ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC

CÁC PHƯƠNG PHÁP TÍNH GẦN ĐÚNG MA TRẬN NGHỊCH ĐẢO

Hà Minh Dũng - MSSV 20200096

GIẢNG VIÊN HƯỚNG DẪN
TS. Hà Thị Ngọc Yến

Hà Nội, tháng 1, 2022

Mục lục

1 Đặt vấn đề

1.1 Phát biểu bài toán

Bài toán. *Cho ma trận vuông A cấp n. Tìm ma trận nghịch đảo của A nếu có.*

Trong báo cáo này, ta sẽ nghiên cứu các phương pháp tìm gần đúng ma trận nghịch đảo của một ma trận thực vuông cấp n.

1.2 Lí do cần giải gần đúng ma trận nghịch đảo

Do khuyết điểm của các phương pháp tính đúng ma trận nghịch đảo:

- Máy tính bắt buộc phải thực hiện tính toán quá nhiều khi cần độ chính xác cao
- Không kiểm soát được sai số tính toán do giới hạn tính toán của máy tính

Nên ta đề xuất sử dụng các phương pháp lặp để làm tăng độ chính xác của các ma trận nghịch đảo có độ chính xác thấp và để kiểm soát sai số dễ dàng hơn.

1.3 Các phương pháp giải

Báo cáo này sẽ trình bày 3 phương pháp giải bao gồm phương pháp Newton có thể áp dụng với ma trận bất kì và hai phương pháp Jacobi và Gauss-Seidel áp dụng với ma trận chéo trội.

2 Phương pháp Newton

2.1 Ý tưởng

Xuất phát từ công thức lặp tính gần đúng nghịch đảo của số thực $a > 0$. Xét hàm $f(x) = a - \frac{1}{x}$ và áp dụng phương pháp lặp Newton để tìm nghiệm của nó ta có công thức lặp là:

$$x_{n+1} = x_n(2 - ax_n)$$

Một cách tương tự, ta thử áp dụng công thức này để tìm ma trận nghịch đảo X của ma trận A:

$$X_{k+1} = X_k(2E - AX_k) = X_k + X_k(E - AX_k)$$

Ta chứng minh dãy X_k hội tụ về ma trận nghịch đảo A^{-1} của A theo chuẩn 2 của ma trận. Kí hiệu $\|A\|$ là chuẩn 2 của ma trận A.
Thật vậy, ta đặt

$$G_k = E - AX_k = E - AX_{k-1}(2E - AX_{k-1}) = (E - AX_{k-1})^2 = G_{k-1}^2$$

Suy ra

$$G_k = G_{k-1}^2 = G_{k-2}^4 = ... = G_0^{2^k}$$

Mặt khác

$$A^{-1} - X_k = A^{-1}(E - AX_k) = A^{-1}G_k = A^{-1}G_0^{2^k}$$

Nên

$$\|A^{-1} - X_k\| \leq \|A^{-1}\| \|G_0\|^{2^k} \tag{1}$$

Như vậy, nếu ta chọn xấp xỉ ban đầu X_0 gần A^{-1}, sao cho

$$\|G_0\| = \|E - AX_0\| < 1$$

Thì $\|A^{-1} - X_k\| \to 0$ rất nhanh khi $k \to \infty$ hay dãy (X_k) hội tụ và

$$\lim_{k\to\infty} X_k = A^{-1}$$

Cụ thể, ta sẽ sử dụng $X_0 = \frac{A^T}{\|A\|^2}$.

2.2 Lí do chọn $X_0 = \frac{A^T}{\|A\|^2}$

Kí hiệu $\lambda(A)$ là giá trị lớn nhất của các trị tuyệt đối của các trị riêng $\lambda_i(A)$ của ma trận A. Khi đó, với i bất kì, ta có

$$AA^T x_i = \lambda_i(AA^T)x_i$$

Trong đó x_i là một vector khác 0. Nhân hai vế với x_i^T ta được

$$\begin{aligned} x_i^T AA^T x = x_i^T \lambda_i(AA^T)x_i &\Leftrightarrow (x_i^T A)(x_i^T A)^T = \lambda_i(AA^T)x_i^T x_i \\ &\Leftrightarrow \|x_i^T A\|^2 = \lambda_i(AA^T)\|x_i\| \\ &\Leftrightarrow \lambda_i(AA^T) = \frac{\|x_i^T A\|^2}{\|x_i\|^2} \qquad (2) \end{aligned}$$

Trước hết ta sẽ xét trường hợp $det(A) \neq 0$.
Do $det(A) \neq 0$ và $x_i \neq 0$ nên ta suy ra $\lambda_i(AA^T) > 0, \forall i$.
Từ đó ta có

$$0 < \frac{\lambda_i(AA^T)}{\lambda(AA^T)} < 1, \forall i$$

Đặt

$$X_0 = \alpha A^T$$

Trong đó

$$\alpha = \frac{1}{\|A\|^2} = \frac{1}{\lambda(AA^T)}$$

Ta có $E - AX_0 = E - \alpha AA^T$ là ma trận đối xứng nên

$$\begin{aligned} \|E - AX_0\| &= \sqrt{\lambda((E - AX_0)(E - AX_0)^T)} \\ &= \sqrt{\lambda((E - AX_0)^2)} \\ &= \sqrt{(\lambda(E - AX_0))^2} = \lambda(E - \alpha AA^T) \\ &= \lambda(E - \alpha AA^T) \\ &= max|\lambda_i(E - \alpha AA^T)| \\ &= max|1 - \alpha\lambda_i(AA^T)| \\ &= max|1 - \frac{\lambda_i(AA^T)}{\lambda(AA^T)}| \qquad (3) \\ &< 1 \end{aligned}$$

Như vậy, ma trận $X_0 = \frac{A^T}{\|A\|^2}$ thỏa mãn điều kiện hội tụ.

Mặt khác, ta xét trường hợp $det(A) = 0$. Lúc này, phương trình $Ax = 0$ có nghiệm x_0 khác 0 nên $\lambda_0 = 0$ là một giá trị riêng của ma trận A.

Từ (2) ta suy ra $\lambda_i(AA^T) \geq 0$, kéo theo

$$0 \leq \frac{\lambda_i(AA^T)}{\lambda(AA^T)} < 1, \forall i$$

Suy ra

$$|1 - \frac{\lambda_i(AA^T)}{\lambda(AA^T)}| \leq 1, \forall i$$

Đẳng thức xảy ra khi $i = 0$ nên

$$max|1-\frac{\lambda_i(AA^T)}{\lambda(AA^T)}|=1$$

Mà biến đổi (3) vẫn đúng nên ta có $\|E-AX_0\|=1$.

Vậy với cách chọn $X_0=\frac{A^T}{\|A\|^2}$ thì nếu ma trận A không khả nghịch, ta luôn có $\|E-AX_0\|=1$. Mà ngược lại, nếu ma trận A khả nghịch thì lại có $\|E-AX_0\|<1$ nên dãy (X_k) sẽ luôn hội tụ.

2.3 Công thức sai số

Giả sử $\|G_0\|\leq q<1$. Ta có

$$\begin{aligned}G_0=E-AX &\Leftrightarrow E-G_0=AX_0\\ &\Leftrightarrow A^{-1}(E-G_0)=X_0\\ &\Leftrightarrow A^{-1}(E-G_0^t)=X_0(E+G_0+G_0^2+...+G_0^t)\end{aligned}$$

Suy ra

$$\begin{aligned}\|A^{-1}\|-\|A^{-1}\|q^t &\leq \|A^{-1}\|-\|A^{-1}\|\|G_0\|^t\\ &\leq \|A^{-1}\|-\|A^{-1}G_0^t\|\\ &\leq \|A^{-1}-A^{-1}G_0^t\|\\ &= \|A^{-1}(E-G_0^t)\|\\ &= \|X_0(E+G_0+G_0^2+...+G_0^t)\|\\ &= \|X_0\|\|E+G_0+G_0^2+...+G_0^t\|\\ &\leq \|X_0\|(\|E\|+\|G_0\|+\|G_0^2\|+...+\|G_0^t\|)\\ &\leq \|X_0\|(1+q+q^2+...+q^t)\\ &= \frac{\|X_0\|(1-q^{t+1})}{1-q}\end{aligned}$$

Cho $t\to\infty$ ta được $\|A^{-1}\|\leq\frac{\|X_0\|}{1-q}$. Kết hợp với (1) ta có công thức sai số

$$\|A^{-1}-X_k\|\leq\frac{\|X_0\|q^{2^k}}{1-q}$$

2.4 Thuật toán

Gói lặp newton_iter(A, n, ε, X_0)
Input: Ma trận A, kích cỡ n, sai số ε, ma trận xấp xỉ đầu X_0 **Output:** Ma trận X^* xấp xỉ của A^{-1}
Bước 1: Tính các giá trị $E \leftarrow$ ma trận đơn vị cỡ n $X \leftarrow X_0$ $q \leftarrow \|E - AX_0\|$ $q^{2^k} \leftarrow q$ $error \leftarrow \frac{\varepsilon \times (1-q)}{\|X_0\|}$ **Bước 2:** Nếu $q = 1$, thông báo ma trận không khả nghịch và trả về ma trận NaN **Bước 3:** While $q^{2^k} \geq error$: $X \leftarrow X(2E - AX)$ $q^{2^k} \leftarrow (q^{2^k})^2$ **Bước 4:** Trả về ma trận X_0

Chương trình newton_inverse(A, n, ε)
Input: Ma trận A, kích cỡ n, sai số ε **Output:** Ma trận X^* xấp xỉ của A^{-1}
Bước 1: Chương trình tự chọn xấp xỉ đầu $X_0 = \frac{A^T}{\|A\|^2}$ **Bước 2:** Trả về ***newton_iter***(A, n, ε, X_0)

3 Các phương pháp tìm gần đúng nghịch đảo của ma trận chéo trội

3.1 Giới thiệu

Ma trận $A = (a_{ij}) \in n \times n$ được gọi là ma trận chéo trội, nếu 1 trong 2 điều kiện dưới đây thỏa mãn:

- $\forall i = \overline{1,n} \quad \sum_{j \neq i} \|a_{ij}\| < \|a_{ii}\|$ (chéo trội hàng)
- $\forall j = \overline{1,n} \quad \sum_{i \neq j} \|a_{ij}\| < \|a_{jj}\|$ (chéo trội cột)

Ý tưởng của các phương pháp này là áp dụng các phương pháp giải gần đúng hệ vuông để giải hệ phương trình $AX = E$ trong đó A là ma trận chéo trội. Các phương pháp sẽ được trình bày ở đây là:

- Phương pháp lặp Jacobi
- Phương pháp lặp Gauss-Seidel

3.2 Phương pháp lặp Jacobi

3.2.1 Lí thuyết

Đặt $T = diag\{a_{11}^{-1}, a_{22}^{-1}, ..., a_{nn}^{-1}\}, B = E - TA, B_1 = E - AT$

- Công thức lặp:
$$X_{k+1} = BX_k + T$$
- Công thức sai số:
  - Trường hợp chéo trội hàng: Với $\|B\|_\infty \leq q < 1$
$$\|X_k - X^*\|_\infty \leq \frac{q}{1-q}\|X_k - X_{k-1}\|_\infty$$
$$\|X_k - X^*\|_\infty \leq \frac{q^k}{1-q}\|X_1 - X_0\|_\infty$$
  - Trường hợp chéo trội cột: Với $\lambda = \frac{max\|a_{ii}\|}{min\|a_{ii}\|}$, $\|B_1\|_1 \leq q < 1$
$$\|X_k - X^*\|_1 \leq \lambda\frac{q}{1-q}\|X_k - X_{k-1}\|_1$$
$$\|X_k - X^*\|_1 \leq \lambda\frac{q^k}{1-q}\|X_1 - X_0\|_1$$

3.2.2 Thuật toán

Hàm check_dom(A, n)
Input: Ma trận A, kích cỡ n
Output: Trả về trạng thái trội 1, -1, 0
tương ứng với ma trận A chéo trội hàng, chéo trội cột hoặc không chéo trội
Bước 1: $K \leftarrow \lvert diag(A) \rvert$
$sumRow$ là dãy tổng trị tuyệt đối các phần tử trên các hàng của A trừ các phần tử trong K
$sumCol$ là dãy tổng trị tuyệt đối các phần tử trên các cột của A trừ các phần tử trong K
Bước 2:
Nếu $K[i] > sumRow[i], \forall i = \overline{1, n}$, hàm trả về giá trị 1
Nếu $K[i] > sumCol[i], \forall i = \overline{1, n}$, hàm trả về giá trị -1
Nếu cả hai trường hợp trên không xảy ra, hàm trả về giá trị 0

Hàm get_norm$(A, domStatus)$
Input: Ma trận A, trạng thái trội $domStatus$
Output: Chuẩn của A theo trạng thái trội
Nếu $domStatus = 1$, trả về $\lVert A \rVert_\infty$. Nếu không, trả về $\lVert A \rVert_1$.

Hàm get_lambda$(A, domStatus)$
Input: Ma trận A, trạng thái trội $domStatus$
Output: Trả về $\lambda = 1$ nếu A chéo trội hàng, $\lambda = \frac{max\lVert a_{ii} \rVert}{min\lVert a_{ii} \rVert}$ nếu A chéo trội cột
Nếu $domStatus = 1$, trả về giá trị 1
Nếu không, tính $K \leftarrow \lvert diag(A) \rvert$, trả về $\frac{max(K)}{min(K)}$

Hàm get_q$(B, E, T, A, domStatus)$
Input: Ma trận B, E, T, A, trạng thái trội $domStatus$
Output: Trả về hệ số q
Nếu $domStatus = 1$, trả về ***get_norm***$(B, domStatus)$
Nếu không, trả về ***get_norm***$(E - AT, domStatus)$

Gói đánh giá tiên nghiệm predecessor_iter$(X_0,\, B,\, T,\, domStatus,\, \lambda,\, q,\, \varepsilon)$
Input: Ma trận X_0, B, T, $domStatus$, λ, hệ số q, sai số ε **Output:** Ma trận X^* xấp xỉ của A^{-1}
Bước 1: Tính các giá trị $norm \leftarrow$ ***get_norm***$((BX_0 + T) - X_0,\, domStatus)$ $X \leftarrow X_0$ $q^k \leftarrow 1$ $error \leftarrow \frac{\varepsilon \times (1-q)}{\lambda \times norm}$ **Bước 2:** While $q^k > error$: $X \leftarrow BX + T$ $q^k \leftarrow q^k \times q$ **Bước 3:** Trả về X

Gói đánh giá hậu nghiệm successor_iter$(X_0,\, B,\, T,\, domStatus,\, \lambda,\, q,\, \varepsilon)$
Input: Ma trận X_0, B, T, $domStatus$, λ, hệ số q, sai số ε **Output:** Ma trận X^* xấp xỉ của A^{-1}
Bước 1: Tính các giá trị $oldX \leftarrow X_0$ $newX \leftarrow BX_0 + T$ $error \leftarrow \frac{\varepsilon(1-q)}{\lambda q}$ **Bước 2:** While ***get_norm***$(newX - oldX,\, domStatus) > error$: $oldX \leftarrow newX$ $newX \leftarrow B \times oldX + T$ **Bước 3:** Trả về $newX$

Chương trình jacobi_ inverse(A, n, ε, $mode$)
Input: Ma trận A, kích cỡ n, sai số ε, chế độ đánh giá $mode$ **Output:** Ma trận X^* xấp xỉ của A^{-1} theo chế độ đánh giá tiên nghiệm nếu $mode = 1$. Nếu không, thực hiện theo chế độ đánh giá hậu nghiệm.
Bước 1: $domStatus \leftarrow$ ***check_ dom***(A, n) Nếu $domStatus = 0$, thông báo A không chéo trội và dừng chương trình **Bước 2:** Tính các giá trị $E \leftarrow$ ma trận đơn vị cỡ n $T \leftarrow diag\{a_{11}^{-1}, a_{22}^{-1}, ..., a_{nn}^{-1}\}$ $B \leftarrow E - TA$ $\lambda \leftarrow$ ***get_ lambda***(A, $domStatus$) $q \leftarrow$ ***get_ q***(B, E, T, A, $domStatus$) **Bước 3:** Chọn chế độ đánh giá Nếu $mode = 1$, trả về ***predecessor_ iter***(A, B, T, $domStatus$, λ, q, ε) Nếu không, trả về ***successor_ iter***(A, B, T, $domStatus$, λ, q, ε)

3.3 Phương pháp lặp Gauss-Seidel

3.3.1 Lí thuyết

Đặt $T = diag\{a_{11}^{-1}, a_{22}^{-1}, ..., a_{nn}^{-1}\}$, $B = E - TA$, $B_1 = E - AT$
Kí hiệu $X^{(i)}$ là dòng thứ i của ma trận X

- Công thức lặp:

$$X_{k+1}^{(i)} = \sum_{j=1}^{i-1} b_{ij} X_{k+1}^{(j)} + \sum_{j=i+1}^{n} b_{ij} X_{k}^{(j)} + T^{(i)}, \qquad \forall i = \overline{1, n}$$

- Công thức sai số:
  - Trường hợp chéo trội hàng: Đặt

$$q = \max_{1 \le i \le n} \frac{\sum_{j=i}^{n} |b_{ij}|}{1 - \sum_{j=1}^{i-1} |b_{ij}|} \le \|B\|_\infty < 1$$

$\Rightarrow$ Công thức sai số là:

$$\|X_k - X^*\|_\infty \le \frac{q}{1-q} \|X_k - X_{k-1}\|_\infty$$

$$\|X_k - X^*\|_\infty \le \frac{q^k}{1-q} \|X_1 - X_0\|_\infty$$

– Trường hợp chéo trội cột: Đặt

$$\lambda = \frac{max|a_{ii}|}{min|a_{ii}|}$$

$$q = \max_{1 \le i \le n} \frac{\sum_{j=1}^{i} |b_{1_{ji}}|}{1 - \sum_{j=i+1}^{n} |b_{1_{ji}}|} \le \|B_1\|_1 < 1$$

$$S = \max_{1 \le i \le n} \sum_{j=i+1}^{n} |b_{1_{ij}}|$$

$\Rightarrow$ Công thức sai số là:

$$\|X_k - X^*\|_1 \le \frac{\lambda q}{(1-S)(1-q)} \|X_k - X_{k-1}\|_1$$

$$\|X_k - X^*\|_1 \le \frac{\lambda q^k}{(1-S)(1-q)} \|X_1 - X_0\|_1$$

3.3.2 Thuật toán

Hàm check_ dom(A, n): như trên
Hàm get_ norm$(A, domStatus)$: như trên
Hàm get_ lambda$(A, domStatus)$: như trên

Hàm get_ q$(B, n, domStatus)$
Input: Ma trận B, kích cỡ n, trạng thái trội $domStatus$ **Output:** Hệ số q
Bước 1: $q \leftarrow 0$ **Bước 2:** Nếu $domStatus = 1$: For i from 1 to n: $q_1 \leftarrow \sum_{j=i}^{n} |b_{ij}|, q_2 \leftarrow \sum_{j=1}^{i-1} |b_{ij}|$ $q \leftarrow max(q, \frac{q_1}{1-q_2})$ Nếu không: For i from 1 to n: $q_1 \leftarrow \sum_{j=1}^{i} |b_{ji}|, q_2 \leftarrow \sum_{j=i+1}^{n} |b_{ji}|$ $q \leftarrow max(q, \frac{q_1}{1-q_2})$ **Bước 3:**Trả về q

Hàm get_S(B_1, n)

Input: Ma trận B_1, kích cỡ n

Output: Hệ số S

Bước 1: $S \leftarrow 0$

Bước 2: For i from 1 to n:

$tmp \leftarrow \sum_{j=i+1}^{n} |b_{1_{ji}}|$

$S \leftarrow max(S, tmp)$

Bước 3: Trả về S

Hàm next_iter$(oldX, B, T, n)$

Input: Ma trận $oldX$, B, T, kích cỡ n

Output: Ma trận lặp tiếp theo $newX$

Bước 1: For i from 1 to n:

$newX_{k+1}^{(i)} \leftarrow \sum_{j=1}^{i-1} b_{ij} newX_{k+1}^{(j)} + \sum_{j=i+1}^{(k)} b_{ij} oldX_k^{(j)} + T^{(i)}$

Bước 2: Trả về $newX$

Gói đánh giá tiên nghiệm predecessor_iter$(X_0, B, T, n, domStatus, \lambda, S, q, \varepsilon)$

Input: Ma trận X_0, B, T, kích cỡ n, $domStatus$, λ, hệ số S, q, sai số ε

Output: Ma trận X^* xấp xỉ của A^{-1}

Bước 1: Tính các giá trị

$q^k \leftarrow 1$

$X \leftarrow X_0$

$X_1 \leftarrow$ ***next_iter***(X_0, B, T, n)

$norm \leftarrow$ ***get_norm***$(X_1 - X_0, domStatus)$

$error \leftarrow \frac{\varepsilon(1-q)(1-S)}{\lambda \times norm}$

Bước 2: While $q^k > error$:

$X \leftarrow$ ***next_iter***(X, B, T, n)

$q^k \leftarrow q^k \times q$

Bước 3: Trả về X

Gói đánh giá hậu nghiệm successor_iter$(X_0, B, T, n, domStatus, \lambda, S, q, \varepsilon)$
Input: Ma trận X_0, B, T, kích cỡ n, $domStatus$, λ, hệ số S, q, sai số ε **Output:** Ma trận X^* xấp xỉ của A^{-1}
Bước 1: $oldX \leftarrow X_0$; $newX \leftarrow$ ***next_iter***(X_0, B, T, n); $error \leftarrow \frac{\varepsilon(1-q)(1-S)}{\lambda \times norm}$ **Bước 2:** While ***get_norm***$(newX - oldX, domStatus) > error$: $oldX \leftarrow newX$ $newX \leftarrow$ ***next_iter***$(newX, B, T, n)$ **Bước 3:** Trả về $newX$

Chương trình gauss_seidel_inverse$(A, n, \varepsilon, mode)$
Input: Ma trận A, kích cỡ n, sai số ε, chế độ đánh giá $mode$ **Output:** Ma trận X^* xấp xỉ của A^{-1} theo chế độ đánh giá tiên nghiệm nếu $mode = 1$. Nếu không, thực hiện theo chế độ đánh giá hậu nghiệm.
Bước 1: $domStatus \leftarrow$ ***check_dom***(A, n) Nếu $domStatus = 0$, thông báo A không chéo trội và dừng chương trình **Bước 2:** Tính các giá trị $E \leftarrow$ ma trận đơn vị cỡ n $T \leftarrow diag\{a_{11}^{-1}, a_{22}^{-1}, ..., a_{nn}^{-1}\}$ $B \leftarrow E - TA$ $\lambda \leftarrow$ ***get_lambda***$(A, domStatus)$ Nếu $domStatus = 1$: $S \leftarrow 0$ $q \leftarrow$ ***get_q***$(B, n, domStatus)$ Nếu không: $B_1 \leftarrow E - AT$ $S \leftarrow$ ***get_S***(B_1, n) $q \leftarrow$ ***get_q***$(B_1, n, domStatus)$ **Bước 3:** Chọn chế độ đánh giá Nếu $mode = 1$, trả về ***predecessor_iter***$(A, B, T, n, domStatus, \lambda, S, q, \varepsilon)$ Nếu không, trả về ***successor_iter***$(A, B, T, n, domStatus, \lambda, S, q, \varepsilon)$

4 Hệ thống ví dụ

4.1 Ví dụ 1 (Ma trận không chéo trội)

Tìm nghịch đảo ma trận sau:

$$\begin{bmatrix} 1 & 2 & 3 & 8 \\ 2 & 4 & 0 & 6 \\ 1 & 2 & 7 & 1 \\ 3 & 5 & 0 & 0 \end{bmatrix}$$

Đây là ma trận không chéo trội nên ta không thể sử dụng các phương pháp Jacobi và Gauss-Seidel để giải ma trận nghịch đảo. Khi chạy phương pháp Newton với sai số 10^{-10} ta được kết quả như sau:

```
--------------------------------------------------------------------
Ma trận A:
[[1. 2. 3. 8.]
 [2. 4. 0. 6.]
 [1. 2. 7. 1.]
 [3. 5. 0. 0.]]
--------------------------------------------------------------------
--------------------------------------------------------------------
Phương pháp Newton kết thúc sau 17 bước lặp.
Ma trận nghịch đảo của A:
[[ 2.56097561e+00 -3.23170732e+00 -1.09756098e+00  2.00000000e+00]
 [-1.53658537e+00  1.93902439e+00  6.58536585e-01 -1.00000000e+00]
 [ 4.87804878e-02 -8.53658537e-02  1.21951220e-01 -3.36927471e-15]
 [ 1.70731707e-01 -4.87804878e-02 -7.31707317e-02 -8.52308592e-15]]
--------------------------------------------------------------------
Kiểm tra nhân ngược:
[[ 1.00000000e+00  5.70654635e-14 -2.22044605e-15 -1.11022302e-14]
 [ 6.37268016e-14  1.00000000e+00  2.22044605e-16  4.66293670e-15]
 [-1.30930667e-15  7.50661375e-16  1.00000000e+00 -8.32667268e-17]
 [-3.33704169e-15  1.84900255e-15 -1.80411242e-16  1.00000000e+00]]
[Finished in 325ms]
```

4.2 Ví dụ 2 (Ma trận chéo trội hàng)

Tìm nghịch đảo ma trận sau:

$$\begin{bmatrix} 15 & 1 & 4 & -8 \\ 2 & 16 & 9 & 3 \\ 1 & -9 & 12 & 1 \\ 1 & 0 & 5 & 23 \end{bmatrix}$$

Ta có thể áp dụng cả 3 phương pháp Newton, Jacobi, Gauss-Seidel cho ma trận trên. Với sai số 10^{-10} ta thu được các kết quả sau:

```
--------------------------------------------------------------------
Ma trận A:
[[15.  1.  4. -8.]
 [ 2. 16.  9.  3.]
 [ 1. -9. 12.  1.]
 [ 1.  0.  5. 23.]]
--------------------------------------------------------------------
--------------------------------------------------------------------
Phương pháp Newton kết thúc sau 7 bước lặp.
Ma trận nghịch đảo của A:
[[ 0.0685159  -0.01648494 -0.02169369  0.02692503]
 [-0.00364951  0.04556805 -0.03050675 -0.00588667]
 [-0.00834981  0.03614485  0.06332976 -0.0103723 ]
 [-0.00116378 -0.00714084 -0.01282413  0.04456245]]
--------------------------------------------------------------------
Kiểm tra nhân ngược:
[[ 1.00000000e+00  3.74673903e-12  5.21050772e-11 -2.36412626e-11]
 [ 3.74671474e-12  1.00000000e+00 -3.34758263e-12  1.51886489e-12]
 [ 5.21051605e-11 -3.34768324e-12  1.00000000e+00  2.11228743e-11]
 [-2.36414915e-11  1.51890479e-12  2.11228882e-11  1.00000000e+00]]
[Finished in 541ms]
```

```
--------------------------------------------------------------------
Ma trận A:
[[15.  1.  4. -8.]
 [ 2. 16.  9.  3.]
 [ 1. -9. 12.  1.]
 [ 1.  0.  5. 23.]]
--------------------------------------------------------------------
--------------------------------------------------------------------
A chéo trội hàng.
Phương pháp Jacobi đánh giá tiên nghiệm kết thúc sau 336 bước lặp.
Ma trận nghịch đảo của A:
[[ 0.0685159  -0.01648494 -0.02169369  0.02692503]
 [-0.00364951  0.04556805 -0.03050675 -0.00588667]
 [-0.00834981  0.03614485  0.06332976 -0.0103723 ]
 [-0.00116378 -0.00714084 -0.01282413  0.04456245]]
--------------------------------------------------------------------
Kiểm tra nhân ngược:
[[ 1.00000000e+00 -6.24500451e-17 -1.38777878e-17 -1.38777878e-17]
 [ 8.67361738e-18  1.00000000e+00  6.07153217e-17  1.56125113e-17]
 [-1.73472348e-18  9.71445147e-17  1.00000000e+00  1.73472348e-18]
 [ 0.00000000e+00 -1.21430643e-17  6.93889390e-18  1.00000000e+00]]
[Finished in 348ms]
```

```
--------------------------------------------------------------------
Ma trận A:
[[15.  1.  4. -8.]
 [ 2. 16.  9.  3.]
 [ 1. -9. 12.  1.]
 [ 1.  0.  5. 23.]]
--------------------------------------------------------------------
--------------------------------------------------------------------
A chéo trội hàng.
Phương pháp Jacobi đánh giá hậu nghiệm kết thúc sau 67 bước lặp.
Ma trận nghịch đảo của A:
[[ 0.0685159  -0.01648494 -0.02169369  0.02692503]
 [-0.00364951  0.04556805 -0.03050675 -0.00588667]
 [-0.00834981  0.03614485  0.06332976 -0.0103723 ]
 [-0.00116378 -0.00714084 -0.01282413  0.04456245]]
--------------------------------------------------------------------
Kiểm tra nhân ngược:
[[ 1.00000000e+00  1.65811045e-11  7.90560326e-12  4.57126351e-12]
 [-5.92288892e-12  1.00000000e+00 -6.18872141e-12 -7.85573169e-12]
 [ 9.25497179e-12  1.22516441e-11  1.00000000e+00  2.88745416e-11]
 [-1.85603616e-12  1.32685685e-11 -7.34433347e-13  1.00000000e+00]]
[Finished in 357ms]
```

```
-------------------------------------------------------------------
Ma trận A:
[[15.  1.  4. -8.]
 [ 2. 16.  9.  3.]
 [ 1. -9. 12.  1.]
 [ 1.  0.  5. 23.]]
-------------------------------------------------------------------
-------------------------------------------------------------------
A chéo trội hàng.
Phương pháp Gauss-Seidel đánh giá tiên nghiệm kết thúc sau 201 bước lặp.
Ma trận nghịch đảo của A:
[[ 0.0685159  -0.01648494 -0.02169369  0.02692503]
 [-0.00364951  0.04556805 -0.03050675 -0.00588667]
 [-0.00834981  0.03614485  0.06332976 -0.0103723 ]
 [-0.00116378 -0.00714084 -0.01282413  0.04456245]]
-------------------------------------------------------------------
Kiểm tra nhân ngược:
[[ 1.00000000e+00 -6.93889390e-18  4.16333634e-17 -1.38777878e-17]
 [ 7.80625564e-18  1.00000000e+00  5.63785130e-17 -4.33680869e-18]
 [ 0.00000000e+00  9.71445147e-17  1.00000000e+00  4.16333634e-17]
 [ 0.00000000e+00 -1.21430643e-17  6.93889390e-18  1.00000000e+00]]
[Finished in 348ms]
```

```
-------------------------------------------------------------------
Ma trận A:
[[15.  1.  4. -8.]
 [ 2. 16.  9.  3.]
 [ 1. -9. 12.  1.]
 [ 1.  0.  5. 23.]]
-------------------------------------------------------------------
-------------------------------------------------------------------
A chéo trội hàng.
Phương pháp Gauss-Seidel đánh giá hậu nghiệm kết thúc sau 23 bước lặp.
Ma trận nghịch đảo của A:
[[ 0.0685159  -0.01648494 -0.02169369  0.02692503]
 [-0.00364951  0.04556805 -0.03050675 -0.00588667]
 [-0.00834981  0.03614485  0.06332976 -0.0103723 ]
 [-0.00116378 -0.00714084 -0.01282413  0.04456245]]
-------------------------------------------------------------------
Kiểm tra nhân ngược:
[[ 1.00000000e+00 -6.56398547e-12  5.19126409e-12  1.14187548e-11]
 [ 8.68419052e-13  1.00000000e+00  4.14834399e-12  9.12463941e-12]
 [ 5.09278383e-13 -3.07603942e-12  1.00000000e+00  5.35114487e-12]
 [-1.57963920e-13  9.54101381e-13 -7.54570018e-13  1.00000000e+00]]
[Finished in 336ms]
```

4.3 Ví dụ 3 (Ma trận chéo trội cột gần suy biến)

Tìm nghịch đảo ma trận sau:

$$\begin{bmatrix} 3 & 3.001 & 0 \\ 2 & 7.1 & 0.0001 \\ 0 & 4 & 0.001 \end{bmatrix}$$

Vẫn với sai số là 10^{-10}, ta thu được các kết quả sau:

```
---------------------------------------------------------------------
Ma trận A:
[[3.000e+00 3.001e+00 0.000e+00]
 [2.000e+00 7.100e+00 1.000e-04]
 [0.000e+00 4.000e+00 1.000e-03]]
---------------------------------------------------------------------
---------------------------------------------------------------------
Phương pháp Newton kết thúc sau 33 bước lặp.
Ma trận nghịch đảo của A:
[[ 4.75244716e-01 -2.12867073e-01  2.12867073e-02]
 [-1.41864094e-01  2.12796141e-01 -2.12796141e-02]
 [ 5.67456377e+02 -8.51184565e+02  1.08511846e+03]]
---------------------------------------------------------------------
Kiểm tra nhân ngược:
[[ 1.00000000e+00  2.49800181e-16  4.50887676e-21]
 [ 0.00000000e+00  1.00000000e+00 -5.70135149e-21]
 [-2.27373675e-13  9.09494702e-13  1.00000000e+00]]
[Finished in 319ms]
```

```
---------------------------------------------------------------------
Ma trận A:
[[3.000e+00 3.001e+00 0.000e+00]
 [2.000e+00 7.100e+00 1.000e-04]
 [0.000e+00 4.000e+00 1.000e-03]]
---------------------------------------------------------------------
---------------------------------------------------------------------
A chéo trội cột.
Phương pháp Jacobi đánh giá tiên nghiệm kết thúc sau 3306 bước lặp.
Ma trận nghịch đảo của A:
[[ 4.75244716e-01 -2.12867073e-01  2.12867073e-02]
 [-1.41864094e-01  2.12796141e-01 -2.12796141e-02]
 [ 5.67456377e+02 -8.51184565e+02  1.08511846e+03]]
---------------------------------------------------------------------
Kiểm tra nhân ngược:
[[ 1.00000000e+00  4.16333634e-17  4.50887676e-21]
 [-1.11022302e-16  1.00000000e+00 -2.31321970e-21]
 [ 4.54747351e-13  9.09494702e-13  1.00000000e+00]]
[Finished in 338ms]
```

```
-------------------------------------------------------------------
Ma trận A:
[[3.000e+00 3.001e+00 0.000e+00]
 [2.000e+00 7.100e+00 1.000e-04]
 [0.000e+00 4.000e+00 1.000e-03]]
-------------------------------------------------------------------
-------------------------------------------------------------------
A chéo trội cột.
Phương pháp Jacobi đánh giá hậu nghiệm kết thúc sau 78 bước lặp.
Ma trận nghịch đảo của A:
[[ 4.75244716e-01 -2.12867073e-01  2.12867073e-02]
 [-1.41864094e-01  2.12796141e-01 -2.12796141e-02]
 [ 5.67456377e+02 -8.51184565e+02  1.08511846e+03]]
-------------------------------------------------------------------
Kiểm tra nhân ngược:
[[ 1.00000000e+00  4.16333634e-17  4.50887676e-21]
 [-1.11022302e-16  1.00000000e+00 -2.31321970e-21]
 [ 4.54747351e-13  9.09494702e-13  1.00000000e+00]]
[Finished in 328ms]
```

```
-------------------------------------------------------------------
Ma trận A:
[[3.000e+00 3.001e+00 0.000e+00]
 [2.000e+00 7.100e+00 1.000e-04]
 [0.000e+00 4.000e+00 1.000e-03]]
-------------------------------------------------------------------
-------------------------------------------------------------------
A chéo trội cột.
Phương pháp Gauss-Seidel đánh giá tiên nghiệm kết thúc sau 1402 bước lặp.
Ma trận nghịch đảo của A:
[[ 4.75244716e-01 -2.12867073e-01  2.12867073e-02]
 [-1.41864094e-01  2.12796141e-01 -2.12796141e-02]
 [ 5.67456377e+02 -8.51184565e+02  1.08511846e+03]]
-------------------------------------------------------------------
Kiểm tra nhân ngược:
[[ 1.00000000e+00 -1.52655666e-16  1.12074497e-21]
 [-5.55111512e-17  1.00000000e+00  1.07491209e-21]
 [ 2.27373675e-13  9.09494702e-13  1.00000000e+00]]
[Finished in 368ms]
```

```
-----------------------------------------------------------------
Ma trận A:
[[3.000e+00 3.001e+00 0.000e+00]
 [2.000e+00 7.100e+00 1.000e-04]
 [0.000e+00 4.000e+00 1.000e-03]]
-----------------------------------------------------------------
-----------------------------------------------------------------
A chéo trội cột.
Phương pháp Gauss-Seidel đánh giá hậu nghiệm kết thúc sau 38 bước lặp.
Ma trận nghịch đảo của A:
[[ 4.75244716e-01 -2.12867073e-01  2.12867073e-02]
 [-1.41864094e-01  2.12796141e-01 -2.12796141e-02]
 [ 5.67456377e+02 -8.51184565e+02  1.08511846e+03]]
-----------------------------------------------------------------
Kiểm tra nhân ngược:
[[ 1.00000000e+00 -1.52655666e-16  1.12074497e-21]
 [-5.55111512e-17  1.00000000e+00  1.07491209e-21]
 [ 2.27373675e-13  9.09494702e-13  1.00000000e+00]]
[Finished in 328ms]
```

4.4 Ví dụ 4 (Ma trận chéo trội cỡ lớn)

Tìm nghịch đảo ma trận 100×100 sau:

$$\begin{bmatrix} 999 & 5 & 1 & \dots & 0 & 2 & 5 \\ 0 & 999 & 7 & \dots & 8 & 6 & 9 \\ 9 & 7 & 999 & \dots & 8 & 8 & 3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 6 & 9 & 8 & \dots & 999 & 5 & 8 \\ 3 & 9 & 3 & \dots & 4 & 999 & 7 \\ 1 & 8 & 7 & \dots & 5 & 0 & 999 \end{bmatrix}$$

Do kích cỡ của ma trận này quá lớn để trình bày ở đây nên ma trận đầy đủ sẽ được đính kèm dưới tên file matrancheotroi100.txt. Chọn sai số 10^{-10} ta được kết quả như sau:

```
--------------------------------------------------------------------
Ma trận A:
[[999.    5.    1. ...    0.    2.    5.]
 [  0.  999.    7. ...    8.    6.    9.]
 [  9.    7.  999. ...    8.    8.    3.]
 ...
 [  6.    9.    8. ...  999.    5.    8.]
 [  3.    9.    3. ...    4.  999.    7.]
 [  1.    8.    7. ...    5.    0.  999.]]
--------------------------------------------------------------------
--------------------------------------------------------------------
Phương pháp Newton kết thúc sau 5 bước lặp.
Ma trận nghịch đảo của A:
[[ 1.00244304e-03 -3.45959074e-06  3.54674807e-07 ...  1.21327941e-06
  -5.62693147e-07 -3.59949943e-06]
 [ 1.51521929e-06  1.00250176e-03 -5.51422571e-06 ... -6.76570027e-06
  -4.73958374e-06 -7.68281909e-06]
 [-7.45077964e-06 -5.33089198e-06  1.00242208e-03 ... -6.63351876e-06
  -6.70475446e-06 -1.39146292e-06]
 ...
 [-4.56705435e-06 -7.50822517e-06 -6.75048495e-06 ...  1.00249977e-03
  -3.55687126e-06 -6.78505638e-06]
 [-1.49983476e-06 -7.57941233e-06 -1.49847491e-06 ... -2.59865127e-06
   1.00243644e-03 -5.80656674e-06]
 [ 5.59454591e-07 -6.67856183e-06 -5.49126163e-06 ... -3.73954046e-06
   1.49756506e-06  1.00245737e-03]]
--------------------------------------------------------------------
Kiểm tra nhân ngược:
[[ 9.99999998e-01 -2.73850895e-10  1.30134681e-10 ...  3.58990111e-11
  -2.05209577e-10 -1.19083965e-10]
 [-2.73850892e-10  9.99999998e-01  5.37557115e-11 ...  3.35787490e-10
   1.74908231e-10  2.41580352e-10]
 [ 1.30134692e-10  5.37557185e-11  9.99999998e-01 ...  3.97266677e-10
   1.33756658e-10 -1.95935474e-12]
 ...
 [ 3.58990178e-11  3.35787479e-10  3.97266671e-10 ...  9.99999998e-01
  -5.71353338e-11  3.63753618e-10]
 [-2.05209576e-10  1.74908233e-10  1.33756655e-10 ... -5.71353366e-11
   9.99999998e-01 -5.87579211e-11]
 [-1.19083964e-10  2.41580339e-10 -1.95936409e-12 ...  3.63753617e-10
  -5.87579201e-11  9.99999998e-01]]
[Finished in 366ms]
```

```
---------------------------------------------------------------------
Ma trận A:
[[999.   5.   1. ...   0.   2.   5.]
 [  0. 999.   7. ...   8.   6.   9.]
 [  9.   7. 999. ...   8.   8.   3.]
 ...
 [  6.   9.   8. ... 999.   5.   8.]
 [  3.   9.   3. ...   4. 999.   7.]
 [  1.   8.   7. ...   5.   0. 999.]]
---------------------------------------------------------------------
---------------------------------------------------------------------
A chéo trội hàng.
Phương pháp Jacobi đánh giá tiên nghiệm kết thúc sau 47 bước lặp.
Ma trận nghịch đảo của A:
[[ 1.00244304e-03 -3.45959045e-06  3.54674684e-07 ...  1.21327937e-06
  -5.62692929e-07 -3.59949931e-06]
 [ 1.51521957e-06  1.00250176e-03 -5.51422576e-06 ... -6.76570062e-06
  -4.73958392e-06 -7.68281933e-06]
 [-7.45077978e-06 -5.33089203e-06  1.00242208e-03 ... -6.63351918e-06
  -6.70475461e-06 -1.39146291e-06]
 ...
 [-4.56705440e-06 -7.50822552e-06 -6.75048536e-06 ...  1.00249977e-03
  -3.55687120e-06 -6.78505677e-06]
 [-1.49983455e-06 -7.57941251e-06 -1.49847504e-06 ... -2.59865121e-06
   1.00243644e-03 -5.80656669e-06]
 [ 5.59454719e-07 -6.67856208e-06 -5.49126163e-06 ... -3.73954084e-06
   1.49756513e-06  1.00245737e-03]]
---------------------------------------------------------------------
Kiểm tra nhân ngược:
[[ 1.00000000e+00 -5.05986339e-13 -4.69380554e-13 ... -4.46502942e-13
  -4.68670489e-13 -4.67098793e-13]
 [-5.31206325e-13  1.00000000e+00 -4.73797406e-13 ... -4.50701583e-13
  -4.73079562e-13 -4.71493274e-13]
 [-5.90405715e-13 -5.67661874e-13  1.00000000e+00 ... -5.00931461e-13
  -5.25800250e-13 -5.24038834e-13]
 ...
 [-5.38618293e-13 -5.17879061e-13 -4.80404688e-13 ...  1.00000000e+00
  -4.79680536e-13 -4.78073009e-13]
 [-5.26764802e-13 -5.06466390e-13 -4.69834479e-13 ... -4.46934233e-13
   1.00000000e+00 -4.67550092e-13]
 [-5.45899263e-13 -5.24875282e-13 -4.86901752e-13 ... -4.63170734e-13
  -4.86163862e-13  1.00000000e+00]]
[Finished in 342ms]
```

```
----------------------------------------------------------------------
Ma trận A:
[[999.   5.   1. ...   0.   2.   5.]
 [  0. 999.   7. ...   8.   6.   9.]
 [  9.   7. 999. ...   8.   8.   3.]
 ...
 [  6.   9.   8. ... 999.   5.   8.]
 [  3.   9.   3. ...   4. 999.   7.]
 [  1.   8.   7. ...   5.   0. 999.]]
----------------------------------------------------------------------
----------------------------------------------------------------------
A chéo trội hàng.
Phương pháp Jacobi đánh giá hậu nghiệm kết thúc sau 38 bước lặp.
Ma trận nghịch đảo của A:
[[ 1.00244304e-03 -3.45959069e-06  3.54674463e-07 ...  1.21327916e-06
  -5.62693150e-07 -3.59949953e-06]
 [ 1.51521932e-06  1.00250176e-03 -5.51422598e-06 ... -6.76570083e-06
  -4.73958414e-06 -7.68281955e-06]
 [-7.45078005e-06 -5.33089229e-06  1.00242208e-03 ... -6.63351941e-06
  -6.70475486e-06 -1.39146316e-06]
 ...
 [-4.56705466e-06 -7.50822576e-06 -6.75048559e-06 ...  1.00249977e-03
  -3.55687143e-06 -6.78505699e-06]
 [-1.49983479e-06 -7.57941275e-06 -1.49847526e-06 ... -2.59865142e-06
   1.00243644e-03 -5.80656691e-06]
 [ 5.59454462e-07 -6.67856233e-06 -5.49126186e-06 ... -3.73954106e-06
   1.49756490e-06  1.00245737e-03]]
----------------------------------------------------------------------
Kiểm tra nhân ngược:
[[ 1.00000000e+00 -3.43863819e-10 -3.18989597e-10 ... -3.03442663e-10
  -3.18506959e-10 -3.17439741e-10]
 [-3.61006795e-10  1.00000000e+00 -3.21990272e-10 ... -3.06297098e-10
  -3.21503095e-10 -3.20425835e-10]
 [-4.01238386e-10 -3.85780085e-10  1.00000000e+00 ... -3.40431683e-10
  -3.57332282e-10 -3.56134977e-10]
 ...
 [-3.66043387e-10 -3.51941036e-10 -3.26482531e-10 ...  1.00000000e+00
  -3.25988556e-10 -3.24896272e-10]
 [-3.57988140e-10 -3.44196127e-10 -3.19297866e-10 ... -3.03735907e-10
   1.00000000e+00 -3.17746512e-10]
 [-3.70991993e-10 -3.56698986e-10 -3.30896302e-10 ... -3.14769060e-10
  -3.30395647e-10  1.00000000e+00]]
[Finished in 333ms]
```

```
-------------------------------------------------------------------
Ma trận A:
[[999.   5.   1. ...   0.   2.   5.]
 [  0. 999.   7. ...   8.   6.   9.]
 [  9.   7. 999. ...   8.   8.   3.]
 ...
 [  6.   9.   8. ... 999.   5.   8.]
 [  3.   9.   3. ...   4. 999.   7.]
 [  1.   8.   7. ...   5.   0. 999.]]
-------------------------------------------------------------------
-------------------------------------------------------------------
A chéo trội hàng.
Phương pháp Gauss-Seidel đánh giá tiên nghiệm kết thúc sau 44 bước lặp.
Ma trận nghịch đảo của A:
[[ 1.00244304e-03 -3.45959045e-06  3.54674684e-07 ...  1.21327937e-06
  -5.62692929e-07 -3.59949931e-06]
 [ 1.51521957e-06  1.00250176e-03 -5.51422576e-06 ... -6.76570062e-06
  -4.73958392e-06 -7.68281932e-06]
 [-7.45077977e-06 -5.33089203e-06  1.00242208e-03 ... -6.63351918e-06
  -6.70475461e-06 -1.39146291e-06]
 ...
 [-4.56705440e-06 -7.50822552e-06 -6.75048536e-06 ...  1.00249977e-03
  -3.55687120e-06 -6.78505677e-06]
 [-1.49983455e-06 -7.57941251e-06 -1.49847503e-06 ... -2.59865121e-06
   1.00243644e-03 -5.80656669e-06]
 [ 5.59454719e-07 -6.67856208e-06 -5.49126163e-06 ... -3.73954084e-06
   1.49756513e-06  1.00245737e-03]]
-------------------------------------------------------------------
Kiểm tra nhân ngược:
[[ 1.00000000e+00  1.30104261e-18  2.00323292e-19 ... -1.34254722e-19
   1.98301001e-18 -4.31944451e-18]
 [-1.86347248e-19  1.00000000e+00 -2.89515861e-18 ... -4.24702320e-18
  -1.20786898e-18 -1.61444480e-18]
 [ 2.44263126e-18 -8.72443936e-19  1.00000000e+00 ...  2.85598334e-18
   2.84179554e-18  1.99328028e-18]
 ...
 [ 1.40268656e-18 -5.42101086e-19 -3.77776694e-18 ...  1.00000000e+00
   5.50994932e-19  1.43317975e-18]
 [ 1.58903381e-18  1.18449087e-17 -1.88718941e-18 ...  1.38913403e-19
   1.00000000e+00  1.72794721e-19]
 [-2.16840434e-19  1.73472348e-18 -3.46944695e-18 ... -8.67361738e-19
   8.02563718e-19  1.00000000e+00]]
[Finished in 1.7s]
```

```
-------------------------------------------------------------------
Ma trận A:
[[999.   5.   1. ...   0.   2.   5.]
 [  0. 999.   7. ...   8.   6.   9.]
 [  9.   7. 999. ...   8.   8.   3.]
 ...
 [  6.   9.   8. ... 999.   5.   8.]
 [  3.   9.   3. ...   4. 999.   7.]
 [  1.   8.   7. ...   5.   0. 999.]]
-------------------------------------------------------------------
-------------------------------------------------------------------
A chéo trội hàng.
Phương pháp Gauss-Seidel đánh giá hậu nghiệm kết thúc sau 14 bước lặp.
Ma trận nghịch đảo của A:
[[ 1.00244304e-03 -3.45959045e-06  3.54674692e-07 ...  1.21327932e-06
  -5.62692955e-07 -3.59949934e-06]
 [ 1.51521958e-06  1.00250176e-03 -5.51422575e-06 ... -6.76570064e-06
  -4.73958393e-06 -7.68281934e-06]
 [-7.45077977e-06 -5.33089202e-06  1.00242208e-03 ... -6.63351919e-06
  -6.70475461e-06 -1.39146292e-06]
 ...
 [-4.56705440e-06 -7.50822552e-06 -6.75048536e-06 ...  1.00249977e-03
  -3.55687120e-06 -6.78505677e-06]
 [-1.49983455e-06 -7.57941251e-06 -1.49847504e-06 ... -2.59865121e-06
   1.00243644e-03 -5.80656670e-06]
 [ 5.59454718e-07 -6.67856208e-06 -5.49126164e-06 ... -3.73954085e-06
   1.49756513e-06  1.00245737e-03]]
-------------------------------------------------------------------
Kiểm tra nhân ngược:
[[ 1.00000000e+00  1.28605243e-11  1.70733466e-11 ... -3.91962820e-11
  -1.71544720e-11 -2.61929731e-11]
 [ 1.98132448e-11  1.00000000e+00  1.83847316e-11 ... -1.61393026e-11
  -5.28412737e-13 -7.83745338e-12]
 [ 1.96577631e-11  1.60521097e-11  1.00000000e+00 ... -2.91846850e-12
   8.34279760e-12  2.38892937e-12]
 ...
 [-2.71687395e-12 -2.53473281e-12 -2.35717011e-12 ...  1.00000000e+00
  -5.64024562e-12 -5.47412722e-12]
 [-2.71246362e-12 -2.55026048e-12 -2.34535568e-12 ... -6.62450365e-12
   1.00000000e+00 -5.78478100e-12]
 [-3.05034947e-12 -2.88106344e-12 -2.63216619e-12 ... -7.72483900e-12
  -6.83232280e-12  1.00000000e+00]]
[Finished in 953ms]
```

5 Phân tích và tổng kết các phương pháp

5.1 Nhận xét các ví dụ

- Nhìn chung cả ba phương pháp đều cho thấy sự ổn định khi ma trận ở trạng thái gần suy biến cũng như ma trận cỡ lớn.
- Phương pháp Newton có tốc độ hội tụ rất nhanh so với phương pháp Jacobi và Gauss-Seidel
- Bằng cách sử dụng đánh giá sai số hậu nghiệm, cả hai phương pháp Jacobi và Gauss-Seidel đều được cắt giảm số bước tính toán đáng kể so với khi sử dụng đánh giá sai số tiên nghiệm, trong một số trường hợp cắt giảm hàng ngàn bước lặp.
- Phương pháp Gauss-Seidel hội tụ nhanh hơn phương pháp Jacobi. Điều này là hoàn toàn phù hợp với lí thuyết do phương pháp lặp Gauss-Seidel là một cải tiến của phương pháp lặp Jacobi.

5.2 Ưu, nhược diểm của các phương pháp

5.2.1 Ưu điểm

- Các phương pháp giải gần đúng ma trận nghịch đảo đều giúp ta kiểm soát và cải thiện sai số tính toán sau một số lần lặp nhất định, đây là điều mà các phương pháp giải đúng không làm được.
- Với phương pháp Newton, công thức lặp hội tụ rất nhanh và không yêu cầu nhiều với ma trận đầu vào, thuật toán lại đơn giản, dễ nhớ.

5.2.2 Nhược điểm

- Với phương pháp Newton, ta cần tính chuẩn 2 của ma trận, ở đây ta đã sử dụng thư viện numpy của Python.
- Hai phương pháp Jacobi và Gauss-Seidel yêu cầu ma trận đầu vào phải chéo trội.

6 Hướng dẫn sử dụng chương trình

Lưu ý: Cài đặt thư viện numpy trước khi sử dụng các chương trình

6.1 Phương pháp Newton

Đặt các file sau trong cùng một thư mục:

- **Chương trình chính:** newton.py
- **File dữ liệu đầu vào:** input.txt

Các bước sử dụng:

- **Bước 1:** Nhập sai số vào dòng đầu của file input.txt.
- **Bước 2:** Nhập ma trận cần tìm nghịch đảo vào các dòng sau, lưu ý cần phân tách các giá trị trong cùng một hàng của ma trận bởi dấu cách và xuống dòng khi sang hàng tiếp theo.
- **Bước 3:** Chạy chương trình newton.py, chương trình sẽ đưa ra số lần lặp, ma trận nghịch đảo và kết quả kiểm tra nhân ngược. Trường hợp ma trận không khả nghịch, chương trình sẽ báo lỗi và đưa ra ma trận NaN.

6.2 Phương pháp Jacobi

Đặt các file sau trong cùng một thư mục:

- **Chương trình chính:** jacobi.py
- **File dữ liệu đầu vào:** input.txt

Các bước sử dụng:

- **Bước 1:** Nhập chế độ đánh giá vào dòng đầu của file input.txt (Nhập "1" tương ứng với chế độ đánh giá tiên nghiệm, nếu không chương trình sẽ chạy theo chế độ đánh giá hậu nghiệm)
- **Bước 2:** Nhập sai số vào dòng thứ hai của file input.txt
- **Bước 3:** Nhập ma trận cần tìm nghịch đảo vào các dòng sau, lưu ý cần phân tách các giá trị trong cùng một hàng của ma trận bởi dấu cách và xuống dòng khi sang hàng tiếp theo.
- **Bước 4:** Chạy chương trình jacobi.py, chương trình sẽ đưa ra số lần lặp, ma trận nghịch đảo và kết quả kiểm tra nhân ngược. Trường hợp ma trận không chéo trội, chương trình sẽ báo lỗi và đưa ra ma trận NaN.

6.3 Phương pháp Gauss-Seidel

Đặt các file sau trong cùng một thư mục:

- **Chương trình chính:** gauss_seidel.py
- **File dữ liệu đầu vào:** input.txt

Các bước sử dụng:

- **Bước 1:** Nhập chế độ đánh giá vào dòng đầu của file input.txt (Nhập "1" tương ứng với chế độ đánh giá tiên nghiệm, nếu không chương trình sẽ chạy theo chế độ đánh giá hậu nghiệm)
- **Bước 2:** Nhập sai số vào dòng thứ hai của file input.txt
- **Bước 3:** Nhập ma trận cần tìm nghịch đảo vào các dòng sau, lưu ý cần phân tách các giá trị trong cùng một hàng của ma trận bởi dấu cách và xuống dòng khi sang hàng tiếp theo.
- **Bước 4:** Chạy chương trình jacobi.py, chương trình sẽ đưa ra số lần lặp, ma trận nghịch đảo và kết quả kiểm tra nhân ngược. Trường hợp ma trận không chéo trội, chương trình sẽ báo lỗi và đưa ra ma trận NaN.

Tài liệu

[1] J. Douglas (Douglas Faires) Faires, Richard L.Burden - *Numerical methods* (2003)

[2] Lê Trọng Vinh - *Giáo trình giải tích số* (2007)

[3] Phạm Kỳ Anh - *Giải tích số* (1996)

[4] Jaan Kiusalaas - *Numerical Methods in Engineering with Python* (2010)

[5] Adi Ben-Israel - *A Note on an Iteractive Method for Generalized Inversion of Matrices* (1966)

[6] Samuel Daniel Conte - *Elementary numerical analysis, an algorithmic approach* (1980)