**MỤC LỤC**

[I. Ý tưởng phương pháp 2](#_Toc109734170)

[II. Chuẩn ma trận 2](#_Toc109734171)

[III. Điều kiện thực hiện thuật toán 2](#_Toc109734172)

[IV. Điều kiện dừng của thuật toán 2](#_Toc109734173)

[1. Công thức tiên nghiệm 2](#_Toc109734174)

[2. Công thức hậu nghiệm 3](#_Toc109734175)

[V. Thuật toán 3](#_Toc109734176)

[1. Thuật toán theo công thức hậu nghiệm 3](#_Toc109734177)

[a. Bằng chữ 3](#_Toc109734178)

[b, Bằng sơ đồ khối 4](#_Toc109734179)

[2. thuật toán theo công thức tiên nghiệm 4](#_Toc109734180)

[VI. Đánh giá thuật toán 5](#_Toc109734181)

[1. Ưu điểm 5](#_Toc109734182)

[2. Nhược điểm 5](#_Toc109734183)

[VII. Phương pháp biến đổi jacobi 6](#_Toc109734184)

[1. Ý tưởng phương pháp 6](#_Toc109734185)

[2. Điều kiên 6](#_Toc109734186)

[3. Ma trận chéo trội hàng 6](#_Toc109734187)

[4. Ma trận chéo trội cột 6](#_Toc109734188)

[5. Thiết lập điều kiện dừng 7](#_Toc109734189)

[6. Thuật toán 8](#_Toc109734190)

[a. kiểm tra chéo trội hàng 8](#_Toc109734191)

[b. Kiểm tra chéo trội cột 9](#_Toc109734192)

[c. Thuật toán tổng quát 9](#_Toc109734193)

[6. Đánh giá thuật toán 11](#_Toc109734194)

[a. Ưu điểm 11](#_Toc109734195)

[b. Nhược điểm 11](#_Toc109734196)

# I. Ý tưởng phương pháp

- Đưa về phương trình tương đương

Ax = B <=> x = Cx + D

- Lập dãy số

Xn = CXn - 1 + D, x0 |Rm

- Nếu dãy số trên hội tụ thì giới hạn của dãy số chính là nghiệm của hệ phương trình

# II. Chuẩn ma trận

- Chuẩn theo hàng:

- Chuẩn theo cột:

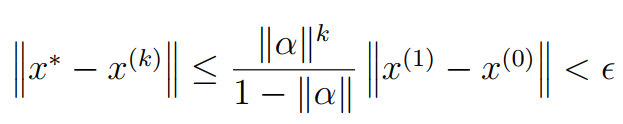
- Chuẩn euclid:

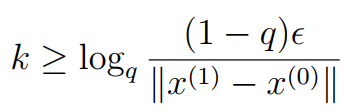
- Chuẩn theo trị triêng:

# III. Điều kiện thực hiện thuật toán

# IV. Điều kiện dừng của thuật toán

## 1. Công thức tiên nghiệm





Với k là số lần lặp cần thiết để cho ra nghiệm đúng sai số

## 2. Công thức hậu nghiệm

A picture containing text, antenna

Description automatically generated

# V. Thuật toán

## 1. Thuật toán theo công thức hậu nghiệm

### a. Bằng chữ

Input: eps, A, B

Output: x, k

Bước 1: nhập input

Bước 2: gán a = I + A, b = -B

Bước 3: nếu q = ||a||p < 1 thì chuyển sang bước 4, ngược lại thì chuyển sang bước 7

Bước 4: tính x = a.x0 + b và tăng k++

Trong đó x0 là điểm ban đầu của hệ phương trình

Bước 5: nếu thì chuyển sang bước 6, ngược lại thì chuyển sang bước 9

Bước 6: gán x0 = x rồi quay lại bước 4

Bước 7: gán a = I - A, b = B

Bước 8: nếu q = ||a||p < 1 thì quay lại bước 4, ngược lại thì kết luận không thỏa mãn điều kiện

Bước 9: in ra output

### b, Bằng sơ đồ khối

Diagram

Description automatically generated

## 2. thuật toán theo công thức tiên nghiệm

Input: eps, A, B, x0

Output: x, k

Bước 1: nhập input

Bước 2: gán a = I + A, b = -B

Bước 3: nếu q = ||a||p < 1 thì chuyển sang bước 4, ngược lại thì chuyển sang bước 10

Bước 4: tính x = a.x0 + b

Bước 5: tính k = ceil()

Bước 6: khởi tạo i = 2 để tính số lần lặp

Bước 7: nếu i <= n + 1 thì chuyển sang bước 8, ngược lại thì chuyển sang bước 12

Bước 8: gán x0 = x và tính x = a.x0 + b

Bước 9: tăng i++ và quay lại bước 7

Bước 10: gán a = I - A, b = B

Bước 11: nếu q = ||a||p < 1 thì quay lại bước 4, ngược lại thì kết luận không thỏa mãn điều kiện

Bước 12: in ra output

# VI. Đánh giá thuật toán

1. Ưu điểm

* Phương pháp lặp đơn giải quyết được sự bất ổn định của nghiệm khi giải hệ bằng phương pháp đúng (Gauss, Gauss-Jordan, Choleski)
* Tối ưu được bộ nhớ
* Dễ cài đặt trên máy tính
* Chi phí xử lí và tốc độ hội tụ nhanh hơn phương pháp tính toán trực tiếp

1. Nhược điểm

* Phương pháp lặp đơn: chỉ lặp được khi ||I – A|| < 1 hoặc

||I + A|| < 1

* Phương pháp lặp Jacobi: chỉ lặp được khi ma trận A là chéo trội hàng hoặc chéo trội cột
* Lớp phương trình đại số tuyến tính giải được bằng phương pháp lặp đơn và lặp Jacobi là tương đối hẹp

\*Chú ý: Tốc độ hội tụ của phương pháp lặp đơn và lặp Jacobi thì chậm hơn nhiều so với phương pháp lặp Seidel và Gauss Seidel

\* Phương pháp lặp Jacobi chỉ giải được với điều kiện ma trận A là ma trận vuông

# VII. Phương pháp biến đổi jacobi

## 1. Ý tưởng phương pháp

Tương tự như phương pháp lặp đơn cho hệ phương trình với A mà ma trận chéo trội

## 2. Điều kiên

Ma trận là chéo trội hàng hoặc chéo trội cột

## 3. Ma trận chéo trội hàng

- Là ma trận có dạng:

A picture containing watch

Description automatically generated

- Cho hệ phương trình Ax = B có ma trận mở rộng như sau

Chart, scatter chart

Description automatically generated with medium confidence

Ta đưa về hệ phương trình A’x = d có ma trận hệ số mở rộng như sau:

Chart, scatter chart

Description automatically generated

Ta đưa về ma trận x = Bx + d với B = I – A’ thì ta luôn có

Diagram

Description automatically generated with medium confidence

## 4. Ma trận chéo trội cột

- là ma trận có dạng

A picture containing text, watch, clock

Description automatically generated

- Cho hệ phương trình Ax = B có ma trận mở rộng như sau

Chart, scatter chart

Description automatically generated with medium confidence

- Đặt yi = aiixi và đưa về hệ phương trình y = B1y + b có ma trận bổ sung

Chart, scatter chart

Description automatically generated

- Dễ gàng nhận thấy

A picture containing diagram

Description automatically generated

- Biến đổi về vòng lặp với x

Diagram

Description automatically generated with medium confidence

## 5. Thiết lập điều kiện dừng

Đặt , q = ||B1||1

a. Công thức sai số hậu nghiệm

A picture containing clock, antenna

Description automatically generated

b. Công thức sai số tiên nghiệm

A picture containing object, antenna, gauge

Description automatically generated

## 6. Thuật toán

### a. kiểm tra chéo trội hàng

input: A

output: true hoặc false

Bước 1: nhập input

Bước 2: khởi tạo biến i = 1 để duyệt từng hàng

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

Bước 4: gán max = Aii

Bước 5: khởi tạo biến j = 1 để duyệt từng cột

Bước 6: nếu j <= n thì chuyển sang bước 7, ngược lại thì chuyển sang bước 10

Bước 7: nếu j i thì tính max = max – |Aij|, ngược lại thì chuyển sang bước 9

Bước 8: nếu max <= 0 thì trả về giá trị là false và kết thúc, ngược lại thì chuyển sang bước 9

Bước 9: tăng j++ rồi quay lại bước 6

Bước 10: tăng i++ rồi quay lại bước 3

Bước 11 trả về giá trị là true và kết thúc

### b. Kiểm tra chéo trội cột

input: A

output: true hoặc false

Bước 1: nhập input

Bước 2: khởi tạo biến i = 1 để duyệt từng cột

Bước 3: Nếu i <= n thì chuyển sang bước 4, ngược lại thì chuyển sang bước 11

Bước 4: gán max = Aii

Bước 5: khởi tạo biến j = 1 để duyệt từng hàng

Bước 6: nếu j <= n thì chuyển sang bước 7, ngược lại thì chuyển sang bước 10

Bước 7: nếu j i thì tính max = max – |Aji|, ngược lại thì chuyển sang bước 9

Bước 8: nếu max <= 0 thì trả về giá trị là false và kết thúc, ngược lại thì chuyển sang bước 9

Bước 9: tăng j++ rồi quay lại bước 6

Bước 10: tăng i++ rồi quay lại bước 3

Bước 11 trả về giá trị là true và kết thúc

### c. Thuật toán tổng quát

- Bằng sơ đồ khối:

Diagram

Description automatically generated

- Bằng chữ:

Input:

Output:

Bước 1: Nhập input

Bước 2: Khởi tạo

Bước 3: Kiểm tra A chéo trội hàng không. Nếu có thì chuyển sang bước 4, nếu sai thì chuyển sang bước 5

Bước 4: khởi tạo , ,

Bước 5: Kiểm tra B chéo trội cột không. Nếu có thì chuyển sang bước 6, nếu không thì thông báo ma trận không chéo trội và kết thúc thuật toán.

Bước 6: khởi tạo , ,

Bước 7: Khởi tạo

Bước 8: Thực hiện phép toán và tăng

Bước 9: Kiểm tra . Nếu đúng thì chuyển sang bước 10, nếu sai thì gán quay lại bước 8.

Bước 10: in ra output

## 6. Đánh giá thuật toán

### a. Ưu điểm

* Phương pháp lặp đơn giải quyết được sự bất ổn định của nghiệm khi giải hệ bằng phương pháp đúng (Gauss, Gauss-Jordan, Choleski)
* Tối ưu được bộ nhớ
* Dễ cài đặt trên máy tính
* Chi phí xử lí và tốc độ hội tụ nhanh hơn phương pháp tính toán trực tiếp

### b. Nhược điểm

* Phương pháp lặp đơn: chỉ lặp được khi ||I – A|| < 1 hoặc

||I + A|| < 1

* Phương pháp lặp Jacobi: chỉ lặp được khi ma trận A là chéo trội hàng hoặc chéo trội cột
* Lớp phương trình đại số tuyến tính giải được bằng phương pháp lặp đơn và lặp Jacobi là tương đối hẹp

\*Chú ý: Tốc độ hội tụ của phương pháp lặp đơn và lặp Jacobi thì chậm hơn nhiều so với phương pháp lặp Seidel và Gauss Seidel

\* Phương pháp lặp Jacobi chỉ giải được với điều kiện ma trận A là ma trận vuông