**Mục lục**

[I. Ý tưởng phương pháp 2](#_Toc104528730)

[II. Lý thuyết ma trận 2](#_Toc104528731)

[1. Các phép biến đổi ma trận 2](#_Toc104528732)

[2. Ma trận bậc thang 2](#_Toc104528733)

[3. Hạng của ma trận 3](#_Toc104528734)

[III. Cách xét nghiệm 3](#_Toc104528735)

[IV. Công thức khử 3](#_Toc104528736)

[V. Thuật toán 3](#_Toc104528737)

[1. Thuật toán tổng quát 3](#_Toc104528738)

[2. Thuật toán chi tiết 4](#_Toc104528739)

[a. quá trình thuận 4](#_Toc104528740)

[b. tìm rankA và rankB 5](#_Toc104528741)

[c. quá trình nghịch 6](#_Toc104528742)

[V. Đánh giá phương pháp 7](#_Toc104528743)

# I. Ý tưởng phương pháp

Thực hiện lần lượt quy trình thuận và quy trình nghịch để giải biến đổi ma trận bổ sung và kết luận nghiệm của hệ phương trình

- Quy trình thuận: Dùng phép biến đổi ma trận để đưa ma trận mở rộng về ma trận bậc thang

- Quy trình nghịch: Dùng phép thế dần dần các biến từ dưới lên cho đến khi tìm được các nghiệm của hệ phương trình

# II. Lý thuyết ma trận

## 1. Các phép biến đổi ma trận

- đổi chỗ hai hàng

- Nhân một số khác 0 bất kì vào một hàng

- Nhân một hàng với một số rồi cộng với hàng khác

## 2. Ma trận bậc thang

- Một ma trận được gọi là ma trận bậc thang nếu thỏa mãn

+ nếu có hàng 0 (mọi phần tử trên hàng bằng 0) thì nó phải nằm dưới các hàng có phần tử khác 0

+ với hai hàng khác 0, phần tử khác 0 đầu tiên kể từ trái của hàng dưới nằm ở bên phải cột chứa phần tử khác 0 đầu tiên của hàng trên

## 3. Hạng của ma trận

- Hạng của ma trận là số hàng khác 0 của ma trận đó

# III. Cách xét nghiệm

- gọi rank(A), rank(B) là hạng của ma trận A và ma trận bổ sung B:

+ rank(A) rank(B): hệ vô nghiệm

+ rank(A) rank(B) n: hệ có nghiệm duy nhất

+ rank(A) rank(B) < n: hệ có vô số nghiệm

# IV. Công thức khử

# V. Thuật toán

## 1. Thuật toán tổng quát

Input: m, n, A, B

Trong đó: m, n là số hàng và số cột của ma trận A

Output: Kết luận nghiệm của hệ phương trình

Bước 1: nhập input

Bước 2: sử dụng quá trình thuận để đưa ma trận bổ sung về ma trận bậc thang

Bước 3: tính rankA của ma trận A và rankB của ma trận bổ sung AB

Bước 4: kiểm tra rankA và rankB

+ rank(A) rank(B): hệ vô nghiệm

+ rank(A) rank(B) n: hệ có nghiệm duy nhất

+ rank(A) rank(B) < n: hệ có vô số nghiệm

Bước 5: sử dụng quá trình nghịch để đưa ra nghiệm của hệ phương trình

Bước 6: in ra output

## 2. Thuật toán chi tiết

### a. quá trình thuận

Input: m, n, AB

Trong đó: m, n là số hàng và số cột của ma trận A

AB là ma trận bổ sung

Output: ma trận bổ sung AB sau khi biến đổi

Bước 1: nhập input

Bước 2: khởi tạo i = 1, ind[]1xm

Bước 3: Nếu i <= m thì chuyển sang bước 4

Bước 4: khởi tạo j = i

Bước 5: nếu j <= n thì chuyển sang bước 6, ngược lại thì tăng i++ rồi quay lại bước 3

Bước 6: kiểm tra nếu ABij 0 thì ind[i] = j, ngược lại thì chuyển sang bước 10

Bước 7: khởi tạo k = i + 1, l = 0

Bước 8: nếu k <= m thì chuyển sang bước 9, ngược lại thì tăng i++ rồi quay lại bước 3

Bước 9: nếu l <= n + 1 thì ABkl = ABkl – .ABil, ngược lại thì tăng k++ rồi quay lại bước 8

Bước 10: khởi tạo t = i + 1

Bước 11: Nếu t <= m thì chuyển sang bước 12, ngược lại thì j++ rồi quay lại bước 5

Bước 12: nếu ABtj  0 thì đổi chỗ hai hàng i và t rồi quay lại bước 4, ngược lại thì tăng i++ rồi quay lại bước 11

Bước 13: in ra output

### b. tìm rankA và rankB

Input: A[][]

Output: rankA, rankB

Bước 1: nhập input

Bước 2: khởi tạo rankA = 0, rankB = 0

Bước 3: khởi tạo i = 1 để đo số vòng lặp

Bước 4: nếu i <= m thì chuyển sang bước 5, ngược lại thì chuyển sang bước 10

Bước 5: nếu MaxH[i, n] 0 thì chuyển sang bước 6, ngược lại thì chuyển sang bước 7

Trong đó: MaxH[i, n] là giá trị lớn nhất của hàng i từ cột 1 đến n

Bước 6: tăng rankA ++

Bước 7: nếu MaxH[i, n + 1] 0 thì chuyển sang bước 8, ngược lại chuyển sang bước 9

Bước 8: tăng rankB ++

Bước 9: tăng i++ rồi quy lại bước 4

Bước 10: in ra output

### c. quá trình nghịch

Input: m, n, A, B

Output: nghiệm của hệ phương trình

Bước 1: nhập input

Bước 2: nếu hệ vô nghiệm thì kết luận hệ vô nghiệm ngược lại thì chuyển sang bước 3

Bước 3: nếu hệ có 1 nghiệm thì chuyển sang bước 4, ngược lại thì chuyển sang bước 6

Bước 4: khởi tạo i = m

Bước 5: nếu i >= 1 thì xi = Bi - , ngược lại thì in ra nghiệm của hệ phương trình

Bước 6: khởi tạo i = m

Bước 7: nếu i >= 1 thì chuyển sang bước 8, ngược lại thì in ra nghiệm của phương trình

Bước 8: nếu ind[i – 1] – ind[i] > 1

# V. Đánh giá phương pháp

1. Ưu điểm

- Giải được tất cả các phương trình đại số tuyến tính kể cả ma trận A không vuông

- Dễ dàng lập trình chạy trên máy tính để tính toán lời giải

- Tốc độ tìm ra nghiệm nhanh hơn phương pháp Gauss - Jordan

2. Nhược điểm

* Sai số trong tính toán với số gần 0 lớn, điều này sẽ được khác phục trong phương pháp Gauss Jordan
* Sai số trong quá trình tính toán không thể kiểm soát được
* Độ phức tạp của thuật toán lớn: O(n^3)

\*Chú ý:

* Đối với những ma trận hệ số đơn giản, hoặc có thể coi là “ước”, “bội” của nhau thì kết quả nhận được sẽ hoàn toàn đúng, các phép tính đơn giản sẽ không có ảnh hưởng đến sai số
* Đối với những ma trận cỡ lớn thì nên sử dụng các phương pháp lặp để giải vì tốc độ hội tụ nó sẽ nhanh hơn rất nhiều so với phương pháp Gauss