Chương 1

Giải hệ phương trình

Chúng ta có 2 loại hệ phương trình:

- Hệ phương trình tuyến tính
- Hệ phương trình phi tuyến

Chúng ta đã biết các phương pháp giải trực tiếp và gián tiếp hệ phương trình tuyến tính:

- Các phương pháp giải trực tiếp:
  - Phương pháp Cramer
  - Phương pháp thế
  - Phương pháp sử dụng ma trận nghịch đảo
  - Phương pháp khử Gauss, Gauss-Jordan
- Các phương pháp giải gián tiếp
  - Phương pháp lặp đơn
  - Phương pháp lặp Seidel
- Các phương pháp tìm trị riêng và véc tơ riêng của ma trận

1.1 Đặt bài toán và phương pháp giải

Hệ phương trình tuyến tính n phương trình, n ẩn x_1, x_2, ..., x_n là tập n phương trình E_1, E_2, ..., E_n dạng

$$\begin{cases} E_1 : a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ E_2 : a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \ldots \\ E_n : a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \end{cases}$$

với các hệ số a_{jk} và b_j đã biết. Hệ được gọi là thuần nhất nếu các b_j bằng 0, trong trường hợp ngược lại được gọi là hệ không thuần nhất.

Dùng cách biểu diễn ma trận, ta có thể viết hệ (1.1) dưới dạng:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{1.1}$$

Ở đây, ma trận hệ số $A = [a_{jk}]$ là ma trận vuông cấp n, x và b là các véc tơ cột:

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Ma trận $\tilde{\boldsymbol{A}}$ sau được gọi là ma trận mở rộng (augmented matrix) của hệ (1.1):

$$\tilde{\boldsymbol{A}} = [\boldsymbol{A}, \boldsymbol{b}] = \left[\begin{array}{cccc:c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} & b_n \end{array}\right]$$

Nghiệm của (1.1) là bộ số x_1, x_2, ..., x_n thỏa mãn tất cả phương trình của hệ.

Véc tơ nghiệm của (1.1) là $\boldsymbol{x}$ mà các thành phần của nó lập thành một nghiệm của (1.1).

Hệ phương trình (1.1) có thể giải được bằng

- *phương pháp trực tiếp* (phương pháp khử Gauss, ...), hoặc
- *phương pháp gián tiếp* hay *phương pháp lặp*

1.2 Phương pháp khử Gauss

Chúng ta đã biết phương pháp sử dụng định thức để giải hệ phương trình (1.1), đó là *phương pháp Cramer*. Ở đây ta xét *phương pháp khử Gauss* để giải hệ phương trình tuyến tính.

1.2.1 Phương pháp khử Gauss

Phương pháp. Phương pháp khử Gauss (Gaussian elimination)

Phương pháp khử Gauss là phép giải trực tiếp được thực hiện bằng quá trình khử liên tiếp các ẩn để từng bước đưa ma trận mở rộng $\tilde{\boldsymbol{A}}$ về dạng tam giác trên (upper triangular matrix, hay dạng bậc thang).

Phương pháp này gồm hai phần thực hiện lần lượt như sau:

1. Quá trình thuận (forward elimination)*: Dùng các phép biến đổi sơ cấp về hàng để đưa $\tilde{\boldsymbol{A}}$ về dạng tam giác trên:*

(a) *Khử x_1 từ $E_{\geq 2}$ bằng cách:*

$$E_j \coloneqq E_j - \frac{a_{j1}}{a_{11}} E_1 \, \forall j \in [2, n] \tag{1.2}$$

a_{11} *gọi là* phần tử trục xoay (pivot), E_1 *gọi là phương trình chính (pivot equation).*

(b) *Khử x_i từ $E_{>i}$ bằng cách tương tự như trên, cuối cùng thu được dạng tam giác trên.*

2. Quá trình ngược (back substitution)*: Giải $\boldsymbol{x}$ từ cuối lên:*

(a) *Giải x_n từ E_n, giải tiếp được x_{n-1} do đã biết x_n.*

(b) *Tương tự giải được đến x_1, cuối cùng thu được nghiệm $\boldsymbol{x}$.*

Xem xét kĩ hơn, ta thấy (1.2) có một chi tiết nhạy cảm là phép chia. Nếu $a_{11} = 0$, phương pháp không thể thực hiện được, cho dù thực tế hệ phương trình có thể có nghiệm. Một cách xử lí trường hợp này là *đổi vị trí hàng*, như trong hai ví dụ sau:

Ví dụ 1.1. *Hãy giải hệ phương trình:*

$$\begin{cases} 8x_2 + 2x_3 &= -7 & (E_1) \\ 3x_1 + 5x_2 + 2x_3 &= 8 & (E_2) \\ 6x_1 + 2x_2 + 8x_3 &= 26 & (E_3) \end{cases}$$

Chúng ta xoay trục từ E_1 , nhưng do E_1 không có ẩn x_1, trong khi đó hệ số của x_1 trong phương trình E_3 là lớn nhất. Vì vậy ta đổi chỗ E_1 và E_3 cho nhau.

Tới đây ta có ma trận mở rộng như sau:

$$\tilde{\boldsymbol{A}} = \left[\begin{array}{ccc:c} 6 & 2 & 8 & 26 \\ 3 & 5 & 2 & 8 \\ & 8 & 2 & -7 \end{array}\right]$$

Khử x_1 được:

$$\left[\begin{array}{ccc:c} 6 & 2 & 8 & 26 \\ & 4 & -2 & -5 \\ & 8 & 2 & -7 \end{array}\right]$$

Khử x_2 được:

$$\left[\begin{array}{ccc:c} 6 & 2 & 8 & 26 \\ & 4 & -2 & -5 \\ & & 6 & 3 \end{array}\right]$$

Vậy ta giải được $x_3 = 0{,}5$, $x_2 = -1$, $x_1 = 4$.

Ví dụ 1.2. *Hãy giải hệ phương trình:*

$$\begin{cases} x_1 - x_2 + 2x_3 - x_4 &= -8 & (E_1) \\ 2x_1 - 2x_2 + 3x_3 - 3x_4 &= -20 & (E_2) \\ x_1 + x_2 + x_3 &= -2 & (E_3) \\ x_1 - x_2 + 4x_3 + 3x_4 &= 4 & (E_4) \end{cases}$$

Ta có ma trận mở rộng như sau:

$$\tilde{\boldsymbol{A}} = \left[\begin{array}{rrrr:r} 1 & -1 & 2 & -1 & -8 \\ 2 & -2 & 3 & -3 & -20 \\ 1 & 1 & 1 & 0 & -2 \\ 1 & -1 & 4 & 3 & 4 \end{array}\right]$$

Phương pháp khử Gauss còn có một số biến thể:

- Phương pháp Gauss-Jordan: đưa $\boldsymbol{A}$ về dạng đường chéo thay vì dạng tam giác trên.
- Phương pháp Doolittle, phương pháp Crout, phương pháp Cholesky: đều dựa trên phân tích LU (LU factorization), sẽ giới thiệu ở phần sau.

1.2.2 Độ phức tạp

Ta phân tích độ phức tạp của phương pháp khử Gauss:

- Có tổng cộng $n-1$ bước khử.
- Ở bước k, ta khử x_k trong các phương trình $E_{>k}$, tổng cộng là $n-k$ phương trình.
- Trong mỗi phương trình:
  - Có 1 phép chia: ví dụ $\frac{a_{i1}}{a_1 1}$ trong (1.2)
  - Có $n-k+1$ phép nhân: ví dụ $\frac{a_{i1}}{a_1 1} E_1$ trong (1.2)
  - Có $n-k+1$ phép trừ: ví dụ chính (1.2)

Do đó, tổng số phép tính của phương pháp này là:

$$\begin{aligned} C(n) &= \sum_{k=1}^{n-1}(n-k) + 2\sum_{k=1}^{n-1}(n-k)(n-k+1) \\ &= \sum_{s=1}^{n-1} s + 2\sum_{k=1}^{n-1} s(s+1)\,(\text{with } s = n-k) \\ &= \frac{1}{2}n(n-1) + \frac{2}{3}n(n^2-1) \\ &\approx \frac{2}{3}n^3 \end{aligned}$$

Ta nói rằng phương pháp khử Gauss có độ phức tạp $\mathcal{O}(n^3)$.

1.3 Phân tích LU và Ma trận nghịch đảo

Chúng ta tiếp tục thảo luận các phương pháp số giải hệ phương trình tuyến tính n phương trình, n ẩn. Trong phần này, ta xem xét ba phương pháp cải tiến từ phương pháp khử Gauss, cho phép tìm nghiệm nhanh chóng hơn, gồm phương pháp Doolittle, phương pháp Cholevsky, và phương pháp Crout. Cả ba phương pháp đều dựa trên phân tích LU.

Điểm chung của các phương pháp này là đều cố gắng đưa ma trận về tích của các ma trận tam giác trên (Upper triangular matrix) và ma trận tam giác dưới (Lower triangular matrix). Hai dạng ma trận này rất hữu ích vì cho phép tìm $\boldsymbol{x}$ với độ phức tạp $\mathcal{O}(n^2)$; nếu xét theo khía cạnh này, phương pháp khử Gauss khác ở điểm là *trực tiếp* biến đổi về dạng tam giác. Tất nhiên, phần lớn tính toán lại chuyển về việc phân tích ra dạng ma trận đặc biệt này, và với một số dạng ma trận đặc biệt, độ phức tạp có thể thấp hơn $\mathcal{O}(n^3)$.

Ba phương pháp được trình bày trong phần này đều gắn chặt với một kiểu phân tích ra ma trận tam giác. Vì rất dễ để giải nghiệm từ dạng tam giác, tên phương pháp vừa chỉ phương pháp phân tích ra dạng tam giác tương ứng, vừa chỉ phương pháp giải hệ tuyến tính từ dạng tam giác đã phân tích.

1.3.1 Phân tích LU & phương pháp Doolittle

Định nghĩa 1. Phân tích LU (LU factorization)

Phân tích LU là việc phân tích ma trận vuông $\boldsymbol{A}$ thành tích của hai ma trận

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$$

trong đó $\boldsymbol{L}$ là ma trận tam giác dưới và $\boldsymbol{U}$ là ma trận tam giác trên.

Một số lưu ý về phân tích LU:

- Không phải mọi ma trận vuông đều có phân tích LU. Tuy nhiên, ta thừa nhận một kết quả quan trọng là mọi ma trận khả nghịch $\boldsymbol{A}_0$ đều có thể sắp xếp lại các hàng để thu được một ma trận $\boldsymbol{A}$ có phân tích LU.

- Có thể có nhiều cách phân tích LU.

Sau đây ta tìm hiểu phương pháp Doolittle để phân tích LU. Ta sẽ xem xét kết quả của phương pháp phân tích LU này với bài toán giải hệ phương trình tuyến tính trước, sau đó mới đi sâu vào công thức toán học của nó.

Phương pháp Doolittle: Dùng dạng LU để giải hệ tuyến tính

Định lí 1.1. *Nếu hệ tuyến tính $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ có thể giải được bằng khử Gauss mà không cần đổi vị trí hàng, thì $\boldsymbol{A}$ có thể phân tích thành tích của ma trận tam* *giác trên* $\boldsymbol{L}$ *và tam giác dưới* $\boldsymbol{U}$ *(tức* $\boldsymbol{A} = \boldsymbol{LU}$*) với dạng sau:*

$$\boldsymbol{U} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \dots & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ & & & a_{nn}^{(n)} \end{bmatrix}, \text{ và } \boldsymbol{L} = \begin{bmatrix} 1 & & & \\ m_{21} & 1 & & \\ \vdots & \ddots & \ddots & \\ m_{n1} & \dots & m_{n,n-1} & 1 \end{bmatrix}$$

trong đó

- a_{ji}^{k} *là hệ số của* x_i *trong phương trình* E_j *tại bước khử thứ* k
- $m_{ji} = \dfrac{a_{ji}^{(i)}}{a_{ii}^{(i)}}$ *là hệ số của* E_j *trong, ví dụ, công thức* (1.2).

$\boldsymbol{U}$ *cũng chính là ma trận* $\boldsymbol{A}$ *thu được sau khi khử Gauss.*

Định lí trên cũng chính là công thức cho phương pháp Doolittle.

Chú ý rằng $\boldsymbol{U}$ là $\boldsymbol{A}$ sau khi khử Gauss, tức ma trận $\tilde{\boldsymbol{A}}$ trong phần 1.2 nhưng bỏ đi cột cuối.

Cần nhắc lại rằng, đến đây, ta vẫn cần giả sử hệ có thể giải được bằng khử Gauss mà *không* cần đổi vị trí hàng. Mở rộng phương pháp Doolittle cho trường hợp cần đổi vị trí hàng không khó. Trước hết, ta cần tìm cách biểu diễn việc tráo đổi vị trí hàng.

Định nghĩa 2. *Ma trận hoán vị (permutation matrix)* $\boldsymbol{P}$ *là ma trận có được bằng cách sắp xếp lại các hàng của* I *tùy ý.*

Nhân P vào bên trái A sẽ tráo các hàng của A theo đúng cách tráo các hàng của I để tạo ra P. Nói cách khác, P là tích các ma trận biến đổi sơ cấp tráo hàng (row swapping elementary operation).

Ta chọn $\boldsymbol{P}$ sao cho $\boldsymbol{PA}$ có thể giải bằng khử Gauss mà không cần tráo vị trí hàng. $\boldsymbol{P}$ được xây dựng đơn giản bằng cách áp dụng cách tráo hàng của Gauss cho ma trận $\boldsymbol{I}$.

Ví dụ 1.3. *Tìm ma trận hoán vị cho ma trận trong ví dụ 1.1, sao cho sau khi ma trận đó với ma trận gốc, nhận được một ma trận có thể dùng phương pháp khử Gauss.*

Trong ví dụ trên, hệ có thể dùng phương pháp khử Gauss sau khi đổi chỗ E_1 *và* E_3*. Vậy một ma trận hoán vị phù hợp là:*

$$\boldsymbol{P} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Sau khi tìm được $\boldsymbol{P}$, $\boldsymbol{PA}$ có thể áp dụng phương pháp khử Gauss mà không cần tráo vị trí hàng. Theo định lí 1.1, $\boldsymbol{PA}$ có phân tích LU:

$$\begin{aligned} &\boldsymbol{PA} = \boldsymbol{LU} \\ \iff &\boldsymbol{A} = \boldsymbol{P}^{-1}\boldsymbol{LU} = (\boldsymbol{P}^t\boldsymbol{L})\boldsymbol{U} \text{ (do } \boldsymbol{P}^{-1} = \boldsymbol{P}^t) \end{aligned}$$

Chứng minh phương pháp Doolittle

Phần này chứng minh kĩ hơn về tính đúng đắn của phương pháp Doolittle, và có thể bỏ qua.

Chứng minh định lí 1.1. Ta xem xét công thức (1.2). Xét trường hợp $j = 2$:

$$E_2 \coloneqq E_2 - \frac{a_{21}}{a_{11}} E_1 = E_2 - m_{21} E_1$$

Sử dụng phép biến đổi sơ cấp cộng một hàng với α lần một hàng khác (row addition elementary operation), ta có ma trận biến đổi sau:

$$\boldsymbol{M}_2^{(1)} = \begin{pmatrix} 1 & & & \\ -m_{21} & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix}$$

Tương tự, cùng trong bước khử đầu tiên (khử x_1) này, ta có dạng tổng quát hơn của $\boldsymbol{M}_j^{(1)}$:

$$\boldsymbol{M}_j^{(1)} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ -m_{j1} & & \ddots & \\ & & & 1 \end{pmatrix}$$

Sau khi kết thúc bước khử x_1, $\boldsymbol{A}$ được biến đổi thành

$$\boldsymbol{M}_2^{(1)} \boldsymbol{M}_3^{(1)} \dots \boldsymbol{M}_n^{(1)} \boldsymbol{A} = \boldsymbol{M}^{(1)} \boldsymbol{A} = \boldsymbol{M}^{(1)} \boldsymbol{A}^{(1)} = \boldsymbol{A}^{(2)}$$

và hơn nữa

$$\boldsymbol{A}^{(2)} \boldsymbol{x} = \boldsymbol{M}^{(1)} \boldsymbol{A} \boldsymbol{x} = \boldsymbol{M}^{(1)} \boldsymbol{b} = \boldsymbol{b}^{(2)}$$

Dễ dàng chứng minh được $\boldsymbol{M}^{(1)}$ ở trên, gọi là ma trận biến đổi Gauss thứ nhất, có dạng sau:

$$\boldsymbol{M}^{(1)} = \begin{pmatrix} 1 & & & \\ -m_{21} & 1 & & \\ \vdots & & \ddots & \\ -m_{n1} & & & 1 \end{pmatrix}$$

Tương tự, ta chứng minh được $\boldsymbol{M}^k$ (ma trận biến đổi Gauss thứ k) có dạng sau:

$$\boldsymbol{M}^{(k)} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -m_{k+1,k} & \ddots & & \\ & & \vdots & & \ddots & \\ & & -m_{n,k} & & & 1 \end{pmatrix}$$

Cuối cùng, sau khi kết thúc bước khử x_n:

$$\boldsymbol{M}^{(1)}\boldsymbol{M}^{(2)}\dots\boldsymbol{M}^{(n)}\boldsymbol{A} = \boldsymbol{M}\boldsymbol{A} = \boldsymbol{A}^{(n)}$$

trong đó $\boldsymbol{A}^{(n)}$ chính là ma trận $\boldsymbol{A}$ sau khi khử Gauss:

$$\boldsymbol{A}^{(n)} = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} \\ & a_{22}^{(2)} & \dots & a_{2n}^{(2)} \\ & & \ddots & \vdots \\ & & & a_{nn}^{(n)} \end{pmatrix}$$

Tới đây, chúng ta đã biến đổi được phương trình ban đầu (1.1) sang dạng sau:

$$\boldsymbol{A}^{(n)}\boldsymbol{x} = \boldsymbol{M}^{(n-1)}\boldsymbol{A}^{(n-1)}\boldsymbol{x} = \boldsymbol{M}^{(n-1)}\boldsymbol{b}^{(n-1)} = \boldsymbol{b}^{(n)} \tag{1.3}$$

trong đó $\boldsymbol{A}^{(n)}$ là một ma trận tam giác trên.

Giờ đây, nếu coi $\boldsymbol{A}^{(n)}$ là thành phần $\boldsymbol{U}$ cần tìm, ta cần nhân vào trước hai vế của (1.3) một thành phần $\boldsymbol{L}$ nào đó để đưa (1.3) về lại (1.1).

Thành phần $\boldsymbol{U}$ đã thấy chỉ đơn giản là $\boldsymbol{A}$ qua một chuỗi các biến đổi sơ cấp cộng một hàng với α lần một hàng khác. Do đó $\boldsymbol{L}$ chỉ cần là một ma trận có thể đảo ngược chuỗi biến đổi này.

Xét biến đổi $\boldsymbol{M}_2^{(1)}$, biến đổi

$$\boldsymbol{L}_2^{(1)} = \begin{pmatrix} 1 & & & \\ m_{21} & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix}$$

sẽ đảo ngược được $\boldsymbol{M}_2^{(1)}$. Lý do cũng rất đơn giản:

- $\boldsymbol{M}_2^{(1)}$ lấy E_2 trừ đi m_{21} lần E_1, thì
- $\boldsymbol{L}_2^{(1)}$ lấy E_2 cộng với m_{21} lần E_1

Tương tự, ta dễ dàng thấy

$$\boldsymbol{L}^{(k)} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & m_{k+1,k} & \ddots & & \\ & & \vdots & & \ddots & \\ & & m_{n,k} & & & 1 \end{pmatrix}$$

sẽ đảo ngược được $\boldsymbol{M}^{(k)}$.

Nhân $\boldsymbol{L}^{(k)}$ vào trước hai vế của (1.3) theo thứ tự k tăng dần từ 1 đến $n-1$, ta có:

$$\begin{aligned} \boldsymbol{L}^{(1)}\boldsymbol{L}^{(2)}\dots\boldsymbol{L}^{(n-1)}\boldsymbol{A}^{(n)}\boldsymbol{x} &= \boldsymbol{L}^{(1)}\boldsymbol{L}^{(2)}\dots\boldsymbol{L}^{(n-1)}\boldsymbol{b}^{(n)} \\ \iff \boldsymbol{L}^{(1)}\boldsymbol{L}^{(2)}\dots\boldsymbol{L}^{(n-1)}\boldsymbol{M}^{(n-1)}\dots\boldsymbol{M}^{(2)}\boldsymbol{M}^{(1)}\boldsymbol{A}\boldsymbol{x} &= \boldsymbol{L}^{(1)}\boldsymbol{L}^{(2)}\dots\boldsymbol{L}^{(n-1)}\boldsymbol{M}^{(n-1)}\dots\boldsymbol{M}^{(2)}\boldsymbol{M}^{(1)}\boldsymbol{b} \\ \iff \boldsymbol{A}\boldsymbol{x} &= \boldsymbol{b} \end{aligned}$$

Đến đây, ta nhận được phương trình (1.1) ban đầu. Vậy tích các $\boldsymbol{L}^{(k)}$ theo thứ tự trên là một giá trị $\boldsymbol{L}$ phù hợp. Không khó để chứng minh rằng:

$$\boldsymbol{L} = \boldsymbol{L}^{(1)}\boldsymbol{L}^{(2)}\dots\boldsymbol{L}^{(n-1)} = \begin{pmatrix} 1 & & & & & \\ m_{21} & \ddots & & & & \\ \vdots & \ddots & 1 & & & \\ \vdots & \vdots & m_{k+1,k} & \ddots & & \\ \vdots & \vdots & \vdots & \ddots & \ddots & \\ m_{n1} & \dots & m_{n,k} & \dots & m_{n,n-1} & 1 \end{pmatrix}$$

Vậy ta đã xây dựng được phân tích LU của $\boldsymbol{A}$, với $\boldsymbol{L}$ và $\boldsymbol{U}$ có dạng như yêu cầu.

đpcm.

1.3.2 Phân tích LL' & phương pháp Cholevsky

Thuật toán Cholesky phân tích một ma trận xác định dương ra dạng LL'. Để tìm hiểu phân tích này, ta cần biết về ma trận xác định dương.

Định nghĩa 3. Ma trận xác định dương (positive definitive matrix)
Ma trận $\boldsymbol{A}$ $n \times n$ gọi là xác định dương nếu:

- *$\boldsymbol{A}$ đối xứng, và*
- $\boldsymbol{x}^t\boldsymbol{A}\boldsymbol{x} > 0\, \forall \boldsymbol{x} \neq 0$

$\boldsymbol{L}$ có dạng như sau (viết rõ lại để tiện biểu diễn về sau):

$$\boldsymbol{L} = \begin{pmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & \dots & \dots & l_{nn} \end{pmatrix}$$

Ta thừa nhận định lí sau:

Định lí 1.2. *Ma trận $\boldsymbol{A}$ là khi và chỉ khi nó phân tích được ra dạng $\boldsymbol{L}\boldsymbol{L}^t$, trong đó $\boldsymbol{L}$ là ma trận tam giác dưới với đường chéo chính khác 0.*

Do có số ẩn không lớn (với ma trận $n \times n$ cần tìm tổng cộng $\frac{n(n+1)}{2}$ ẩn), đồng thời ma trận tích có dạng phù hợp, nên phương pháp này có thể giải bằng tay các ẩn theo cách thế thông thường với n nhỏ. Nếu không tiện tính tay, ta có phương pháp chính xác sau:

Phương pháp. Phương pháp Cholevsky

Phương pháp Cholevsky phân tích ma trận xác định dương $\boldsymbol{A}$ thành dạng LL'.

Lần lượt tính phần tử khác 0 ở cột 1, 2, ... Bước thứ i sẽ tính các phần tử thuộc cột i như sau:

- *Tính l_{ii}:*

$$l_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2\right)^{0,5}$$

- *Tính các phần tử còn lại, nếu có:*

$$l_{ji} = \frac{1}{l_{ii}}\left(a_{ji} - \sum_{k=i+1}^{n} l_{jk} l_{ik}\right) \mid j \in [i+1, n]$$

1.3.3 Phân tích LU cho ma trận dải & thuật toán Crout

Ta quay trở lại với phân tích LU, tuy nhiên sử dụng thuật toán khác, nhanh vượt trội so với Doolittle, cho một loại ma trận đặc biệt, ma trận dải.

Định nghĩa 4. Ma trận dải (band matrix)

Ma trận $n \times n$ gọi là ma trận dải nếu có $1 < p,\ q < n$ sao cho $a_{ij} = 0$ khi $j - i \geq p$ hoặc $i - j \geq q$.

Nói cách khác, hai chỉ số p, q chỉ số đường chéo mà các phần tử trên đó không nhất thiết bằng 0:

- p đường chéo trên đường chéo chính, gồm cả đường chéo chính
- q đường chéo dưới đường chéo chính, gồm cả đường chéo chính

Ta lại xét tiếp một trường hợp đặc biệt: ma trận dải với $p = q = 2$, gọi là ma trận ba đường chéo (tridiagonal matrix).

Nếu $\boldsymbol{A}$ trong (1.1) có dạng ba đường chéo:

$$\boldsymbol{A} = \begin{pmatrix} a_{11} & a_{12} & & & \\ a_{21} & a_{22} & a_{23} & & \\ & a_{32} & a_{33} & \ddots & \\ & & \ddots & \ddots & a_{n-1,n} \\ & & & a_{n,n-1} & a_{nn} \end{pmatrix}$$

thì có thể phân tích $\boldsymbol{A}$ ra dạng LU như sau:

$$\boldsymbol{L} = \begin{pmatrix} l_{11} & & & & \\ l_{21} & l_{22} & & & \\ & l_{32} & l_{33} & & \\ & & \ddots & \ddots & \\ & & & l_{n,n-1} & l_{nn} \end{pmatrix}, \text{ và } \boldsymbol{U} = \begin{pmatrix} 1 & u_{12} & & & \\ & 1 & u_{23} & & \\ & & \ddots & \ddots & \\ & & & \ddots & u_{n-1,n} \\ & & & & 1 \end{pmatrix}$$

Do có số ẩn không lớn (với ma trận $n \times n$ cần tìm tổng cộng $3n - 2$ ẩn), đồng thời ma trận tích có dạng phù hợp, nên phương pháp này có thể giải bằng tay các ẩn theo cách thế thông thường với n nhỏ. Nếu không tiện tính tay, ta giới thiệu qua phương pháp sau:

Phương pháp. Phương pháp Crout cho trường hợp ma trận ba đường chéo

Phương pháp Crout dùng để phân tích ma trận ra dạng LU. Dạng LU của phương pháp Crout có khác biệt so phương pháp Doolittle:

- *Phương pháp Crout tạo ra ma trận $\boldsymbol{U}$ có đường chéo chính bằng* 1.
- *Phương pháp Doolittle tạo ra ma trận $\boldsymbol{L}$ có đường chéo chính bằng* 1.