**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC**

**----o0o----**

**A close up of a sign

Description automatically generated**

**Báo cáo Giải tích số**

**Phương pháp lặp Seidel và Gauss Seidel**

Thực Hiện: Vũ Minh Hiếu

MSSV: 20195872

Vũ Hoài Nam

MSSV: 20190059

Hà Nội - 2020

**Mục lục**

**1. Phương pháp lặp Seidel**

1.1 Lý thuyết 3

1.2 Công thức 4

1.3 Điều kiện hội tụ và sai số 5

**2. Phương pháp lặp Gauss Seidel**

2.1 Chéo trội hàng 9

2.2 Chéo trội cột ..11

**3. Đánh giá phương pháp**

**4 .Chương trình**

1. **Phương pháp lặp Seidel**
   1. Lý thuyết

Xét phương trình dạng

X = αX +β

Ta phân tích ma trận α thành 2 ma trận tam giác dưới đúng và ma trận tam giác trên: α =

Với:

=

và =

Từ đó ta có

X = αX +β=

= X +β

1.2 Công thức

Chọn xấp xỉ đầu bất kỳ.

Tính theo công thức lặp:

= +β

k=1,2,3,…

Cụ thể:

Hay i=1,2,…n

Tổng quát cho phép lặp thứ k:

1.3 Điều kiện hội tụ và sai số.

Nếu quá trình lặp hội tụ: = thì = là nghiệm đúng của hệ

Chứng minh:

Do khi

nên = +β

= α+β vậy =

1. 3.1Điều kiện.

Nếu một chuẩn nào đó của ma trận α thỏa mãn điều kiện:

thì phương pháp lặp Seidel hội tụ tới nghiệm đúng của hệ phươngtrình với xấp xỉ đầu bất kỳ.

1. 3.2 Công thức đánh giá sai số

a) Nếu

Trong đó λ=

Hoặc

b) Nếu

Trong đó:

S= ; =

Chứng minh:

Ta có:

k=1,2,…

Nếu là nghiệm đúng của phương trình thì:

Từ đó suy ra

=>

Đặt

Mà ∀i

nên + ∀i

=> +

hay

Đặt λ==

ta được

Ta sẽ chứng minh λ

Thật vậy, ta có

Ta có:

=> λ=

Mặt khác

=>

Hay =

Chứng minh các công thức đánh giá sai số:

Công thức sai số thứ nhất:

Tương tự như trên, ta có:

Xét p nguyên dương bất kỳ

+…+

=

Cố định k, cho p thì

=>

Để chứng minh công thức sai số thứ 2, ta thay k bằng k+p-1

Truy hồi :

Lại thay k=1, ta được:

Thay p=k-1

=>

**2. Phương pháp lặp Gauss Seidel**

2.1 Đối với ma trận chéo trội hàng

Ma trận A = được gọi là ma trận chéo trội hàng nếu nó thỏa mãn:

, ∀i

Khi đó với phương trình có dạng

AX=B

Hay

=

Nhân 2 vế với ma trận T=diag( i= vào bên trái, ta được

=

⇔ =

⇔ +

Khi đó, ta đã đưa phương trình ban đầu về dạng

X=A`X+B`

Dễ thấy < 1 do A là ma trận chéo trội hàng

Từ đó ta có thể áp dụng công thức lặp Seidel để giải phương trình sau khi biến đổi

2.2 Đối với ma trận chéo trội cột

Ma trận A = được gọi là ma trận chéo trội cột nếu nó thỏa mãn:

, ∀i

Khi đó với phương trình có dạng

AX=B

Hay

=

Xét

Đặt Y==

=> X=TY=

Thay X=TY vào phương trình ban đầu ta được

=

⇔ =

⇔ +

Khi đó ta đã đưa phương trình ban đầu về dạng

Y=A”Y+B

Dễ thấy < 1 do A là ma trận chéo trội cột.

Lại có

Y=A”Y+B

⇔ TY=TA”Y+TB

⇔ X=T(I-AT)

⇔ X=(I-TA)X+TB

⇔ X=A’X+B’

Vậy ma trận lặp của phương pháp vẫn giống với trường hợp chéo trội hàng, nhưng khi lặp cần thêm hệ số lặp là ζ và S theo công thức

Trong đó:

S= ; =

Từ đó ta có thể áp dụng công thức lặp Seidel để giải phương trình sau khi biến đổi.

**3. Đánh giá**

Ta nhận thấy rằng phương pháp lặp Gauss-Seidel là một phiên bản cải tiến của phương pháp Jacobi và Seidel là phiên bản cải tiến của phương pháp lặp đơn về tốc độ hội tụ (số vòng lặp). Do đó nó vẫn giữ lại các ưu điểm của phương pháp cũ như:

+) Giá trị lặp ban đầu là tùy ý.

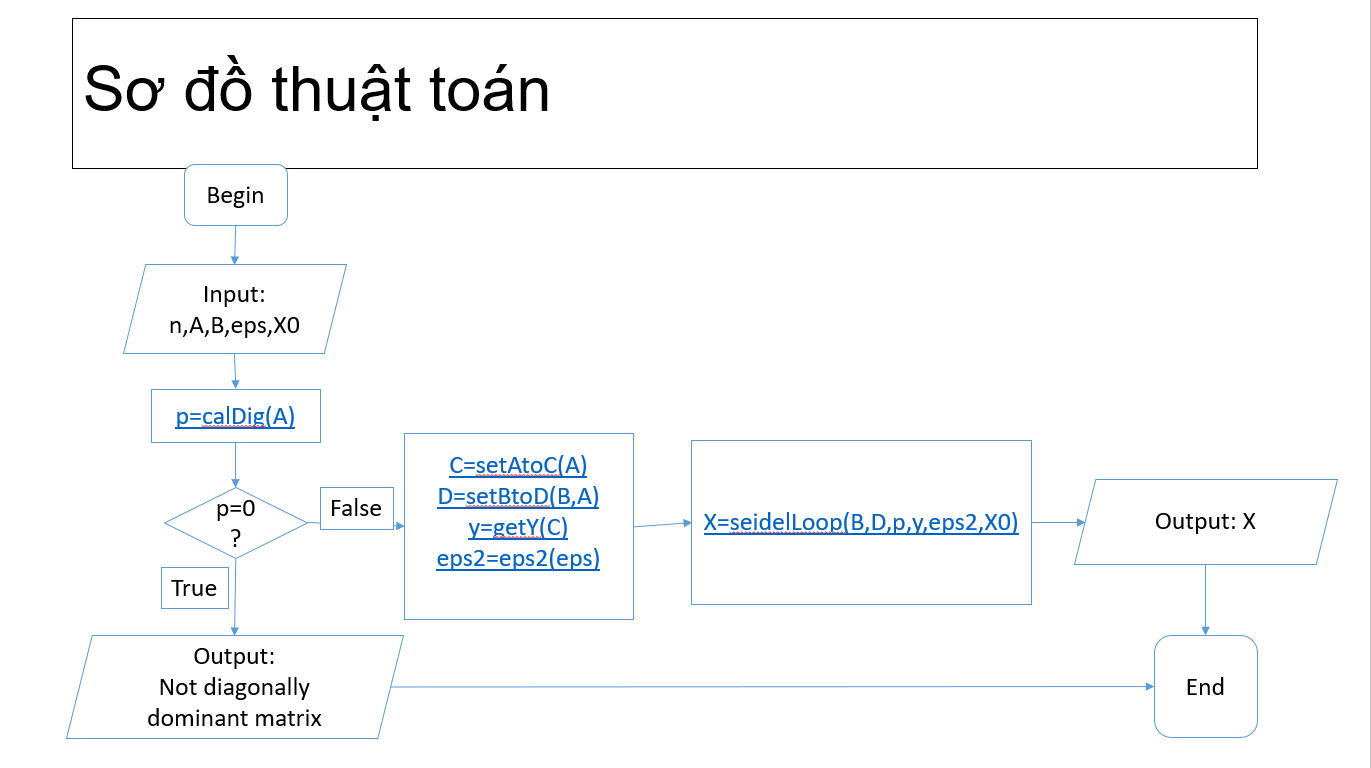
+) Kiểm soát được sai số của nghiệm.

và cả nhược điểm như:

+) Phải là ma trận chéo trội với phương pháp (Gauss-Seidel).

**4. Chương trình**

4.1 Thuật toán



3.1.1. Thuật toán các gói nhỏ:

**calDig**

Input : A ( ma trận A)

Output : p {0,1,2} ( tương ứng với ma trận A không chéo trội, chéo trội hàng hoặc chéo trội cột)

B1: If ( > with i= )

{p=1; Skip to B4}

B2: If ( > with j= )

{p=2; Skip to B4}

B3: p=0

B4: Return p

**setAtoC**

Input : A ( ma trận A)

Output : C (ma trận C trong pt x=Cx+D biến đổi từ pt Ax=B)

for i=1 to n:

for j=1 to n:

if ( i );

else

Return C

**setBtoD**

Input : B,A. ( vector B, ma trận A)

Output : D ( vector D trong pt x=Cx+D biến đổi từ pt Ax=B)

for i=1 to n:

;

Return C

**getNorm**

Input : D,p (D là vector, p là loại chuẩn )

Output : ( Chuẩn p của vector D)

B1:If (p=1) skip to B2;Else skip to B3

B2: = Max with i=

B3: =

B4: Return

**getNorm**

Input : C,p (C là ma trận,p là loại chuẩn )

Output : ( Chuẩn p của ma trận C)

B1:If (p=1) skip to B2;Else skip to B3

B2: =

B3: =

B4: Return

**getY**

Input :C (ma trận C)

Output : y ( y là hệ số co)

y=0

For i=1 to n

y=

Return y

**SeidelLoop**

Input :B,D,p,y,eps2,X0 (Trong đó B với D là ma trận và vector trong pt x=Bx+D, p là loại chuẩn, y là hệ số sai số, eps2 là sai số phương pháp, X0 là xấp xỉ đầu,)

Output : X ( X là vector kết quả có sai số với vector kết quả chuẩn sai số eps)

Lấy vector X=X0

X1

Do:

X1=X;

+ + with i= ;

while (y/(1-y)\*getNorm(X-Xi,p) >eps2)

Return X

**eps2**

Input : eps (eps là sai số đầu vào)

Output : eps2 ( eps2 là sai số phương pháp)

eps1=\*0.5;

eps2=eps-eps1;

Return eps2

.2.Hệ thống ví dụ

VD1: Ta sẽ giải hệ phương trình sau với sai số 0.02

A= B=

Input:Ta dễ ràng thấy rằng đây là ma trận chéo trội hàng nên có thể áp dụng phương pháp Gauss-Seidel ( ) để giải.

Ta chọn giá trị lặp ban đầu hoàn toàn bất kì X0=.

Sau khi ta chạy chương trình sẽ tìm ra được nghiệm cần tìm và giá trị của A\*X là:

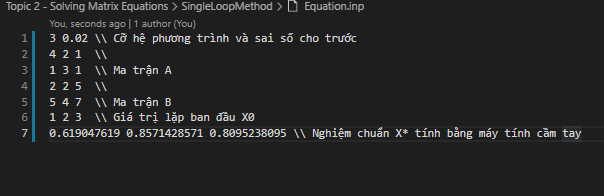
X= A\*X=

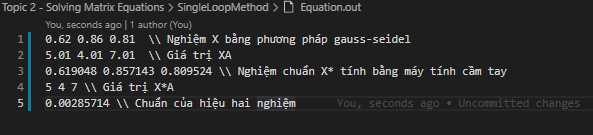
Sau đó ta thử tính giá trị của trong phương trình trên bằng máy tính cầm tay sẽ ra kết quả:

= A\* =

Vậy ta có thể coi giá trị X tính được ở máy tính cầm tay là nghiệm chuẩn. Ta có so sánh sự chênh lệch giữa nghiệm tìm được của phương pháp gauss-seidel và nghiệm chuẩn: < 0.02.

Kết luận: Vậy ta thấy rằng chương trình có thể chạy ra nghiệm đúng với sai số cho trước.





Chú thích: phía trên là hình ảnh về file input và output của ví dụ trên ( những thông tin sau \\ là chú thích được người viết thêm vào nhằm giúp người đọc dễ hiểu các giá trị đầu vào cũng như đầu ra hoàn toàn không có trong chương trình).

VD 2: Ta tiếp tục giải phương trình ở ví dụ trên tuy nhiên lần này ta sẽ chọn các giá trị lặp ban đầu khác nhau để xem chúng có đều hội tụ tới nghiệm của phương pháp không?

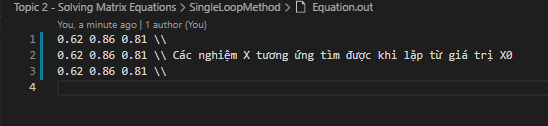
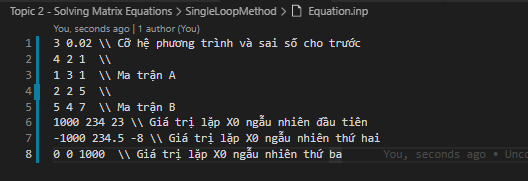
Ta lần lượt chọn các giá trị và tính nghiệm X của phương pháp khi sử dụng giá trị đó.:

= X =

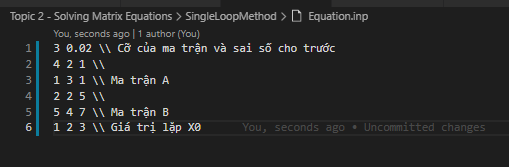
= X =

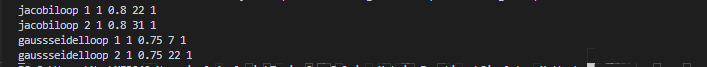
= X =

Kết luận: Ta thấy rằng phương pháp sẽ hội tụ tới nghiệm thỏa mãn sai số cho trước không phụ thuộc vào giá trị lặp ban đầu là bao nhiêu.



Ví dụ 3: Tiếp tục với ví dụ trên tuy nhiên lần này ta sẽ so sánh tốc độ hội tụ của công thức lặp tiên nghiệm và hậu nghiệm của phương pháp Gauss-Seidel cũng như của phương pháp Jacobi:





Chú thích : Các giá trị ở trên stder ( luồng báo lỗi ) lần lượt là : tên phương pháp, công thức hội tụ (1 ứng với hậu nghiệm và 2 ứng với tiên nghiệm), loại chuẩn sử dụng, hệ số co , số lần lặp để tới được nghiệm của phương pháp và giá trị của hệ số w trong công thức sai số.

Nhận xét : Ta có thể nhận ra rằng phương pháp lặp Gauss-Seidel có tốc độ hội tụ tối ưu hơn hẳn so với phương pháp Jacobi. Và việc sử dụng công thức sai số hậu nghiệm cũng sẽ đem lại tốc độ lặp tốt hơn với công thức sai số tiên nghiệm ( Một phần lí do là phép tính số lần lặp của công thức sai số tiên nghiệm dùng các phép tính loga, làm tròn lên để chắc chắn đủ số bước để lặp tới nghiệm yêu cầu nên nhiều khi nghiệm đã lặp tới sai số cần thiết nhưng nó vẫn tiến tiếp) (Ta có thể tối ưu bằng cách chỉ lặp khi giá trị khác nhau tuy nhiên ở ví dụ này nhóm em muốn chỉ ra sự khác biệt khi sử dụng công thức sai số hậu nghiệm và tiên nghiệm nên không cải tiến).