Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інтелектуальних програмних систем

Алгоритми та складність

Завдання №22-1

«Побудова оберненої матриці методом LU-розкладання»

Варіант №5

Виконала студентка 2-го курсу

Групи ІПС-22

Клевчук Марія Вячеславівна

Київ - 2024

Завдання

Реалізація алгоритму побудови оберненої раціональної матриці типу vector<vector<T>> методом LU-розкладання. В проєкті використаний алгоритм Штрассена для множення матриць.

Алгоритм Штрассена для множення матриць був реалізований у проєкті Гулковської Олександри.

Теорія

Побудова оберненої матриці шляхом LU-розкладання — це метод, який передбачає розкладання матриці A на добуток двох трикутних матриць: нижньої трикутної матриці L та верхньої трикутної матриці U (тобто A = LU). Далі, щоб знайти обернену матрицю A^{-1}, розв'язуються системи рівнянь для кожного стовпця одиничної матриці, використовуючи матриці L і U.

Цей метод дозволяє обчислити обернену матрицю швидше і з меншою обчислювальною складністю, оскільки розв'язання трикутних систем лінійних рівнянь є відносно простим завдяки послідовним підстановкам.

Алгоритм LU-розкладання дозволяє представити квадратну матрицю A як добуток нижньої трикутної матриці L і верхньої трикутної матриці U (тобто A = LU). Цей метод полягає у послідовному обчисленні елементів L і U з урахуванням певних умов на їхні значення.

LU-розкладання дозволяє спростити обчислення, зокрема розв’язання систем лінійних рівнянь та обчислення обернених матриць, завдяки зниженню обчислювальної складності.

Обернена матриця для квадратної матриці A — це така матриця A^{-1}, що добуток A та A^{-1} дорівнює одиничній матриці E. Обернена матриця існує лише для квадратних матриць, визначник яких не дорівнює нулю.

Алгоритм

1. Для обчислення оберненої матриці A^{-1} за допомогою LU-розкладу, матрицю A потрібно розкласти на дві трикутні матриці — нижню трикутну матрицю L і верхню трикутну матрицю U. Після цього алгоритм складається з кількох етапів:

2. Потрібно розв'язати системи рівнянь для кожного стовпця оберненої матриці.

Обернену матрицю A^{-1} можна представити як матрицю, яка складається з стовпців X_i, де кожен стовпець X_i є розв'язком системи рівнянь AX = E, де E — одинична матриця.

Отже, для обчислення оберненої матриці потрібно знайти стовпці X_i, розв'язавши для кожного стовпця систему рівнянь $LU*X_i = E_i$.

3. Повторення для кожного стовпця одиничної матриці E, щоб обчислити всі стовпці оберненої матриці A^{-1}.

Складність алгоритму

Складність обчислення оберненої матриці методом LU-розкладу: $O(n^3)$

Мова реалізації алгоритму

C++

Модулі програми.

Клас Fraction - клас для роботи з раціональними числами. Його методи:

void simplify() - метод, що скорочує дріб, використовуючи найбільший спільний дільник чисельника і знаменника.

Fraction operator+(const Fraction& other) const - додає поточний дріб із іншим, повертаючи результат як новий об'єкт Fraction.

Fraction operator-(const Fraction& other) const - віднімає інший дріб від поточного, повертаючи результат як новий об'єкт Fraction.

Fraction operator*(const Fraction& other) const - перемножує поточний дріб з іншим, повертаючи результат як новий об'єкт Fraction.

Fraction operator/(const Fraction& other) const - ділить поточний дріб на інший, повертаючи результат як новий об'єкт Fraction.

friend istream& operator>>(istream& is, Fraction& frac) - зчитує дріб зі стандартного вводу у форматі чисельник/знаменник.

friend ostream& operator<<(ostream& os, const Fraction& frac) - виводить дріб у форматі чисельник/знаменник.

Клас Matrix - клас для роботи з квадратними матрицями, що містять елементи у вигляді звичайних дробів (Fraction).

void input() - зчитує елементи матриці зі стандартного вводу як дроби.

Matrix inverse() const - обчислює обернену матрицю, використовуючи LU-розклад, і повертає її як новий об'єкт Matrix.

void display() const - виводить елементи матриці у форматі дробів у консоль.

Matrix(const vector<vector<Fraction>>& data) -створює матрицю на основі вказаних даних у вигляді двовимірного вектора дробів.

Опис інтерфейсу користувача.
Введення даних відбувається через консоль користувачем.

Вхідні дані:

- Розмір матриці A

- Елементи матриці A

Вихідні дані:

- Обчислена обернена матриця A^{-1}

Тестові приклади

Приклад №1

```
Enter the size of the matrix (n x n): 3
Enter elements of the matrix in the format `numerator/denominator`:
1/1
2/1
1/2
2/1
5/1
3/1
1/1
3/1
2/1

The inverse of matrix A is:
-2/1 5/1 -7/1
2/1 -3/1 4/1
-2/1 2/1 -2/1
Program ended with exit code: 0
```

Приклад №2

```
Enter the size of the matrix (n x n): 3
Enter elements of the matrix in the format `numerator/denominator`:
1/1
6/5
4/5
8/9
3/4
1/1
0/1
0/1
5/6

The inverse of matrix A is:
-45/19 72/19 -216/95
160/57 -60/19 104/95
0/1 0/1 6/5
Program ended with exit code: 0
```

Висновки

У даній роботі було реалізовано алгоритм обчислення оберненої матриці методом LU-розкладу. Цей метод дозволяє зменшити обчислювальну складність порівняно з традиційним методом обчислення оберненої матриці, зокрема завдяки розкладанню матриці на дві трикутні матриці L і U. Такий підхід спрощує процес розв'язання систем лінійних рівнянь, необхідних для знаходження оберненої матриці, і зменшує кількість обчислень.

Метод LU-розкладу має часову складність O(n^3), що робить його ефективним для обчислення обернених матриць середнього розміру.

Завдяки виконаним обчисленням ми побачили, що метод LU-розкладу є надійним інструментом для обчислення оберненої матриці і може використовуватися у багатьох завданнях, де важливо швидко знаходити обернені матриці, наприклад, у розв'язанні систем рівнянь та при роботі з графами.

Використані літературні джерела

https://uk.wikipedia.org/wiki/LU-розклад_матриці

https://www.mathros.net.ua/metod-lu-faktoryzacii-dlja-rozvjazuvannja-systemy-linijnyh-rivnjan.html

https://en.wikipedia.org/wiki/Invertible_matrix