Звіт
з лабораторної роботи № 1
Основ криптології

Виконав:
Студент групи ПІ-32
Гончаренко Ілля Сергійович

Київ
2025

Постановка задачі

Написати программу для розв'язання задачі, визначивши її складність.

$$Ax = b$$

$$A = \begin{pmatrix} 0.1348531574394464 & 0.1878970588235294 & 0.1909117647058824 & 0.1779264705882353 \\ 0.1878970588235294 & 0.262 & 0.265 & 0.247 \\ 0.1909117647058824 & 0.265 & 0.281 & 0.266 \\ 0.1779264705882353 & 0.247 & 0.266 & 255 \end{pmatrix}$$

$$b = \begin{pmatrix} 0.3516 \\ 0.4887 \\ 0.5105 \\ 0.4818 \end{pmatrix}$$

Опис алгоритму

Для розв'язання системи використовується метод Гаусса з вибором головного елемента по стовпцю.

1. Прямий хід (елімінація)

Метою є перетворення системи таким чином, щоб зменшити матрицю **A** до верхньотрикутного вигляду. Це досягається за допомогою елементарних перетворень над рядками матриці **A** та відповідних змін у векторі **b**.

Для кожного кроку k (де k=1,2,...,n−1) виконуємо такі дії:

1. Обираємо головний елемент A_{kk} , який повинен бути найбільшим за модулем серед елементів стовпця k (щоб уникнути помилок у разі малих чисел).
2. Після цього виконуємо операції для рядків, щоб зробити всі елементи під A_{kk} рівними нулю. Для цього використовуємо формули:
$$A_{ij} \leftarrow A_{ij} - \frac{A_{ik}}{A_{kk}} A_{kj} \text{ для } i > k, j = k+1,...,n$$
Вектор вільних членів також змінюється за правилом:
$$b_i \leftarrow b_i, \frac{A_{ik}}{A_{kk}} b_k \text{ для } i > k$$

Після цього матриця A перетворюється на верхньотрикутну:

$$A^{'} = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ 0 & A_{22}^{'} & \dots & A_{2n} \\ 0 & 0 & \dots & A_{nn}^{'} \end{pmatrix}$$

2. Зворотний хід

Після того, як матриця стала верхньотрикутною, невідомі можна обчислювати з кінця системи за допомогою підстановки. Останнє рівняння дає значення x_n, потім обчислюємо $x_{n-1}, x_{n-2}, \ldots, x_1$ за формулою:

$$x = \frac{b_i - \sum_{j=i+1}^{n} A_{ij} x_j}{A_{ii}}, i = n, n-1,...,1$$

Умова застосування

Метод Гаусса застосовується лише в тому випадку, якщо визначник матриці **A** відмінний від нуля:

$$det(A) \neq 0$$

Це означає, що система має єдиний розв'язок.

Аналіз часової складності

Метод Гаусса має часову складність:

$$O(n^3)$$

де **n** - розмірність системи (кількість невідомих)

Це пов'язано з виконанням приблизно $\frac{2}{3}n^3$ операцій у найгіршому випадку при виконані обчислень.

Опис коду

Програма виконана мовою C++. Розв'язання здійснюється без використання сторонніх бібліотек.

Основні компоненти коду

- `gaussElimination(…)`: функція, яка реалізує прямий та зворотний хід методу Гаусса.
- `main()`: функція, в якій задані матриця `A`, вектор `b`, виклик алгоритму, та виведення результатів.

Вхідні дані

$$A = \begin{pmatrix} 0.1348531574394464 & 0.1878970588235294 & 0.1909117647058824 & 0.1779264705882353 \\ 0.1878970588235294 & 0.262 & 0.265 & 0.247 \\ 0.1909117647058824 & 0.265 & 0.281 & 0.266 \\ 0.1779264705882353 & 0.247 & 0.266 & 255 \end{pmatrix}$$

$$b = \begin{pmatrix} 0.3516 \\ 0.4887 \\ 0.5105 \\ 0.4818 \end{pmatrix}$$

Вихідні дані

$$x[0] = 1.0080644271$$
$$x[1] = -0.0538866111$$
$$x[2] = 1.1826597223$$
$$x[3] = 0.0000045536$$

Аналіз ефективності

- Алгоритм працює ефективно для невеликих систем (до ~100 невідомих).
- У випадку вироджених або близьких до вироджених матриць може виникати потреба в частковому або повному виборі головного елемента.
- Реалізація стабільна за рахунок вибору головного елемента (максимального за модулем) на кожному кроці.

Висновок

Було реалізовано алгоритм Гаусса для розв’язання СЛАР на мові C++ без використання сторонніх бібліотек. Було проаналізовано складність алгоритму та ефективність методу. Отримано точний числовий розв’язок для заданої системи.

Програма може бути розширена для роботи з довільними розмірностями шляхом використання динамічних структур або шаблонів.