.0 e^{-1.300000000 \cdot 10^8 F1^2} + 1}, \\
& 4750.000000 - 2.375000000 \cdot 10^7 F1 \\
& - 2.375000000 \cdot 10^7 F2 = 9.952229300 \cdot 10^5 F1 \\
& + 1.791401274 \cdot 10^6 F2 \\
& + \frac{7.961783439 \cdot 10^8 F2}{6000.0 e^{-1.300000000 \cdot 10^8 F2^2} + 1}
\end{aligned}$$

> *sols* := *fsolve* ({*sys*}, {*F1*, *F2*}); *assign*(*sols*); # Розв'язок системи і
привласнення значенням *F1* і *F2* знайдених чисельних значень

$$\begin{aligned}
sols := & \{F1 = 0.00007321874320, F2 \\
& = 0.0001120599775\}
\end{aligned}$$

> *F1*; *F2*;

$$\begin{aligned}
& 0.00007321874320 \\
& 0.0001120599775
\end{aligned}$$

Б.6. Диференційні рівняння

Б.6.1. Розв'язок диференційних рівнянь в загальному вигляді

Включення r - L кола на постійну напругу

> restart:

> s1:=L*diff(Tok1(t),t)+r*Tok1(t)=U; #Вигляд рівняння

$$s1 := L \left(\frac{d}{dt} Tok1(t) \right) + r Tok1(t) = U$$

> nu1:=Tok1(0)=0; #Початкові умови

$$v1 := Tok1(0) = 0$$

> F:=dsolve({s1,nu1},{Tok1(t)}); #Функція вирішення

$$F := Tok1(t) = \frac{U}{r} - \frac{e^{-\frac{rt}{L}} U}{r}$$

> Tok1(t):=rhs(F); #Привласнення змінній Tok1(t) значення знайденого вирішення

$$Tok1(t) := \frac{U}{r} - \frac{e^{-\frac{rt}{L}} U}{r}$$

Включення r - L кола на синусоїдальну напругу

> s2:=L*diff(Tok2(t),t)+r*Tok2(t)=Um*sin(om*t+f); #Вид рівняння

$$s2 := L \left(\frac{d}{dt} Tok2(t) \right) + r Tok2(t) = Um \sin(om t + f)$$

> nu2:=Tok2(0)=0; #Початкові умови

$$v2 := Tok2(0) = 0$$

> F:=dsolve({s2,nu2},{Tok2(t)}); #Функція вирішення

$$\begin{aligned} F := Tok2(t) &= \frac{e^{-\frac{rt}{L}} (Um om L \cos(f) - Um r \sin(f))}{r^2 + om^2 L^2} \\ &- \frac{1}{r^2 + om^2 L^2} (Um (om L \cos(om t + f) \\ &- r \sin(om t + f))) \end{aligned}$$

> Tok2(t):=rhs(F); #Привласнення змінній знайденого вирішення

$$Tok2(t) := \frac{-\frac{r t}{L} (Um om L \cos(f) - Um r \sin(f))}{r^2 + om^2 L^2} - \frac{1}{r^2 + om^2 L^2} (Um (om L \cos(om t + f) - r \sin(om t + f)))$$

#Включення r-L-C кола на змінну напругу

> s3:=L*diff(Tok3(t),t,t)+r*diff(Tok3(t),t)+Tok3(t)/C=diff(Um*sin(om*t+f),t);

#Рівняння для розв'язку

$$s3 := L \left(\frac{d^2}{dt^2} Tok3(t) \right) + r \left(\frac{d}{dt} Tok3(t) \right) + \frac{Tok3(t)}{C} = Um \cos(om t + f) om$$

> nu3:=Tok3(0)=0,D(Tok3)(0)=Um*sin(f)/L; #Початкові умови

$$v3 := Tok3(0) = 0, D(Tok3)(0) = \frac{Um \sin(f)}{L}$$

> F:=dsolve({s3,nu3},{Tok3(t)}): #Функція вирішення

> Tok3(t):=rhs(F): #Вирішення не наводиться через громіздкість

#Вирішення системи рівнянь. Включення повітряного трансформатора на постійну напругу

>

> restart;

> sys := L1·diff(T1(t), t) + r1·T1(t) + M
·diff(T2(t), t) = Um,
L2·diff(T2(t), t) + r2·T2(t) + M
·diff(T1(t), t) = 0;

$$sys := L1 \left(\frac{d}{dt} T1(t) \right) + r1 T1(t) + M \left(\frac{d}{dt} T2(t) \right) = Um, L2 \left(\frac{d}{dt} T2(t) \right) + r2 T2(t) + M \left(\frac{d}{dt} T1(t) \right) = 0$$

> nu := T1(0) = 0, T2(0) = 0;

$$v := T1(0) = 0, T2(0) = 0$$

> $F := dsolve(\{sys, nu\}, \{T1(t), T2(t)\});$ #Розв'язок в загальному вигляді не приводиться через громіздкість

Б.6.2. Вирішення диференційних рівнянь в чисельному вигляді

#Включення r - L кола на випрямлену синусоїдальну напругу

> restart:

> $s := L * diff(Tok(t), t) + r * Tok(t) = abs(Um * sin(om * t + f));$

$$s := L \left(\frac{d}{dt} Tok(t) \right) + r Tok(t) = |Um \sin(om t + f)|$$

> $L := 1e-3; r := 1.0; Um := 50.0; om := 314.2; f := 0.1;$ #Завдання параметрів

$$L := 0.001$$

$$r := 1.0$$

$$Um := 50.0$$

$$om := 314.2$$

$$f := 0.1$$

> $nu := Tok(0) = 0;$

$$v := Tok(0) = 0$$

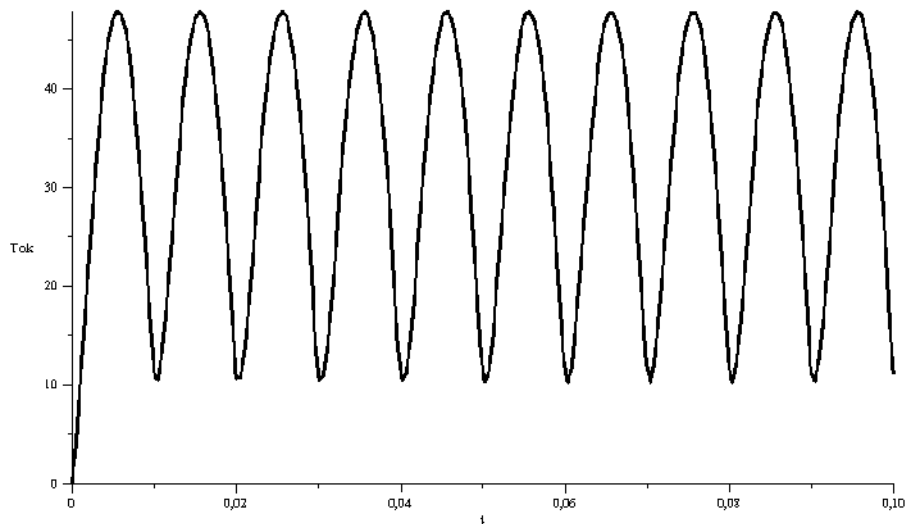
> $sols := dsolve(\{s, nu\}, \{Tok(t)\}, numeric);$ #Вирішення чисельним методом

$sols := proc(x_rkf45) ... end proc$

> with(plots): #Підключення спеціальної бібліотеки

> $odeplot(sols, [t, Tok(t)], 0..100e-3, numpoints=200, thickness=3, color=black);$

#Побудова графіка вирішення на інтервалі $0 \dots 0.1$ секунди, число точок – 200, товщина лінії – 3, колір – чорний.



У випадку, якщо отриманим рішенням необхідно користуватись для подальших обчислень, то необхідно застосувати оператор, який показаний в наступному прикладі.

> restart:

> $L := 0.1; r := 10.0; U := 10.0;$

$L := 0.1$

$r := 10.0$

$U := 10.0$

> $s1 := L * \text{diff}(\text{Tok1}(t), t) + r * \text{Tok1}(t) = U;$

$$s1 := 0.1 \left(\frac{d}{dt} \text{Tok1}(t) \right) + 10.0 \text{Tok1}(t) = 10.0$$

> $\text{nu1} := \text{Tok1}(0) = 0;$

$\text{v1} := \text{Tok1}(0) = 0$

#Оператор запису отриманих значень в спеціальний список

> $F := \text{dsolve}(\{s1, \text{nu1}\}, \{\text{Tok1}(t)\}, \text{numeric}, \text{output} = \text{listprocedure});$

$F := [t = \text{proc}(t) \dots \text{end proc}, \text{Tok1}(t) = \text{proc}(t)$

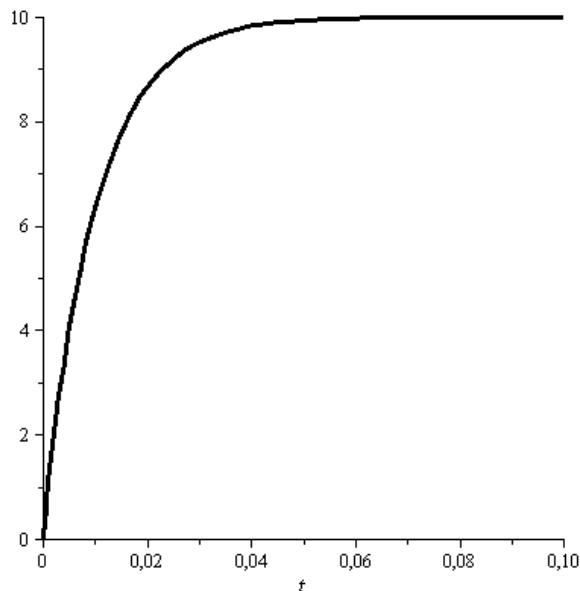
...

$\text{end proc}]$

> $\text{Tok} := \text{subs}(F, \text{Tok1}(t));$ # Підстановка значень списку в змінну

$\text{Tok} := \text{proc}(t) \dots \text{end proc}$

> $\text{plot}(\text{Tok}(t) \cdot U, t = 0 .. 0.1, \text{thickness} = 3, \text{color} = \text{black})$ #Вивід значення потужності



*Б.6.3. Розв'язання системи диференційних рівнянь
в чисельному вигляді*

Підключення повітряного трансформатора до постійної напруги

> restart;

> L1 := 0.1; r1 := 10.0; L2 := 0.2; r2 := 5.0; M
:= 0.8 * sqrt(L1 * L2); Um := 10;

L1 := 0.1

r1 := 10.0

L2 := 0.2

r2 := 5.0

M := 0.1131370850

Um := 10

> sys := L1 * diff(T1(t), t) + r1 * T1(t) + M
* diff(T2(t), t) = Um,
L2 * diff(T2(t), t) + r2 * T2(t) + M
* diff(T1(t), t) = 0;

$$\begin{aligned} \text{sys} &:= 0.1 \left(\frac{d}{dt} T1(t) \right) + 10.0 T1(t) \\ &+ 0.1131370850 \left(\frac{d}{dt} T2(t) \right) = 10, \\ 0.2 \left(\frac{d}{dt} T2(t) \right) + 5.0 T2(t) \\ &+ 0.1131370850 \left(\frac{d}{dt} T1(t) \right) = 0 \end{aligned}$$

> nu := T1(0) = 0, T2(0) = 0;

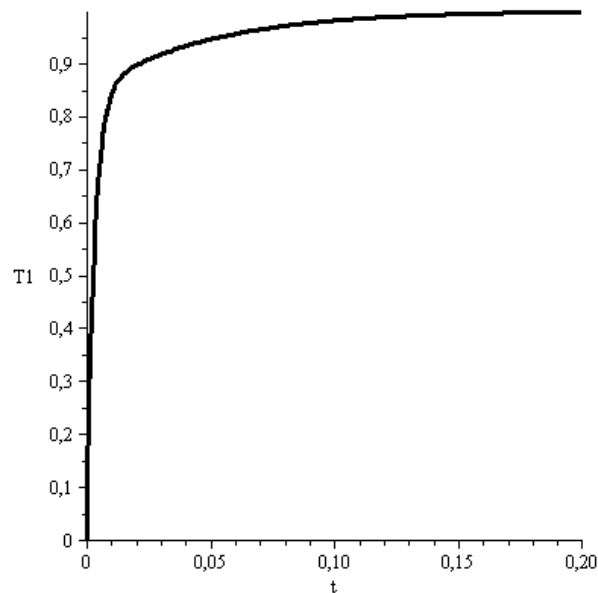
$v := T1(0) = 0, T2(0) = 0$

> $F := dsolve(\{sys, nu\}, numeric);$

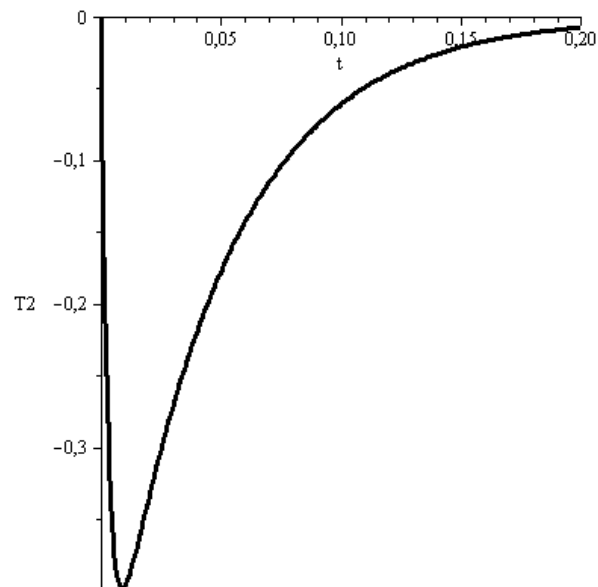
$F := \mathbf{proc}(x_rkf45) \dots \mathbf{end\ proc}$

> $with(plots) :$

> $odeplot(F, [t, T1(t)], 0..0.2, color = black,$
 $thickness = 3, numpoints = 200);$

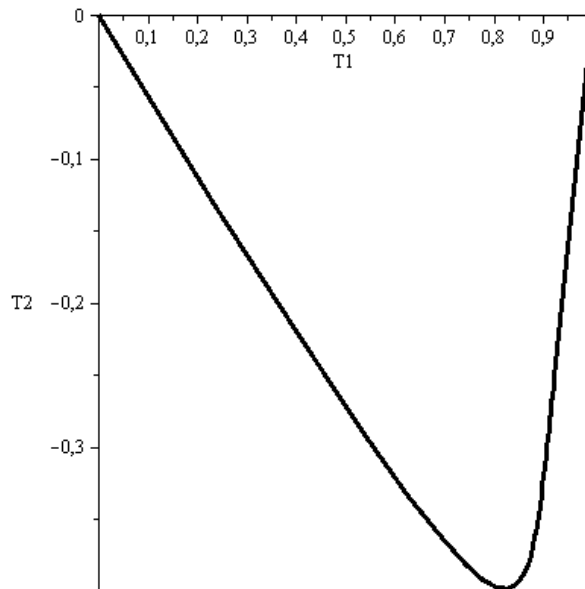


> $odeplot(F, [t, T2(t)], 0..0.2, color = black,$
 $thickness = 3, numpoints = 200);$



#Побудова значень струму 2 в функції струму 1

$odeplot(F, [T1(t), T2(t)], 0..0.2, color = black,$
 $thickness = 3, numpoints = 200);$



Підключення повітряного трансформатора до синусоїдальної напруги

> *restart*;

> $L1 := 0.1; r1 := 10.0; L2 := 0.2; r2 := 5.0; M$
 $:= 0.8 \cdot \text{sqrt}(L1 \cdot L2); Um := 10 \cdot \sin(314.2 \cdot t);$

$L1 := 0.1$

$r1 := 10.0$

$L2 := 0.2$

$r2 := 5.0$

$M := 0.1131370850$

$Um := 10 \sin(314.2 t)$

> $\text{sys} := L1 \cdot \text{diff}(T1(t), t) + r1 \cdot T1(t) + M$
 $\cdot \text{diff}(T2(t), t) = Um,$
 $L2 \cdot \text{diff}(T2(t), t) + r2 \cdot T2(t) + M$
 $\cdot \text{diff}(T1(t), t) = 0;$

$$\begin{aligned} \text{sys} &:= 0.1 \left(\frac{d}{dt} T1(t) \right) + 10.0 T1(t) \\ &+ 0.1131370850 \left(\frac{d}{dt} T2(t) \right) \\ &= 10 \sin(314.2 t), 0.2 \left(\frac{d}{dt} T2(t) \right) + 5.0 T2(t) \\ &+ 0.1131370850 \left(\frac{d}{dt} T1(t) \right) = 0 \end{aligned}$$

```
> nu := T1(0) = 0, T2(0) = 0;
```

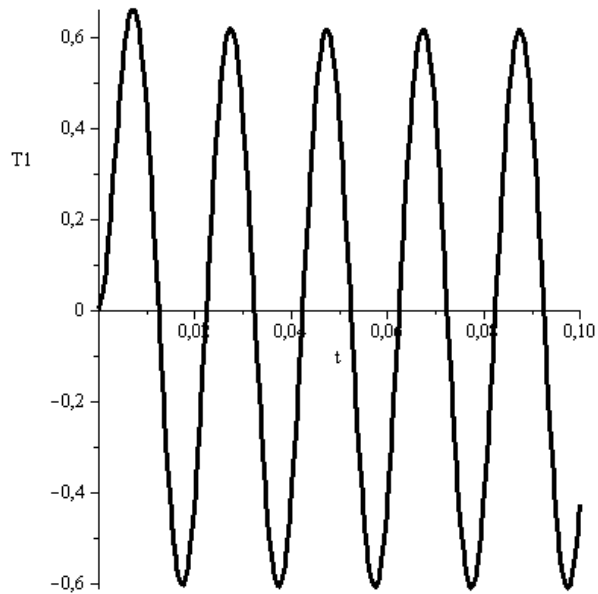
```
v := T1(0) = 0, T2(0) = 0
```

```
> F := dsolve({sys, nu}, numeric);
```

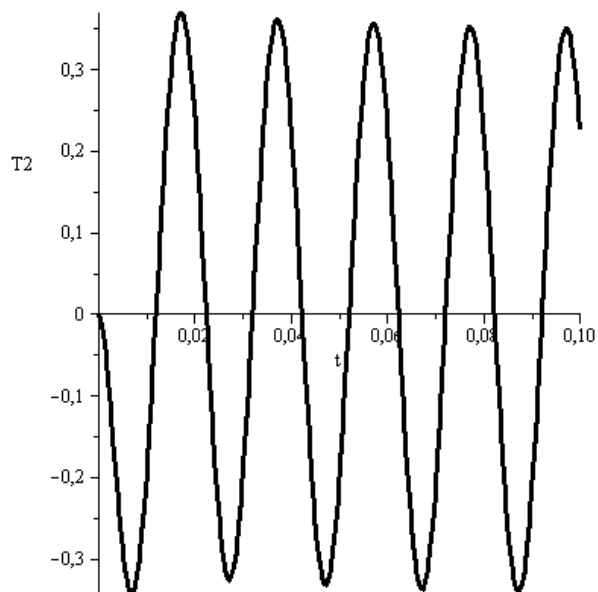
```
F := proc(x_rkf45) ... end proc
```

```
> with(plots) :
```

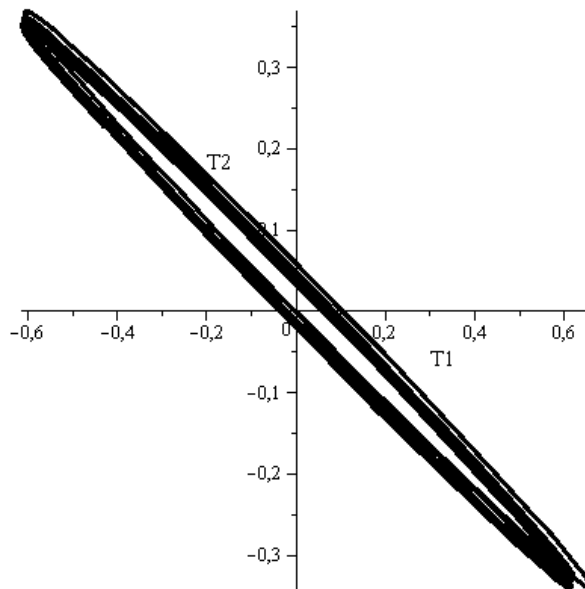
```
> odeplot(F, [t, T1(t)], 0..0.1, color = black,  
thickness = 3, numpoints = 200);
```



```
> odeplot(F, [t, T2(t)], 0..0.1, color = black,  
thickness = 3, numpoints = 200);
```



```
> odeplot(F, [T1(t), T2(t)], 0..0.1, color = black,  
thickness = 3, numpoints = 200);
```



*Б.6.4. Розв'язання нелінійних диференціальних рівнянь
в чисельному вигляді*

Великий прогин затисненої консолі під дією вертикальної сили,
розташованої на її кінці

> restart:

#Рівняння прогину консолі

> s:=diff(y(x),x,x)/sqrt(1+diff(y(x),x)^2)=-1/(E*J)*Q*(l-x);

$$s := \frac{\frac{d^2}{dx^2} y(x)}{\sqrt{1 + \left(\frac{d}{dx} y(x)\right)^2}} = -\frac{Q(l-x)}{EJ}$$

> Q:=12000;E:=2e11;J:=0.1*1e-6/12;l:=50e-2; # Сила, модуль пружності,
момент інерції і довжина

$$Q := 12000$$

$$E := 2 \cdot 10^{11}$$

$$J := 8.333333333 \cdot 10^{-9}$$

$$l := 0.50$$

> s;

$$\frac{\frac{d^2}{dx^2} y(x)}{\sqrt{1 + \left(\frac{d}{dx} y(x)\right)^2}} = -3.600000000$$

$$+ 7.200000000 x$$

> nu:=y(0)=0,D(y)(0)=0; #Граничні умови

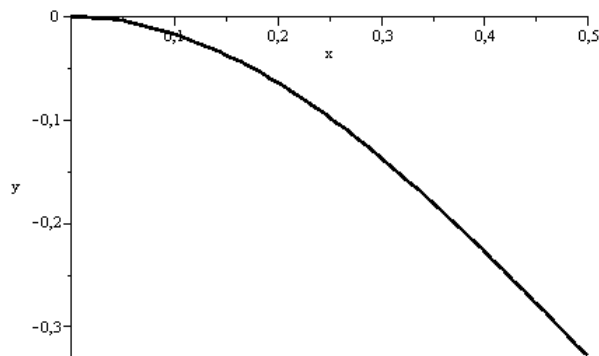
$$v := y(0) = 0, D(y)(0) = 0$$

> F:=dsolve({s,nu},{y(x)},numeric);

F := proc(x_rkf45) ... end proc

> with(plots):

> odeplot(F,[x,y(x)],0..50e-2,color=black,thickness=3);



Використання кусково-заданої функції

> restart:

> u:=piecewise(t<=t1,10.0); # Завдання прикладеної напруги кусково-заданої функцією

$$u := \begin{cases} 10.0 & t \leq t1 \\ 0 & otherwise \end{cases}$$

> s:=L*diff(tok(t),t)+r*tok(t)=u;

$$s := L \left(\frac{d}{dt} tok(t) \right) + r tok(t) = \begin{cases} 10.0 & t \leq t1 \\ 0 & otherwise \end{cases}$$

> t1:=10e-3:r:=2.0:L:=1e-2:

> s;

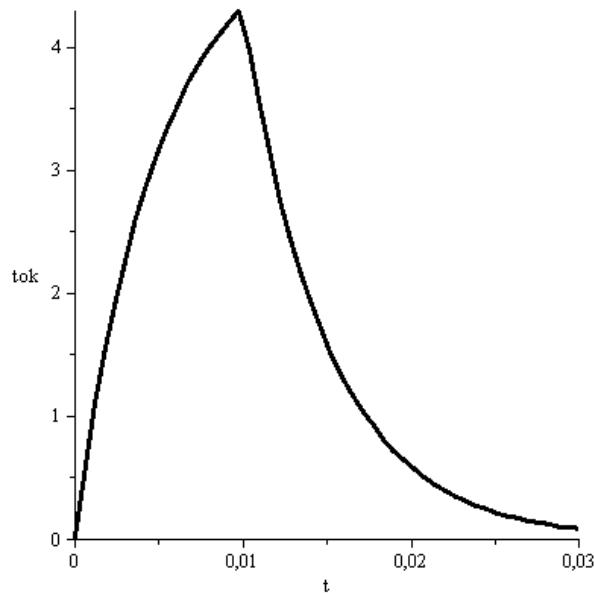
$$0.01 \left(\frac{d}{dt} tok(t) \right) + 2.0 tok(t) = \begin{cases} 10.0 & t \leq 0.010 \\ 0 & otherwise \end{cases}$$

> F:=dsolve({s,tok(0)=0.0},{tok(t)},numeric);

F := proc(x_rkf45) ... end proc

> with(plots):

> odeplot(F,[t,tok(t)],0..30e-3,thickness=3,color=black);



Б.6.5. Розв'язання диференційних рівнянь в частинних похідних

#Нагрівання складної одновимірної тришарової структури внутрішніми джерелами тепла при протіканні по ній імпульсного струму

> restart:

#Задання теплофізичних параметрів шарів

>

c1:=400:g1:=5e3:c2:=300:g2:=9e3:c3:=100:g3:=8e3:lam1:=40:lam2:=400:lam3:=100:

#Створення кусково-заданих функцій

> cg:=piecewise(x<0.25,c1*g1,x>0.25 and x<0.6,c2*g2,c3*g3);

$$cg := \begin{cases} 2.000 \cdot 10^6 & x < 0.25 \\ 2.700 \cdot 10^6 & -x < -0.25 \text{ and } x < 0.6 \\ 8.00 \cdot 10^5 & \text{otherwise} \end{cases}$$

> Lam:=piecewise(x<0.25,lam1,x>0.25 and x<0.6,lam2,lam3);

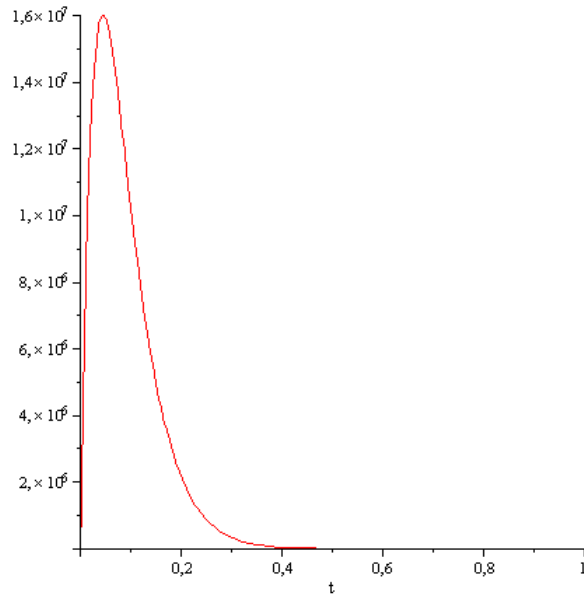
$$Lam := \begin{cases} 40 & x < 0.25 \\ 400 & -x < -0.25 \text{ and } x < 0.6 \\ 100 & \text{otherwise} \end{cases}$$

> L:=10e-2: #Довжина структури

> j:=2e8*(exp(-20.5*t)-exp(-25.5*t)); #Щільність струму в структурі

$$j := 2 \cdot 10^8 e^{-20.5 t} - 2 \cdot 10^8 e^{-25.5 t}$$

> plot(j,t=0..1); #Часова залежність імпульсної щільності струму



> r1:=100e-8:r2:=2e-8:r3:=25e-8; #Питомі електричні опори і їх температурна залежність

> ro1:=r1*(1+4e-2*u(x,t));ro2:=r2*(1+4e-3*u(x,t));ro3:=r3*(1+4e-2*u(x,t));

$$ro1 := 0.00000100 + 4.00 \cdot 10^{-8} u(x, t)$$

$$ro2 := 2. \cdot 10^{-8} + 8. \cdot 10^{-11} u(x, t)$$

$$ro3 := 2.5 \cdot 10^{-7} + 1.00 \cdot 10^{-8} u(x, t)$$

#Потужність внутрішніх джерел

> f:=piecewise(x<0.25,ro1*j^2,x>=0.25 and x<0.6,ro2*j^2,ro3*j^2);

$$f := \begin{cases} (0.00000100 + 4.00 \cdot 10^{-8} u(x, t)) (2. \cdot 10^8 e^{-20.5 t} - 2. \cdot 10^8 e^{-25.5 t})^2 & x < 0.25 \\ (2. \cdot 10^{-8} + 8. \cdot 10^{-11} u(x, t)) (2. \cdot 10^8 e^{-20.5 t} - 2. \cdot 10^8 e^{-25.5 t})^2 & 0.25 - x \leq 0 \text{ and } x < 0.6 \\ (2.5 \cdot 10^{-7} + 1.00 \cdot 10^{-8} u(x, t)) (2. \cdot 10^8 e^{-20.5 t} - 2. \cdot 10^8 e^{-25.5 t})^2 & \text{otherwise} \end{cases}$$

#Запис рівняння в частинних похідних

PDE := cg*diff(u(x,t),t)=Lam*diff(u(x,t),x,x)/L^2+f;

$$\begin{aligned}
PDE := & \left(\begin{array}{l} 2.000 \cdot 10^6 \\ 2.700 \cdot 10^6 \\ 8.00 \cdot 10^5 \end{array} \begin{array}{l} x < 0.25 \\ -x < -0.25 \text{ and } x < 0.6 \\ \text{otherwise} \end{array} \right) \left(\frac{\partial}{\partial t} u(x, t) \right) \\
= & 100.0000000 \left(\begin{array}{l} 40 \\ 400 \\ 100 \end{array} \begin{array}{l} x < 0.25 \\ -x < -0.25 \text{ and } x < 0.6 \\ \text{otherwise} \end{array} \right) \left(\frac{\partial^2}{\partial x^2} u(x, \right. \\
& \left. t) \right) + \left(\begin{array}{l} (0.00000100 \\ + 4.00 \cdot 10^{-8} u(x, t) \\ (2. \cdot 10^8 e^{-20.5 t} \\ - 2. \cdot 10^8 e^{-25.5 t})^2 \\ (2. \cdot 10^{-8} + 8. \cdot 10^{-11} u(x, \\ t)) (2. \cdot 10^8 e^{-20.5 t} \\ - 2. \cdot 10^8 e^{-25.5 t})^2 \\ (2.5 \cdot 10^{-7} \\ + 1.00 \cdot 10^{-8} u(x, t) \\ (2. \cdot 10^8 e^{-20.5 t} \\ - 2. \cdot 10^8 e^{-25.5 t})^2 \end{array} \begin{array}{l} x < 0.25 \\ 0.25 - x \leq 0 \text{ and } x \\ < 0.6 \\ \text{otherwise} \end{array} \right)
\end{aligned}$$

> IBC := {u(x,0)=0, D[1](u)(0,t)=0,D[1](u)(1,t)=0};#Початкові і граничні умови

$$IBC := \{u(x, 0) = 0, D_1(u)(0, t) = 0, D_1(u)(1, t) = 0\}$$

> pds := pdsolve(PDE,IBC,numeric,spacestep=1/100,time=t,range=0..1);

#Функція чисельного вирішення

pds := module()

export plot, plot3d, animate, value, settings;

...

end module

#Отримані рішення в різні моменти часу

> p1 := pds:-plot(t=0.1,color=red):

p2 := pds:-plot(t=0.2,color=black):

p3 := pds:-plot(t=0.3,color=green):

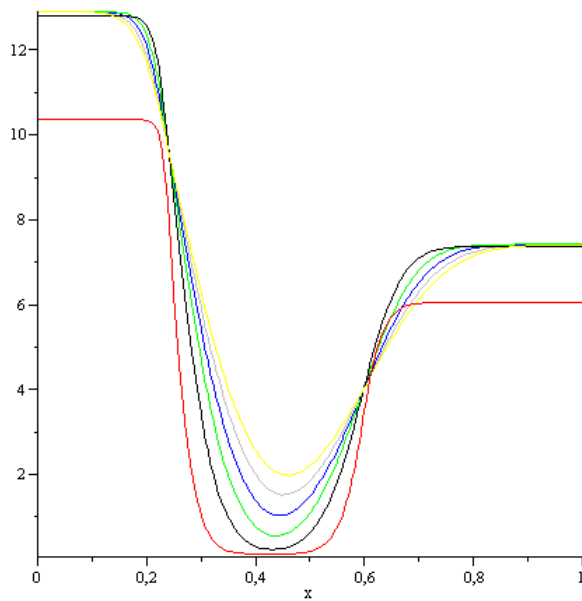
p4 := pds:-plot(t=0.4,color=blue):

p5 := pds:-plot(t=0.5,color=gray):

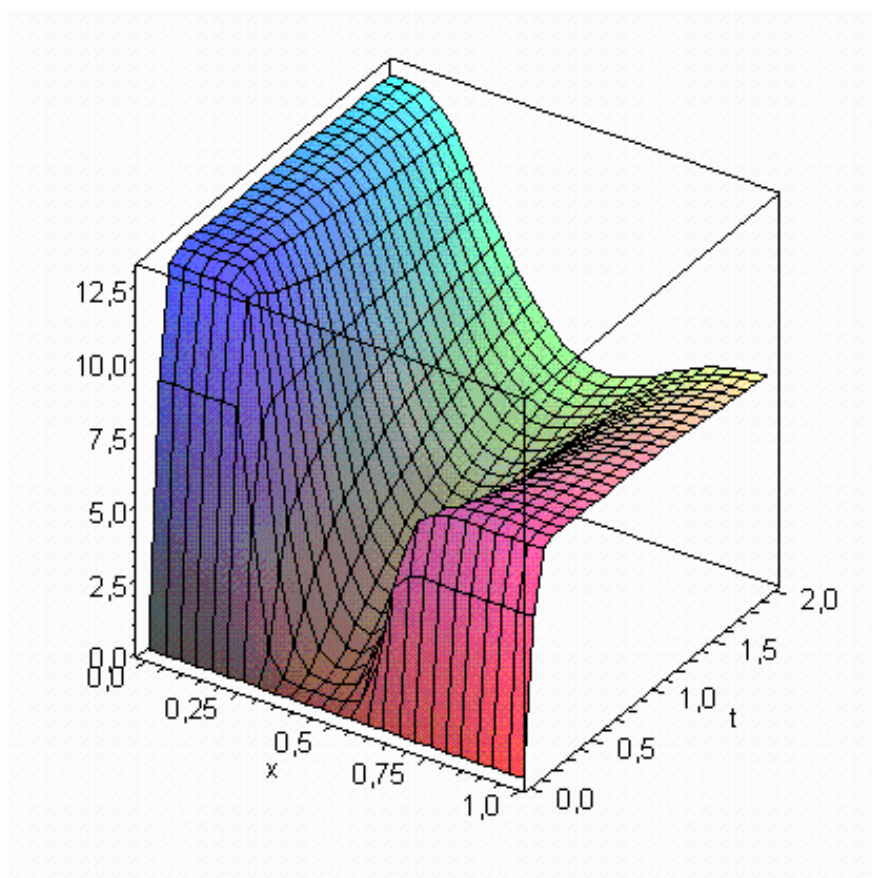
p6 := pds:-plot(t=0.6,color=yellow):

p7 := pds:-plot(t=0.7,color=yellow):

```
p8 := pds:-plot(t=0.8,color=yellow):
plots[display]({p1,p2,p3,p4,p5,p6});
```



```
> pds:-plot3d(t=0..2.0,x=0..1); #Рішення, яке задане у вигляді тривимірного графіку
```



```
pds:-animate(t=1.0,frames=100); #Функція анімації отриманого рішення
# Вирішення задачі теорії пружності.
# Ударна дія на довгий сталевий стрижень
```

> restart;

> $S := 5.0 \cdot 10^{-4}$; $E := 1.0 \cdot 10^9$; $l := 50.0$; $m := S \cdot 7.8 \cdot 10^3$; # Перетин, модуль пружності, довжина і маса одиниці довжини

1.00000000010⁹

50.0

3.900000000

> $a2 := \frac{E \cdot S}{m}$; $a := \text{sqrt}(a2)$; $T := \frac{l}{a}$; #Швидкість руху хвилі і час досягнення хвилею кінця стрижня

1.28205128210⁵

358.0574370

0.139642400

> $PDE := \text{diff}(u(x, t), t, t) = a2 \cdot \text{diff}(u(x, t), x, x)$ #Вигляд рівняння

$$\frac{\partial}{\partial t} \left(\frac{\partial}{\partial t} u(x, t) \right) = 1.28205128210^5 \left(\frac{\partial}{\partial x} \left(\frac{\partial}{\partial x} u(x, t) \right) \right)$$

> $N := 2 \cdot 10^6 \cdot (\exp(-200 \cdot t) - \exp(-250 \cdot t))$; #Часова залежність ударного навантаження

$$2000000e^{-200t} - 2000000e^{-250t}$$

#Запис початкових і граничних умов: нерухомість одного з кінців стрижня; задання відносної деформації під дією ударного навантаження; нульове значення початкового зсуву і швидкості.

> $IBS := \left\{ u(l, t) = 0, D[1](u)(0, t) = \frac{N}{E \cdot S}, u(x, 0) = 0, D[2](u)(x, 0) = 0 \right\};$

$$\left\{ u(50.0, t) = 0, u(x, 0) = 0, D_1(u)(0, t) = 4.000000000e^{-200t} - 4.000000000e^{-250t}, D_2(u)(x, 0) = 0 \right\}$$

#Форма запису чисельного вирішення рівняння з заданим кроком по змінній t .

> $pds := \text{pdsolve} \left(PDE, IBS, \text{numeric}, \text{spacestep} = \frac{T}{10}, \text{time} = t \right);$

module()

export plot, plot3d, animate, value, settings;

...

end module

Рішення в зазначені моменти часу із зазначенням кольору побудови графіка і товщини ліній.

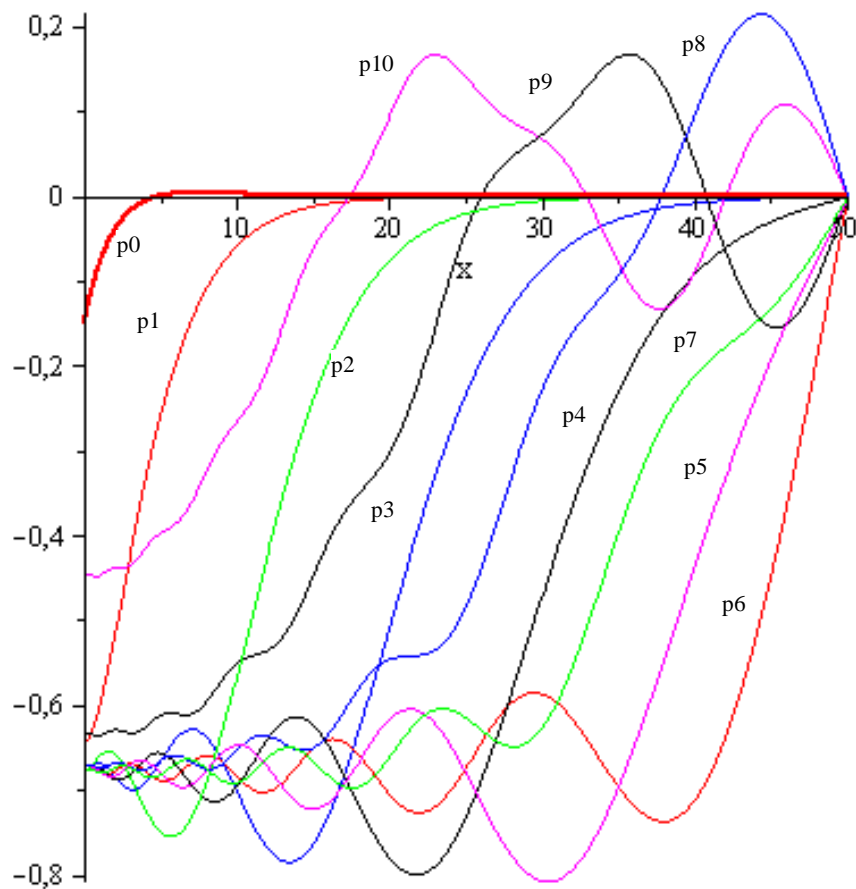
```

p0 := pds:-plot( t = T/20, color = red, thickness = 2 ):
p1 := pds:-plot( t = 2 * T/10, color = red ):
p2 := pds:-plot( t = 4 * T/10, color = green ):
p3 := pds:-plot( t = 6 * T/10, color = blue ):
p4 := pds:-plot( t = 8 * T/10, color = black ):
p5 := pds:-plot( t = T, color = magenta ):
p6 := pds:-plot( t = 1.2 * T, color = red ):
p7 := pds:-plot( t = 1.4 * T, color = green ):
p8 := pds:-plot( t = 1.6 * T, color = blue ):
p9 := pds:-plot( t = 1.8 * T, color = black ):
p10 := pds:-plot( t = 2 * T, color = magenta ):

```

#Підключення бібліотеки *display* і графіки побудови проходження ударної хвилі

```
plots[display]({p0, p1, p2, p3, p4, p5, p6, p7, p8, p9, p10})
```



Б.7. Визначення екстремуму функції

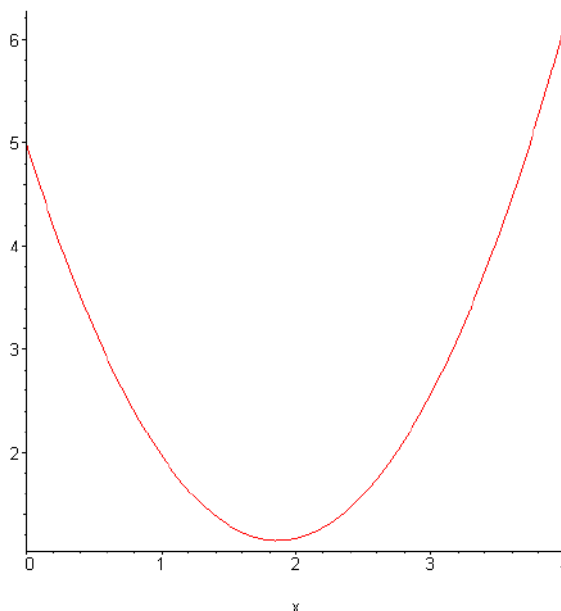
Б.7.1. Визначення екстремуму функції однієї змінної

> **restart:**

> **y:=x->1.1*x^2-4.1*x+4.97;**

$$y := x \rightarrow 1.1x^2 - 4.1x + 4.97$$

> **plot(y(x),x=0..4);**



> **minimize(y(x),x=0..4);**

1.149545455

> **minimize(y(x),x=0..4,location=true);**

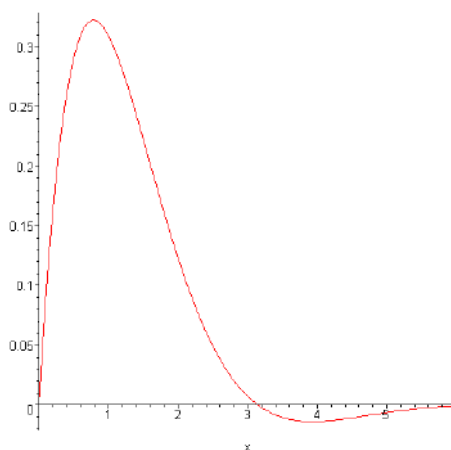
1.149545455 [[{ x = 1.863636364 }, 1.149545455]]

> **restart:**

> **y:=x->exp(-x)*sin(x);**

$$y := x \rightarrow e^{-x} \sin(x)$$

> **plot(y(x),x=0..6);**



> **maximize(y(x),x=0..4,location=true);**

$$\frac{1}{2} e^{\left(-\frac{\pi}{4}\right)} \sqrt{2}, \left\{ \left\{ x = \frac{\pi}{4} \right\}, \frac{1}{2} e^{\left(-\frac{\pi}{4}\right)} \sqrt{2} \right\}$$

> **evalf(%);**

0.3223969418 { [{ x = 0.7853981635 }, 0.3223969418] }

> **minimize(y(x),x=0..6,location=true);**

$$-\frac{1}{2} e^{\left(-\frac{5\pi}{4}\right)} \sqrt{2}, \left\{ \left\{ x = \frac{5\pi}{4} \right\}, -\frac{1}{2} e^{\left(-\frac{5\pi}{4}\right)} \sqrt{2} \right\}$$

> **evalf(%);**

-0.01393203508 { [{ x = 3.926990818 }, -0.01393203508] }

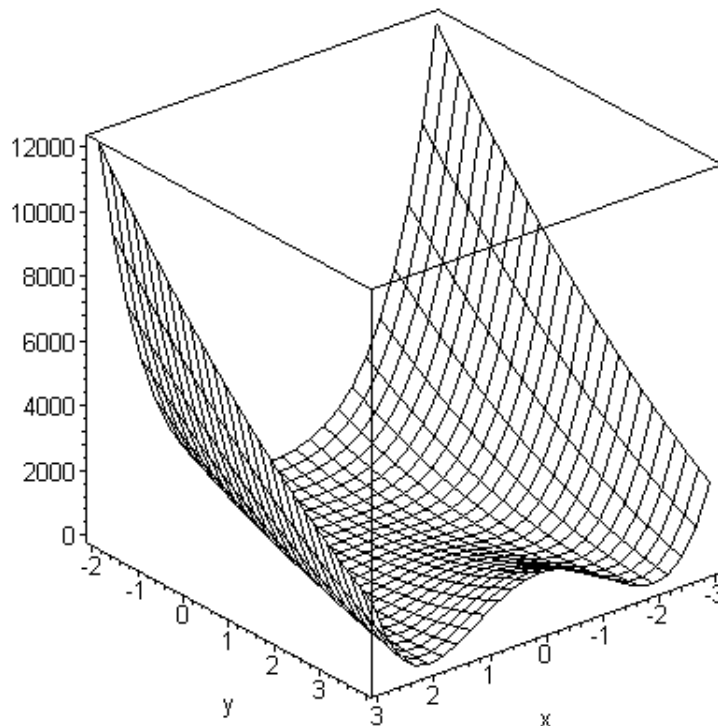
Б.7.2. Визначення екстремуму функції двох змінних

> **restart;**

> **rf:=(x,y)->100*(y-x^2)^2+(1-x)^2;**

$$rf := (x, y) \rightarrow 100(y - x^2)^2 + (1 - x)^2$$

> **plot3d(rf(x,y),x=-3..3,y=-2..4);**



> **minimize(rf(x,y),x=-3..3,y=-2..4,location);**

0, { [{ y = 1, x = 1 }, 0] }

Б.7.3. Визначення екстремуму функції методом Монте-Карло

> **restart :**

> **# -3 ≤ x ≤ 3; -2 ≤ y ≤ 4 #Границі області пошуку оптимуму**

> **randomize() #Установка випадкового значення послідовності випадкових**

чисел

144491660:

> rf := 100*(y-x^2)^2 + (1-x)^2; # Функція Розенброка

rf := 100 (y - x²)² + (1 - x)²

> x := -3 + (3 - (-3))·u;

x := -3 + 6 u

> y := -2 + (4 - (-2))·v;

y := -2 + 6 v

> f := unapply (rf, u, v); #Перетворювальна функція Розенброка

f := (u, v) → 100 (-2 + 6 v - (-3 + 6 u)² + (4 - 6 u)²

>

> Min := 1000.0; # Задання апріорі мінімуму

Min := 1000.0

> N := 20000; #Число випадкових точок

N := 20000

> for i from 1 to N do

u := rand() * 1.0 · 10⁻¹² ;

v := rand() * 1.0 · 10⁻¹² ;

if (f(u, v) < Min) then Min := f(u, v); Xmin := -3 + (3 - (-3))
· u;

Ymin := -2 + (4 - (

-2))·v;

fi;

od:

> Min; Xmin; Ymin; # Значення екстремуму і його координати

0.00115657046

0.98078165:

0.96473840:

Порівняння отриманих значень з раніше знайденими значеннями показує, що для складних функцій пошук екстремуму методом Монте-Карло вимагає великої кількості генерації випадкових значень або зміни алгоритму пошуку

Навчальне видання

БАЙДА Євген Іванович
КРОПАЧЕК Ольга Юріївна

ЧИСЕЛЬНІ МЕТОДИ

Навчально-методичний посібник для магістрів та аспірантів спеціальності
141 «Електроенергетика, електротехніка і електромеханіка» усіх форм
навчання

Українською мовою

Відповідальний за випуск проф. Ю. С. Грищук
Роботу до видання рекомендував проф. В. І. Мілих

Редактор Л.Л. Яковлева

План 2020 р., поз 74

Підп. до друку _____. Формат 60×84 1/16. Папір офсетний..

Riso-друк. Гарнітура Таймс. Ум. друк. арк. 5,2.

Наклад 20 прим. Зам. № . Ціна договірна

Видавничий центр НТУ «ХП».

Свідоцтво про державну реєстрацію ДК № 3657 від 24.12.2009 р.

61002, Харків, вул. Кирпичова, 2
