

Các phương pháp tìm gần đúng ma trận nghịch đảo

Bùi Tiến Thành - MSSV 20190081
Chu Thị Ngân - MSSV 20195904

CTTN Toán tin K64

Tháng Mười một, 2020



Mục lục

- 1 Đặt vấn đề
 - Tại sao phải giải gần đúng?
 - Các phương pháp đưa ra
- 2 Phương pháp Newton
 - Ý tưởng và công thức lặp
 - Điều kiện hội tụ và công thức sai số
 - Thuật toán và chương trình
- 3 Giới thiệu các phương pháp tìm nghịch đảo ma trận chéo trội
 - Ý tưởng chung
 - Các phương pháp
- 4 Phương pháp lặp Jacobi
 - Công thức lặp và điều kiện hội tụ
 - Công thức sai số
 - Thuật toán và chương trình
- 5 Phương pháp lặp Gauss-Seidel
 - Công thức lặp và điều kiện hội tụ
 - Công thức sai số

Tại sao phải giải gần đúng?

Các phương pháp giải đúng:

- Bắt buộc phải duyệt toàn bộ các phần tử của ma trận và phải lưu toàn bộ ma trận trong bộ nhớ, đặc biệt với các thuật toán đệ quy
- **Không thể kiểm soát sai số tính toán** do giới hạn tính toán của máy tính

⇒ **Giải pháp?**



Các phương pháp đưa ra

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



Mục lục

- 1 Đặt vấn đề
 - Tại sao phải giải gần đúng?
 - Các phương pháp đưa ra
- 2 Phương pháp Newton
 - Ý tưởng và công thức lặp
 - Điều kiện hội tụ và công thức sai số
 - Thuật toán và chương trình
- 3 Giới thiệu các phương pháp tìm nghịch đảo ma trận chéo trội
 - Ý tưởng chung
 - Các phương pháp
- 4 Phương pháp lặp Jacobi
 - Công thức lặp và điều kiện hội tụ
 - Công thức sai số
 - Thuật toán và chương trình
- 5 Phương pháp lặp Gauss-Seidel
 - Công thức lặp và điều kiện hội tụ
 - Công thức sai số

Ý tưởng và công thức lặp

Ý tưởng

Từ phương pháp Newton cho $ax = 1$:

$$x_{n+1} = x_n * (2 - ax_n) \text{ với } n \in \mathbb{N}$$

Áp dụng vào ma trận với E là ma trận đơn vị cùng cấp:

Công thức lặp

$$X_{k+1} = X_k(2E - AX_k) \text{ với } k \in \mathbb{N}$$

Điều kiện hội tụ và công thức sai số

Điều kiện hội tụ

$$\|A^{-1} - X_k\| \leq \|A^{-1}\| \|E - AX_0\|^{2^k} \xrightarrow{k \rightarrow \infty} 0 \text{ nếu } \|E - AX_0\| < 1$$

Công thức sai số

Khi $\|E - AX_0\| \leq q < 1$:

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



Input, output

- Input: Ma trận A , xấp xỉ đầu X_0 và sai số ε . Giả thiết coi như các điều kiện hội tụ thỏa mãn.
- Output: Ma trận xấp xỉ A^{-1}
- Code:



<https://github.com/bu1th4nh/TALENTED-K64MI/blob/master/MI3040/report-code/newton.py>



Mã giả

Algorithm 1: Phương pháp Newton tìm ma trận nghịch đảo**Input:** Ma trận A, X_0, ε **Output:** Ma trận $X^* = A^{-1}$ là ma trận nghịch đảo**begin** Nhập A, X_0, ε ; $q \leftarrow \|E - AX_0\|$; $q_{2k} \leftarrow q$; $X \leftarrow X_0$; **while** $\frac{\|X_0\| * q_{2k}}{1 - q} > \varepsilon$ **do** $X \leftarrow X(2E - AX)$; $q_{2k} \leftarrow q_{2k}^2$; **end** Đưa ra X chính là ma trận nghịch đảo;**end**

Mục lục

- 1 Đặt vấn đề
 - Tại sao phải giải gần đúng?
 - Các phương pháp đưa ra
- 2 Phương pháp Newton
 - Ý tưởng và công thức lặp
 - Điều kiện hội tụ và công thức sai số
 - Thuật toán và chương trình
- 3 Giới thiệu các phương pháp tìm nghịch đảo ma trận chéo trội
 - Ý tưởng chung
 - Các phương pháp
- 4 Phương pháp lặp Jacobi
 - Công thức lặp và điều kiện hội tụ
 - Công thức sai số
 - Thuật toán và chương trình
- 5 Phương pháp lặp Gauss-Seidel
 - Công thức lặp và điều kiện hội tụ
 - Công thức sai số

Ý tưởng chung

Ý tưởng

Đưa phương trình $AX = E$ về dạng công thức lặp sau:

$$X_{k+1} = BX_k + D$$

Ta cũng chứng minh được nếu $\|B\| < 1$ thì với giá trị xấp xỉ đầu bất kì thì X luôn hội tụ về nghiệm (Phương pháp lặp đơn).

→ Vậy làm thế nào để $\|B\| < 1$?



Các phương pháp được lựa chọn

- Phương pháp lặp Jacobi (dựa trên phương pháp lặp đơn)
- Phương pháp lặp Gauss - Seidel (dựa trên phương pháp lặp Siedel)



Mục lục

- 1 Đặt vấn đề
 - Tại sao phải giải gần đúng?
 - Các phương pháp đưa ra
- 2 Phương pháp Newton
 - Ý tưởng và công thức lặp
 - Điều kiện hội tụ và công thức sai số
 - Thuật toán và chương trình
- 3 Giới thiệu các phương pháp tìm nghịch đảo ma trận chéo trội
 - Ý tưởng chung
 - Các phương pháp
- 4 Phương pháp lặp Jacobi
 - Công thức lặp và điều kiện hội tụ
 - Công thức sai số
 - Thuật toán và chương trình
- 5 Phương pháp lặp Gauss-Seidel
 - Công thức lặp và điều kiện hội tụ
 - Công thức sai số

Công thức lặp và điều kiện hội tụ

Đặt: $T = \text{diag}\{\frac{1}{A_{11}}, \frac{1}{A_{22}}, \dots, \frac{1}{A_{nn}}\}$ là ma trận đường chéo cấp n , ta có:

Công thức lặp

$$X_{k+1} = (E - TA)X_k + T$$

Điều kiện hội tụ

Để $\|E - TA\| < 1$ thì ma trận A phải chéo trội, nghĩa là:

$$\sum_{j=1, j \neq i}^n |A_{ij}| < |A_{ii}| \quad \forall i = \overline{1, n} \text{ chéo trội hàng}$$

$$\text{hoặc} \quad \sum_{i=1, i \neq j}^n |A_{ij}| < |A_{jj}| \quad \forall j = \overline{1, n} \text{ chéo trội cột}$$

Công thức sai số

Cho $B = E - TA, B_1 = E - AT$

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: Cho $\lambda = \frac{\max |A_{ii}|}{\min |A_{ii}|}, \|B_1\|_1 \leq q < 1$

$$\|X_k - X^*\|_1 \leq \lambda \frac{q}{1-q} \|X_k - X_{k-1}\|_1$$

$$\|X_k - X^*\|_1 \leq \lambda \frac{q^k}{1-q} \|X_1 - X_0\|_1$$

Thuật toán chính

Algorithm 2: Phương pháp Jacobi tìm ma trận nghịch đảo

Input: Ma trận A chéo trội, sai số ε

Output: Ma trận $X^* = A^{-1}$ là ma trận nghịch đảo

begin

Nhập A, ε ;

$p \leftarrow \text{checkDomination}(A)$;

$T \leftarrow \text{diag}\{\frac{1}{A_{11}}, \frac{1}{A_{22}}, \dots, \frac{1}{A_{nn}}\}$;

$B \leftarrow E - TA, B_1 \leftarrow E - AT$;

$q \leftarrow \text{getNorm}(B, B_1, p)$;

$\lambda \leftarrow \text{getLambda}(A, p)$;

$X^* \leftarrow \text{iterate}(X_0 \leftarrow A, B, T, q, \lambda, p, \varepsilon)$;

Trả về X^* là ma trận nghịch đảo;

end

Chuyển sang phần code



Function checkDomination(A)**Input:** Ma trận A **Output:** Giá trị kiểm tra p bằng 1 nếu A chéo trội hàng, -1 nếu A chéo trội cột, 0 khi A không chéo trội**begin** $row_dom \leftarrow 1, col_dom \leftarrow 1;$ **for** $i = 1$ **to** n **do****if** $\sum_{j=1, j \neq i}^n |A_{ij}| \geq |A_{ii}|$ **then** $row_dom \leftarrow 0;$ **if** $\sum_{j=1, j \neq i}^n |A_{ji}| \geq |A_{ii}|$ **then** $col_dom \leftarrow 0;$ **end****if** $row_dom = 1$ **then return** 1;**if** $col_dom = 1$ **then return** -1;**return** 0 ;**end**

Function getNorm(A, A_1, p)**Input:** Ma trận A , giá trị kiểm tra p **Output:** $\|A\|_\infty$ nếu $p = 1$, $\|A_1\|_1$ nếu $p = -1$ **begin**

if $p = 1$ **then return** $\|A\|_\infty$;

if $p = -1$ **then return** $\|A_1\|_1$;

end**Function** getLambda(A, p)**Input:** Ma trận A , giá trị kiểm tra p **Output:** $\lambda = 1$ nếu $p = 1$, $\lambda = \frac{\max |A_{ii}|}{\min |A_{ii}|}$ nếu $p = -1$ **begin** **if** $p = 1$ **then return** 1; $\max_A \leftarrow |A_{11}|$; $\min_A \leftarrow |A_{11}|$; **for** $i = 1$ **to** n **do** $\max_A = \max(\max_A, |A_{ii}|)$; $\min_A = \min(\min_A, |A_{ii}|)$;

Lặp - Đánh giá tiên nghiệm

Function $\text{iterate}(X_0, B, T, q, \lambda, p, \varepsilon)$

Input: Ma trận xấp xỉ đầu X_0 , B , T , hệ số q , λ , giá trị kiểm tra p và sai số ε

Output: X^* là ma trận nghịch đảo theo đánh giá tiên nghiệm
begin

$qk \leftarrow 1, X \leftarrow X_0 ;$

$\text{predecessor_norm} \leftarrow \text{getNorm}((BX_0 + T) - X_0, p) ;$

while $\frac{\lambda * qk * \text{predecessor_norm}}{1 - q} > \varepsilon$ **do**

$X \leftarrow BX + T ;$

$qk \leftarrow qk * q ;$

end

return X

end



Trở về thuật toán chính

Lặp - Đánh giá hậu nghiệm

Function `iterate($X_0, B, T, q, \lambda, p, \epsilon$)`

Input: Ma trận xấp xỉ đầu X_0 , B , T , hệ số q , λ , giá trị kiểm tra p và sai số ϵ

Output: X^* là ma trận nghịch đảo theo đánh giá hậu nghiệm
begin

$old_X \leftarrow X_0;$

$new_X \leftarrow BX_0 + T;$

while $\frac{\lambda * q * getNorm(new_X - old_X, p)}{1 - q} > \epsilon$ **do**

$old_X \leftarrow new_X;$

$new_X \leftarrow B * old_X + T;$

end

return new_X

end



[Trở về thuật toán chính](#)

Chương trình



`https://github.com/bu1th4nh/TALENTED-K64MI/blob/master/MI3040/report-code/jacobian.py`



Mục lục

- 1 Đặt vấn đề
 - Tại sao phải giải gần đúng?
 - Các phương pháp đưa ra
- 2 Phương pháp Newton
 - Ý tưởng và công thức lặp
 - Điều kiện hội tụ và công thức sai số
 - Thuật toán và chương trình
- 3 Giới thiệu các phương pháp tìm nghịch đảo ma trận chéo trội
 - Ý tưởng chung
 - Các phương pháp
- 4 Phương pháp lặp Jacobi
 - Công thức lặp và điều kiện hội tụ
 - Công thức sai số
 - Thuật toán và chương trình
- 5 Phương pháp lặp Gauss-Seidel
 - Công thức lặp và điều kiện hội tụ
 - Công thức sai số

Công thức lặp và điều kiện hội tụ

Đặt $B = E - TA$ với $T = \text{diag}\{\frac{1}{A_{11}}, \frac{1}{A_{22}}, \dots, \frac{1}{A_{nn}}\}$ tương tự phương pháp Jacobi

Công thức lặp - Phương pháp Gauss-Seidel nguyên bản [4]

$$X_i^{(k+1)} = \sum_{j=1}^{i-1} B_{ij} X_j^{(k+1)} + \sum_{j=i+1}^n B_{ij} X_j^{(k)} + T_i$$

với mọi $i = \overline{1, n}$ và A_i là dòng i của ma trận A

Mở rộng: Hệ số điều chỉnh cho phương pháp Gauss-Seidel [1]

$$X_i^{k+1} = (1 - \omega) X_i^k + \omega \left[\sum_{j=1}^{i-1} B_{ij} X_j^{(k+1)} + \sum_{j=i+1}^n B_{ij} X_j^{(k)} + T_i \right]$$

với ω là hệ số điều chỉnh (relaxation factor)

Điều kiện hội tụ: Tương tự phương pháp **Jacobi**, tức là A phải chéo trội

Trường hợp chéo trội hàng

Cho:

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

Công thức sai số [4]

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

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

Trường hợp chéo trội cột

Cho:

$$q = \max_{1 \leq i \leq n} \frac{\sum_{j=1}^i |B_{ji}|}{1 - \sum_{j=i+1}^n |B_{ji}|} \leq \|B\|_1 < 1$$

$$S = \max_{1 \leq i \leq n} \sum_{j=i+1}^n |B_{ji}|$$

Công thức sai số[4]

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

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

Thuật toán chính

Algorithm 3: Phương pháp Gauss-Seidel tìm ma trận nghịch đảo

Input: Ma trận A chéo trội, sai số ε , hệ số điều chỉnh ω

Output: Ma trận $X^* = A^{-1}$ là ma trận nghịch đảo

begin

Nhập A, ε ;

$p \leftarrow \text{checkDomination}(A)$;

$T \leftarrow \text{diag}\{\frac{1}{A_{11}}, \frac{1}{A_{22}}, \dots, \frac{1}{A_{nn}}\}$;

$B \leftarrow E - TA$;

$S \leftarrow \text{getSCoeff}(B, p)$;

$q \leftarrow \text{getqCoeff}(B, p)$;

$X^* \leftarrow \text{iterate}(X_0 \leftarrow A, B, T, S, q, p, \omega, \varepsilon)$;

Trả về X^* là ma trận nghịch đảo;

end

Chuyển sang phần code



Function getSCoeff(A, p)

Input: Ma trận A , giá trị kiểm tra p

Output: $S = 0$ nếu $p = 1$, $\max_{1 \leq i \leq n} \sum_{j=i+1}^n |A_{ji}|$ nếu $p = -1$

begin

if $p = 1$ **then return** 0 ;

$S \leftarrow 0$;

for $i = 1$ **to** n **do** $S \leftarrow \max(S, \sum_{j=i+1}^n |A_{ji}|)$;

return S

end

[Trở về thuật toán chính](#)



Function getqCoeff(A, p)**Input:** Ma trận A , giá trị kiểm tra p

Output: $q = \max_{1 \leq i \leq n} \frac{\sum_{j=i}^n |A_{ij}|}{1 - \sum_{j=1}^{i-1} |A_{ij}|}$ nếu $p = 1$, $\max_{1 \leq i \leq n} \frac{\sum_{j=1}^i |A_{ji}|}{1 - \sum_{j=i+1}^n |A_{ji}|}$ nếu $p = -1$

begin $q \leftarrow 0;$ **for** $i = 1$ **to** n **do** $Q1 \leftarrow 0, Q2 \leftarrow 0;$ **if** $p = 1$ **then** $Q1 \leftarrow \sum_{j=i}^n |A_{ij}|, Q2 \leftarrow \sum_{j=1}^{i-1} |A_{ij}|;$ **else** $Q1 \leftarrow \sum_{j=1}^i |A_{ji}|, Q2 \leftarrow \sum_{j=i+1}^n |A_{ji}|;$ $q \leftarrow \max(q, \frac{Q1}{1-Q2});$ **end****return** q **end**

Lặp - Đánh giá tiên nghiệm

Function `iterate($X_0, B, T, S, q, p, \omega, \varepsilon$)`

Input: Ma trận xấp xỉ đầu X_0, B, T , hệ số S, q , giá trị kiểm tra p , hệ số điều chỉnh ω và sai số ε

Output: X^* là ma trận nghịch đảo theo đánh giá tiên nghiệm
begin

$qk \leftarrow 1;$

$X \leftarrow X_0;$

$X_1 \leftarrow \text{nextIteration}(X_0, B, T, \omega);$

$\text{predecessor_norm} \leftarrow \text{getNorm}(X_1 - X_0, p);$

while $\frac{qk * \text{predecessor_norm}}{(1-q)*(1-S)} > \varepsilon$ **do**

$X \leftarrow \text{nextIteration}(X, B, T, \omega);$

$qk \leftarrow qk * q;$

end

return X

end



Lặp - Đánh giá hậu nghiệm

Function `iterate($X_0, B, T, S, q, p, \omega, \varepsilon$)`

Input: Ma trận xấp xỉ đầu X_0, B, T , hệ số S, q , giá trị kiểm tra p , hệ số điều chỉnh ω và sai số ε

Output: X^* là ma trận nghịch đảo theo đánh giá hậu nghiệm
begin

$old_X \leftarrow X_0;$

$new_X \leftarrow \text{nextIteration}(X_0, B, T, \omega);$

while $\frac{\lambda * q * \text{getNorm}(new_X - old_X, p)}{1 - q} > \varepsilon$ **do**

$old_X \leftarrow new_X;$

$new_X \leftarrow \text{nextIteration}(X, B, T, \omega);$

end

return new_X

end



[Trở về thuật toán chính](#)

Function nextIteration(*old_X*, *B*, *T*, ω)

Input: Ma trận *old_X*, *B*, *T*, và hệ số ω

Output: Đưa ra ma trận tiếp theo thu được từ ma trận ban đầu *old_X* theo công thức lặp Gauss-Seidel

begin

new_X \leftarrow Ma trận không cấp *n*;

for *i* = 1 **to** *n* **do**

$$new_X_i = \sum_{j=1}^{i-1} B_{ij} * new_X_j + \sum_{j=i+1}^n B_{ij} * old_X_j + T_i;$$

end

return $(1 - \omega) * old_X + \omega * new_X$

end

[Trở về thuật toán chính](#)

[Trở về tiến trình lặp](#)



Chương trình



`https://github.com/bu1th4nh/TALENTED-K64MI/blob/master/MI3040/report-code/gauss_seidel.py`



Mục lục

- 1 Đặt vấn đề
 - Tại sao phải giải gần đúng?
 - Các phương pháp đưa ra
- 2 Phương pháp Newton
 - Ý tưởng và công thức lặp
 - Điều kiện hội tụ và công thức sai số
 - Thuật toán và chương trình
- 3 Giới thiệu các phương pháp tìm nghịch đảo ma trận chéo trội
 - Ý tưởng chung
 - Các phương pháp
- 4 Phương pháp lặp Jacobi
 - Công thức lặp và điều kiện hội tụ
 - Công thức sai số
 - Thuật toán và chương trình
- 5 Phương pháp lặp Gauss-Seidel
 - Công thức lặp và điều kiện hội tụ
 - Công thức sai số

Ưu, nhược điểm

Phương pháp	Newton	Lặp Jacobi	Lặp Gauss-Seidel
Ưu điểm	Kiểm soát được sai số tính toán, sai số được cải thiện sau n Tốc độ hội tụ nhanh trong một số trường hợp		
Nhược điểm	Khó tìm giá trị xấp xỉ đầu X_0	Yêu cầu ma trận phải chéo trội	

→ Tìm giá trị xấp xỉ đầu cho X_0 trong phương pháp Newton?



Ưu, nhược điểm

Phương pháp	Newton	Lặp Jacobi	Lặp Gauss-Seidel
Ưu điểm	Kiểm soát được sai số tính toán, sai số được cải thiện sau n Tốc độ hội tụ nhanh trong một số trường hợp		
Nhược điểm	Khó tìm giá trị xấp xỉ đầu X_0	Yêu cầu ma trận phải chéo trội	

→ Tìm giá trị xấp xỉ đầu cho X_0 trong phương pháp Newton?

- Đặt $X_0 = \frac{A}{\|A\|_1 \|A\|_\infty}$ [3]
- Sử dụng kết quả của các phương pháp tính trực tiếp ma trận nghịch đảo làm xấp xỉ đầu



Câu hỏi?



Tài liệu tham khảo

- [1] Richard L. Burden J. Douglas Faires. *Numerical Methods, 4th Edition*. Brooks / Cole, Cengage Learning, 2013, 2003, 1998.
- [2] The Pennsylvania State University Jaan Kiusalaas. *Numerical Methods in Engineering With MATLAB*. Cambridge University Press, 2005.
- [3] Thomas E. Phipps Jr. "The inversion of large matrices: The Pan and Reif algorithm provides a solution". In: *Byte Magazine* 11.04 (4-1986).
- [4] Lê Trọng Vinh. *Giáo trình Giải tích số*. NXB Khoa học và kỹ thuật, 2007.
- [5] Hà Thị Ngọc Yến. *Phương pháp lặp Jacobi*. 2019.



