Phương pháp lặp Gauss-Seidel

Input:

- Ma trận A (kích thước $n \times n$). (Điều kiện: A chéo trội hàng hoặc cột, hoặc A đối xứng xác định dương để đảm bảo hội tụ).
- Vector b (kích thước $n \times 1$).
- Số lần lặp k.
- Vector xấp xỉ ban đầu $X^{(0)}$ (kích thước $n \times 1$).

Output:

- Nghiệm gần đúng $X^{(k)}$ của hệ $AX = b$.
- Sai số ε của nghiệm vừa tìm được.

Các bước:

B1. Khởi tạo:

- $t = 0$.
- $X^{(t)} = X^{(0)}$.

B2. Lặp: WHILE $t < k$ **DO**

(a) $X_{old} = X^{(t)}$ (Lưu lại nghiệm của bước lặp trước cho việc tính sai số)

(b) **FOR** $i = 1$ **TO** n **DO**

$$x_i^{(t+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(t+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(t)} \right)$$

(Nếu $i = 1$, tổng thứ nhất bằng 0. Nếu $i = n$, tổng thứ hai bằng 0.)

(c) $X^{(t+1)}$ là vector với các thành phần $x_i^{(t+1)}$ vừa tính.

(d) $t = t + 1$.

(e) $X^{(t)} = X^{(t+1)}$ (chuẩn bị cho vòng lặp tiếp theo, nhưng để rõ ràng, ta hiểu $X^{(t+1)}$ ở bước trên là nghiệm mới)

END WHILE

(Sau vòng lặp, $X^{(k)}$ là nghiệm cuối cùng, và X_{old} sẽ là $X^{(k-1)}$)

B3. Tính sai số hậu nghiệm ε (chọn một trong các cách sau):

- **Cách 1: Dựa trên hiệu của hai lần lặp cuối (phổ biến):**

$$\varepsilon = \|X^{(k)} - X^{(k-1)}\|_\infty$$

  Hoặc

$$\varepsilon = \|X^{(k)} - X^{(k-1)}\|_1$$

- **Cách 2: Dựa trên vector phần dư:**

$$\varepsilon = \|b - AX^{(k)}\|_\infty$$

- **Cách 3: (Nâng cao, nếu muốn công thức tương tự Jacobi, tính toán nhiều hơn)**

  (a) Tính ma trận lặp của Gauss-Seidel: $B_{GS} = -(D+L)^{-1}U$.

  (b) Tính chuẩn của B_{GS}: $\|B_{GS}\|_1$.

  (c) Nếu $\|B_{GS}\|_1 < 1$:

$$\varepsilon = \frac{\|B_{GS}\|_1}{1 - \|B_{GS}\|_1}\|X^{(k)} - X^{(k-1)}\|_1$$

  (d) Ngược lại (nếu $\|B_{GS}\|_1 \geq 1$): Công thức này không áp dụng, sử dụng Cách 1 hoặc Cách 2.

Phương Pháp Lặp Đơn (Simple Iteration / Successive Substitution)

Phương pháp này dùng để giải hệ phương trình tuyến tính có dạng AX = B.

a) Input (Đầu vào)

- Ma trận α (kích thước $n \times n$).
- Vector β (kích thước $n \times 1$).
- Số lần lặp k (số lần lặp cố định hoặc số lần lặp tối đa).

- Vector xấp xỉ ban đầu $X^{(0)}$ (kích thước $n \times 1$).
- (Tùy chọn) Ngưỡng sai số ϵ_{stop} cho điều kiện dừng sớm (nếu không muốn chạy đủ k lần lặp).

b) Output (Đầu ra)

- Nghiệm gần đúng $X^{(k_{\text{final}})}$ của hệ AX = B.
- Sai số ước lượng ε của nghiệm vừa tìm được.

c) Các bước thực hiện

Bước 1: Kiểm tra điều kiện hội tụ tiên nghiệm (Tùy chọn nhưng hữu ích)

- Chọn một chuẩn ma trận phù hợp (ví dụ: chuẩn 1, chuẩn vô cùng, chuẩn Frobenius).
- Tính $q = \|\alpha\|$.
- **NẾU** $q \geq 1$ **THÌ**
  - In cảnh báo: "Chuẩn của ma trận lặp α là $q \geq 1$. Phương pháp có thể không hội tụ."Kết thúc thuật toán

Bước 2: Khởi tạo các giá trị ban đầu

- Đặt biến đếm số lần lặp: $t = 0$.
- Gán nghiệm hiện tại: $X^{(t)} = X^{(0)}$.

Bước 3: Thực hiện vòng lặp LẶP TRONG KHI $t < k$ (**VÀ** điều kiện dừng sớm dựa trên ϵ_{stop} chưa được thỏa mãn, nếu có)

(a) Lưu lại nghiệm của bước lặp trước: $X_{\text{old}} = X^{(t)}$.

(b) Tính toán nghiệm mới dựa trên công thức lặp:

$$X^{(t+1)} = \alpha X^{(t)} + \beta$$

(c) Tăng biến đếm số lần lặp: $t = t + 1$.

(d) Cập nhật nghiệm hiện tại: $X^{(t)} = X^{(t+1)}$.

(e) (*Tùy chọn: Kiểm tra điều kiện dừng sớm*)

- Tính sai số hiện tại, ví dụ: current_error $= \|X^{(t)} - X_{\text{old}}\|_\infty$.

- **NẾU** current_error $< \epsilon_{\text{stop}}$ **THÌ**
  - Thoát khỏi vòng lặp (BREAK).
- **KẾT THÚC**

KẾT THÚC LẶP
(Sau khi vòng lặp kết thúc, nghiệm cuối cùng là $X^{(t)}$, và X_{old} là nghiệm $X^{(t-1)}$. Đặt $k_{\text{final}} = t$.)

Bước 4: Tính sai số hậu nghiệm ε (Chọn một trong các phương pháp sau)

- **Cách 1: Sử dụng công thức ước lượng sai số (nếu $q = \|\alpha\| < 1$ ở Bước 1)**

$$\varepsilon = \frac{q}{1-q}\|X^{(k_{\text{final}})} - X^{(k_{\text{final}}-1)}\|$$

  (Sử dụng cùng một chuẩn $\|\cdot\|$ đã dùng để tính q cho ma trận α và cho hiệu vector.)

- **Cách 2: Dựa trên hiệu của hai lần lặp cuối (phổ biến và đơn giản)**

$$\varepsilon = \|X^{(k_{\text{final}})} - X^{(k_{\text{final}}-1)}\|_\infty$$

  Hoặc một chuẩn vector khác như $\|\cdot\|_1$ hoặc $\|\cdot\|_2$.

- **Cách 3: Dựa trên vector phần dư của phương trình lặp**

$$\varepsilon = \|X^{(k_{\text{final}})} - (\alpha X^{(k_{\text{final}})} + \beta)\|_\infty$$

  (Đo lường mức độ nghiệm cuối cùng thỏa mãn phương trình $X = \alpha X + \beta$.)

- **Cách 4: Dựa trên vector phần dư của hệ gốc (nếu biết)** Nếu hệ $X = \alpha X + \beta$ được biến đổi từ một hệ gốc $AX_{gc} = b_{gc}$, thì có thể tính:

$$\varepsilon = \|b_{gc} - AX_{gc}^{(k_{\text{final}})}\|_\infty$$

  (Cần đảm bảo $X^{(k_{\text{final}})}$ ở đây tương ứng với nghiệm của hệ gốc).

Phương Pháp Lặp Jacobi

Phương pháp này dùng để giải hệ phương trình tuyến tính $AX = b$.

a) Input (Đầu vào)

- Ma trận A (kích thước $n \times n$). (Điều kiện: A chéo trội hàng hoặc cột để đảm bảo hội tụ).
- Vector b (kích thước $n \times 1$).
- Số lần lặp k (số lần lặp cố định hoặc số lần lặp tối đa).
- Vector xấp xỉ ban đầu $X^{(0)}$ (kích thước $n \times 1$).
- (Tùy chọn) Ngưỡng sai số ϵ_{stop} cho điều kiện dừng sớm.

b) Output (Đầu ra)

- Nghiệm gần đúng $X^{(k_{\text{final}})}$ của hệ $AX = b$ (với $k_{\text{final}} \leq k$).
- Sai số ước lượng ε của nghiệm vừa tìm được.

c) Các bước thực hiện

Bước 1: Tách ma trận A (để xác định ma trận lặp)

- D: Ma trận đường chéo của A (chứa các phần tử a_{ii}).
- L: Ma trận tam giác dưới chặt của A (các phần tử a_{ij} với $i > j$, còn lại là 0).
- U: Ma trận tam giác trên chặt của A (các phần tử a_{ij} với $i < j$, còn lại là 0).
- (Lưu ý: $A = D + L + U$).

Bước 2: Tính ma trận lặp B_J và vector hằng số d_J

- Kiểm tra xem tất cả $a_{ii} \neq 0$. Nếu có $a_{ii} = 0$, phương pháp không áp dụng được trực tiếp.
- D^{-1}: Ma trận đường chéo với các phần tử $1/a_{ii}$.
- Ma trận lặp Jacobi: $B_J = -D^{-1}(L + U)$.
- Vector hằng số Jacobi: $d_J = D^{-1}b$.
- (Công thức lặp sẽ là $X^{(t+1)} = B_J X^{(t)} + d_J$).

Kiểm tra điều kiện hội tụ tiên nghiệm (Tùy chọn nhưng hữu ích):

- Tính một chuẩn nào đó của ma trận B_J, ví dụ: $q_J = \|B_J\|_1$ hoặc $q_J = \|B_J\|_\infty$, hoặc bán kính phổ $\rho(B_J)$.

- **NẾU** $q_J \geq 1$ (hoặc $\rho(B_J) \geq 1$) **THÌ**
  - Cảnh báo: "Điều kiện hội tụ tiên nghiệm có thể không được thỏa mãn. Phương pháp Jacobi có thể không hội tụ."Kết thúc thuật toán

Bước 3: Khởi tạo các giá trị ban đầu và hệ số λ (nếu dùng công thức sai số cụ thể)

- Đặt biến đếm số lần lặp: $t = 0$.
- Gán nghiệm hiện tại: $X^{(t)} = X^{(0)}$.
- Tính hệ số λ (như trong mô tả của bạn, nếu áp dụng công thức sai số đó):

$$\lambda = \frac{\max_i |a_{ii}|}{\min_i |a_{ii}|}$$

(Yêu cầu $\min_i |a_{ii}| \neq 0$).

Bước 4: Thực hiện vòng lặp LẶP TRONG KHI $t < k$ (VÀ điều kiện dừng sớm dựa trên ϵ_{stop} chưa được thỏa mãn, nếu có)

(a) Lưu lại nghiệm của bước lặp trước: $X_{\text{old}} = X^{(t)}$.

(b) Tính toán nghiệm mới $X^{(t+1)}$:

- **Cách 1: Sử dụng ma trận lặp B_J và d_J:**

$$X^{(t+1)} = B_J X^{(t)} + d_J$$

- **Cách 2: Tính theo từng thành phần (phổ biến hơn trong triển khai): FOR $i = 1$ TO n DO**

$$x_i^{(t+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1, j \neq i}^{n} a_{ij} x_j^{(t)} \right)$$

  KẾT THÚC FOR
  ($X^{(t+1)}$ là vector với các thành phần $x_i^{(t+1)}$ vừa tính).

(c) Tăng biến đếm số lần lặp: $t = t + 1$.

(d) Cập nhật nghiệm hiện tại: $X^{(t)} = X^{(t+1)}$.

(e) (*Tùy chọn: Kiểm tra điều kiện dừng sớm*)

- Tính sai số hiện tại, ví dụ: current_error $= \|X^{(t)} - X_{\text{old}}\|_\infty$.
- **NẾU** current_error $< \epsilon_{\text{stop}}$ **THÌ**

– Thoát khỏi vòng lặp (BREAK).

- **KẾT THÚC NẾU**

KẾT THÚC LẶP
(Sau khi vòng lặp kết thúc, nghiệm cuối cùng là $X^{(t)}$, và X_{old} là nghiệm $X^{(t-1)}$. Đặt $k_{\text{final}} = t$.)

Bước 5: Tính sai số hậu nghiệm ε (Chọn một trong các phương pháp sau)

- **Cách 1: Sử dụng công thức từ mô tả của bạn (nếu** $q_J = \|B_J\|_1 < 1$**)**

$$\varepsilon = \lambda \cdot \frac{\|B_J\|_1}{1 - \|B_J\|_1} \cdot \|X^{(k_{\text{final}})} - X^{(k_{\text{final}}-1)}\|_1$$

- **Cách 2: Dựa trên hiệu của hai lần lặp cuối (phổ biến và đơn giản)**

$$\varepsilon = \|X^{(k_{\text{final}})} - X^{(k_{\text{final}}-1)}\|_\infty$$

  Hoặc một chuẩn vector khác như $\|\cdot\|_1$ hoặc $\|\cdot\|_2$.

- **Cách 3: Dựa trên vector phần dư của hệ gốc**

$$\varepsilon = \|b - AX^{(k_{\text{final}})}\|_\infty$$

Phân Tích LU và Giải Hệ AX = B

Quy trình này bao gồm ba phần chính:

1. Phân tích ma trận A thành L (tam giác dưới) và U (tam giác trên với đường chéo chính bằng 1), sao cho $A = LU$.
2. Giải hệ phương trình tam giác dưới $LY = B$ để tìm Y.
3. Giải hệ phương trình tam giác trên $UX = Y$ để tìm X.

I. Phân Tích LU

a) Input (Đầu vào)

- Ma trận A (kích thước $n \times n$).

b) Output (Đầu ra)

- Ma trận tam giác dưới L (kích thước $n \times n$).
- Ma trận tam giác trên U (kích thước $n \times n$) với các phần tử trên đường chéo chính $u_{ii} = 1$.

c) Các bước thực hiện

Bước 1: Khởi tạo ma trận L và U

- Tạo ma trận L kích thước $n \times n$ chứa toàn số 0.
- Tạo ma trận U kích thước $n \times n$ chứa toàn số 0.
- Đặt tất cả các phần tử trên đường chéo chính của U bằng 1: $u_{ii} = 1$ với $i = 1, \ldots, n$.

Bước 2: Tính toán các phần tử của L và U FOR $j = 0$ TO $n - 1$ DO (Duyệt qua các cột, chỉ số 0-based)

2a. **Tính các phần tử của cột j trong ma trận L:**
FOR $i = j$ **TO** $n - 1$ **DO** ($i \geq j$)

$$l_{ij} = a_{ij} - \sum_{k=0}^{j-1} l_{ik} u_{kj}$$

(Nếu $j = 0$, tổng bằng 0.) **KẾT THÚC FOR** (i)

2b. **Kiểm tra điều kiện chia cho l_{jj}:**
NẾU $l_{jj} = 0$ **THÌ**

- In thông báo lỗi: "Phần tử l_{jj} bằng 0. Phân tích LU (Crout) không thể thực hiện mà không có hoán vị."
- Dừng thuật toán.

2c. **Tính các phần tử của hàng j trong ma trận U (ngoại trừ $u_{jj} = 1$):**
FOR $p = j + 1$ **TO** $n - 1$ **DO** ($p > j$)

$$u_{jp} = \frac{1}{l_{jj}} \left(a_{jp} - \sum_{k=0}^{j-1} l_{jk} u_{kp} \right)$$

(Nếu $j = 0$, tổng bằng 0.) **KẾT THÚC FOR** (p)

KẾT THÚC FOR (j)

II. Giải Hệ Tam Giác Dưới $LY = B$ (Forward Substitution)

a) Input (Đầu vào)

- Ma trận tam giác dưới L (từ Phần I).
- Ma trận B (kích thước $n \times m$, $m \geq 1$).

b) Output (Đầu ra)

- Ma trận Y (kích thước $n \times m$).

c) Các bước thực hiện

Bước 1: Khởi tạo ma trận Y

- Tạo ma trận Y kích thước $n \times m$ chứa toàn số 0.

Bước 2: Tính toán các phần tử của Y FOR $p = 0$ TO $m - 1$ DO (Duyệt qua từng cột của B và Y)

2a. **FOR** $i = 0$ **TO** $n - 1$ **DO** (Duyệt qua các hàng)

$$y_{ip} = \frac{1}{l_{ii}} \left(b_{ip} - \sum_{k=0}^{i-1} l_{ik} y_{kp} \right)$$

(Nếu $i = 0$, tổng bằng 0. Yêu cầu $l_{ii} \neq 0$.) **KẾT THÚC FOR** (i)

KẾT THÚC FOR (p)

III. Giải Hệ Tam Giác Trên $UX = Y$ (Backward Substitution)

a) Input (Đầu vào)

- Ma trận tam giác trên U (từ Phần I, với $u_{ii} = 1$).
- Ma trận Y (từ Phần II).

b) Output (Đầu ra)

- Ma trận nghiệm X (kích thước $n \times m$).

c) Các bước thực hiện

Bước 1: Khởi tạo ma trận X

- Tạo ma trận X kích thước $n \times m$ chứa toàn số 0.

Bước 2: Tính toán các phần tử của X

FOR $p = 0$ **TO** $m - 1$ **DO** (Duyệt qua từng cột của Y và X)

2a. **FOR** $i = n - 1$ **DOWNTO** 0 **DO** (Duyệt ngược qua các hàng)

$$x_{ip} = \frac{1}{u_{ii}} \left(y_{ip} - \sum_{k=i+1}^{n-1} u_{ik} x_{kp} \right)$$

(Nếu $i = n-1$, tổng bằng 0. Vì $u_{ii} = 1$ cho Crout, mẫu số là 1.) **KẾT THÚC FOR** (i)

KẾT THÚC FOR (p)