Wstęp do metod numerycznych

Eliminacja Gaussa i faktoryzacja *LU*

P. F. Góra

http://th-www.if.uj.edu.pl/zfs/gora/

2019

Co można zrobić z układem równań

...tak, aby jego rozwiązania się nie zmieniły?

Rozważam układ równań (przykład 3×3 dla oszczędności miejsca):

$$\left\{ \begin{array}{ccccccc} a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & = & b_2 \\ a_{31}x_1 & + & a_{32}x_2 & + & a_{33}x_3 & = & b_3 \end{array} \right. \tag{1}$$

1. Równania można zapisać w innej kolejności:

$$\left\{ \begin{array}{ccccccc} a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & = & b_2 \\ a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & = & b_1 \\ a_{31}x_1 & + & a_{32}x_2 & + & a_{33}x_3 & = & b_3 \end{array} \right. \tag{2}$$

Odpowiada to **permutacji wierszy macierzy układu równań, z jednoczesną permutacją kolumny wyrazów wolnych**.

2. Równania można dodać stronami, po pomnożeniu przez dowolną stałą różną od zera:

$$
\left\{
\begin{array}{rcrcrcr}
a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & = & b_1 \\
a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & = & b_2 \\
(z \cdot a_{11} + a_{31})x_1 & + & (z \cdot a_{12} + a_{32})x_2 & + & (z \cdot a_{13} + a_{33})x_3 & = & z \cdot b_1 + b_3
\end{array}
\right.
\tag{3}
$$

Odpowiada to **zastąpieniu jednego wiersza macierzy układu równań przez dowolną kombinację liniową tego wiersza z innymi, z jednoczesną analogiczną operacją na kolumnie wyrazów wolnych**.

3. We wszystkich równaniach można przestawić kolejność, w jakiej pojawiają się zmienne:

$$\left\{\begin{array}{ccccccc} a_{11}x_1 & + & a_{13}x_3 & + & a_{12}x_2 & = & b_1 \\ a_{21}x_1 & + & a_{23}x_3 & + & a_{22}x_2 & = & b_2 \\ a_{31}x_1 & + & a_{33}x_3 & + & a_{32}x_2 & = & b_3 \end{array}\right. \tag{4}$$

Odpowiada to **permutacji *kolumn* macierzy układu równań, z jednoczesną permutacją kolumny niewiadomych**.

Eliminacja Gaussa

Rozpatrzmy jeszcze raz układ równań

$$\left\{\begin{array}{rcrcrcl} a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & = & b_2 \\ a_{31}x_1 & + & a_{32}x_2 & + & a_{33}x_3 & = & b_3 \end{array}\right. \tag{5}$$

Podzielmy pierwsze równanie stronami przez a_{11}

$$\left\{\begin{array}{rcrcrcl} x_1 & + & \frac{a_{12}}{a_{11}}x_2 & + & \frac{a_{13}}{a_{11}}x_3 & = & \frac{b_1}{a_{11}} \\ a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & = & b_2 \\ a_{31}x_1 & + & a_{32}x_2 & + & a_{33}x_3 & = & b_3 \end{array}\right. \tag{6}$$

Teraz mnożymy pierwsze z równań (6) przez a_{21} i odejmijmy stronami od drugiego, a następnie mnożymy pierwsze z równań (6) przez a_{31} i odejmijmy stronami od trzeciego. Otrzymujemy

$$\left\{\begin{array}{lclcl} x_1 + \frac{a_{12}}{a_{11}}x_2 & + & \frac{a_{13}}{a_{11}}x_3 & = & \frac{b_1}{a_{11}} \\ \left(a_{22} - \frac{a_{21}a_{12}}{a_{11}}\right)x_2 & + & \left(a_{23} - \frac{a_{21}a_{13}}{a_{11}}\right)x_3 & = & b_2 - \frac{a_{21}}{a_{11}}b_1 \\ \left(a_{32} - \frac{a_{31}a_{12}}{a_{11}}\right)x_2 & + & \left(a_{33} - \frac{a_{31}a_{13}}{a_{11}}\right)x_3 & = & b_3 - \frac{a_{31}}{a_{11}}b_1 \end{array}\right. \tag{7a}$$

Przepiszmy to w postaci (tylko zmiana oznaczeń!)

$$\left\{\begin{array}{lclclcl} x_1 & + & a'_{12}x_2 & + & a'_{13}x_3 & = & b'_1 \\ & & a'_{22}x_2 & + & a'_{23}x_3 & = & b'_2 \\ & & a'_{32}x_2 & + & a'_{33}x_3 & = & b'_3 \end{array}\right. \tag{7b}$$

W układzie równań (7b) pierwsza zmienna, x_1, występuje wyłącznie w pierwszym równaniu. Tego równania już nie przekształcamy, natomiast z pozostałymi równaniami postępujemy analogicznie: dzielimy drugie stronami przez a'_{22} i odpowiednio mnożąc, odejmujemy od trzeciego. Otrzymujemy

$$\left\{\begin{array}{ccccccc} x_1 & + & a'_{12}x_2 & + & a'_{13}x_3 & = & b'_1 \\ & & x_2 & + & a''_{23}x_3 & = & b''_2 \\ & & & & a''_{33}x_3 & = & b''_3 \end{array}\right. \tag{8}$$

Teraz pierwsza zmienna występuje wyłącznie w pierwszym równaniu, druga — w pierwszym i w drugim. Gdyby równań było więcej, moglibyśmy to postępowanie kontynuować.

Ostatecznie otrzymalibyśmy równanie postaci

$$\left\{\begin{array}{ccccccccccccc} x_1 & + & \bullet x_2 & + & \bullet x_3 & + & \dots & + & \bullet x_N & = & \tilde{b}_1 \\ & & x_2 & + & \bullet x_3 & + & \dots & + & \bullet x_N & = & \tilde{b}_2 \\ & & & & x_3 & + & \dots & + & \bullet x_N & = & \tilde{b}_3 \\ & & & & & & \dots & & \dots & & \dots \\ & & & & & & & & x_N & = & \tilde{b}_N \end{array}\right. \tag{9}$$

gdzie symbole $\bullet$ oznaczają *jakieś* współczynniki, dające się wyliczyć z pierwotnych współczynników równania, $\tilde{b}_i$ są przekształconymi w toku całej procedury wyrazami wolnymi.

Równanie w postaci (9) nazywamy układem równań *z macierzą trójkątną górną*. Algorytm prowadzący od (5) do (9) nazywamy *eliminacją Gaussa*.

Dygresja: Złożoność obliczeniowa

Niech N oznacza liczbę danych wejściowych pewnego algorytmu. Niech $\mathcal{M}(N)$ oznacza liczbę operacji, jaką algorytm ten wykonuje dla N danych. Mówimy, że algorytm ma złożoność obliczeniową $O\left(\mathcal{P}(N)\right)$ jeżeli

$$\exists N_0 \in \mathbb{N},\ A_1, A_2 > 0\ \forall N > N_0\colon\ A_1 \cdot \mathcal{P}(N) \leqslant \mathcal{M}(N) \leqslant A_2 \cdot \mathcal{P}(N) \tag{10}$$

Złożoność obliczeniowa eliminacji Gaussa

Aby usunąć zmienną x_1 z jednego wiersza, należy wykonać $O(N)$ operacji. Ponieważ zmienną x_1 musimy usunąć z $N-1$ wierszy, musimy łącznie wykonać $O(N^2)$ operacji. Ponieważ musimy to samo zrobić ze zmiennymi $x_2, x_3, \ldots$, ostatecznie musimy wykonać $O(N^3)$ operacji.

Złożoność obliczeniowa eliminacji Gaussa wynosi $O(N^3)$.

Backsubstitution

Rozpatrzmy układ równań w postaci (9). Ostatnie równanie jest rozwiązane ze względu na x_N. Podstawiamy to rozwiązanie do wszystkich poprzednich równań. Teraz drugie od dołu równanie ma tylko jedną nieznaną zmienną — x_{N-1}, a coś takiego umiemy rozwiązać. Podstawiamy to rozwiązanie do równania trzeciego od dołu i do poprzednich. Teraz trzecie od dołu równanie zawiera tylko jedną zmienną, x_{N-2}. Rozwiązujemy, podstawiamy do poprzednich i tak dalej...

Ponieważ wyeliminowanie jednej zmiennej wymaga $O(N)$ operacji, a musimy wyeliminować N zmiennych, cały koszt rozwiązania układu z macierzą trójkątną górną za pomocą algorytmu *backsubstitution* wynosi $O(N^2)$. Jest to *niewiele* w porównaniu z kosztem eliminacji Gaussa.

Całkowity koszt rozwiązania układu N równań liniowych za pomocą eliminacji Gaussa z następującym *backsubstitution* wynosi $O(N^3)$.

Czy coś może pójść źle?

Cały algorytm zawali się, **jeżeli w którymś momencie trzeba będzie wykonać dzielenie przez zero**

$a_{11} = 0$ lub $a'_{22} = 0$, lub $a''_{33} = 0$ itd.

Przykład

Układu równań

$$\left\{\begin{array}{rcrcrcl} & & y & + & z & = & 1 \\ x & + & y & + & z & = & 2 \\ 2x & & & - & z & = & 0 \end{array}\right. \tag{11}$$

nie da się doprowadzić do postaci trójkątnej górnej za pomocą eliminacji Gaussa. Jeśli jednak przestawimy pierwszy wiersz z drugim lub z trzecim, eliminacja Gaussa powiedzie się.

Ze względów numerycznych staramy się także unikać dzielenia przez liczby bardzo małe co do wartości bezwzględnej. *Formalnie*, w arytmetyce dokładnej, jest to wykonalne, ale *w praktyce* może to doprowadzić do bardzo znacznej utraty dokładnosci, tak, że ostateczny wynik będzie numerycznie bezwartościowy.

Wybór elementu podstawowego

Przypuśćmy, że na pewnym etapie eliminacji Gaussa mamy układ równań

$$\left\{ \begin{array}{ccccccccccccc} x_1 & + & \dots & & \dots & & \dots & & \dots & & \dots & = & b_1 \\ & & x_2 & + & \dots & & \dots & & \dots & & \dots & = & b_2 \\ & & & & \dots & & \dots & & \dots & & \dots & & \dots \\ & & & & & & a_{kk}x_k & + & a_{k,k+1}x_{k+1} & + & \dots & = & b_k \\ & & & & & & a_{k+1,k}x_k & + & a_{k+1,k+1}x_{k+1} & + & \dots & = & b_{k+1} \\ & & & & & & \dots & & \dots & & \dots & & \dots \\ & & & & & & a_{Nk}x_k & + & a_{N,k+1}x_{k+1} & + & \dots & = & b_N \end{array} \right. \tag{12}$$

“Powinniśmy” teraz dzielić przez a_{kk}. Zamiast tego wśród współczynników $a_{kk}, a_{k+1,k}, a_{k+2,k}, \dots, a_{Nk}$ wyszukujemy największy co do modułu, permutujemy wiersze tak, aby ten największy co do modułu znalazł się w pozycji diagonalnej i dzielimy przez niego. Współczynnik wypromowany do pozycji diagonalnej nazywa się elementem podstawowym (ang. pivot). Ten krok algorytmu nazywa się *częściowym wyborem elementu podstawowego*. Dalej postępujemy jak poprzednio.

Koszt wyszukania jednego elementu podstawowego wynosi $O(N)$. Jeżeli robimy to w każdym kroku, całkowity koszt jest rzędu $O(N^2)$, a więc jest mały w porównaniu ze złożonością obliczeniową samej eliminacji Gaussa. Wynika z tego, iż częściowego wyboru elementu podstawowego należy zawsze dokonywać, gdyż nie zwiększa to znacznie kosztu całej procedury, może natomiast zapewnić numeryczną stabilność algorytmu.

Zamiast szukać elementu podstawowego wyłącznie w jednej kolumnie, można szukać największego co do modułu współczynnika wśród wszystkich $a_{i,j}, k \leqslant i, j \leqslant N$. Po znalezieniu, należy tak spermutować wiersze i kolumny układu równań, aby element podstawowy znalazł się w pozycji diagonalnej. Nazywa się to *pełnym wyborem elementu podstawowego*. Zauważmy, że koszt numeryczny wynosi $O(N^3)$, a więc staje się porównywalny z kosztem całej eliminacji Gaussa, ponadto zaś permutacja kolumn wymaga późniejszego odwikłania permutacji elementów rozwiązania, co jest kłopotliwe. Pełny wybór elementu podstawowego zapewnia większą stabilność numeryczną, niż wybór częściowy, ale w praktyce jest rzadko używany, ze wskazanych wyżej powodów.

Do skutecznego przeprowadzenia eliminacji Gaussa potrzebna jest znajomość kolumny wyrazów wolnych, gdyż wyrazy wolne także są przekształcane i permutowane w czasie eliminacji.

Uwagi o eliminacji Gaussa

Przypuśćmy, że mamy rozwiązać kilka układów równań z tą samą lewą stroną, a różnymi wyrazami wolnymi:

$$\mathbf{A}\mathbf{x}^{(i)} = \mathbf{b}^{(i)}\,, \quad i = 1, 2, \ldots, M \tag{13}$$

gdzie $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{x}^{(i)}, \mathbf{b}^{(i)} \in \mathbb{R}^N$. Eliminacja Gaussa (z wyborem elementu podstawowego!) jest efektywna, jeżeli z góry znamy *wszystkie* prawe strony $\mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \ldots, \mathbf{b}^{(M)}$, gdyż w tym wypadku przeprowadzając eliminację Gaussa, możemy przekształcać wszystkie prawe strony *jednocześnie*. Całkowity koszt rozwiązania (13) wynosi wówczas $O(N^3) + O(MN^2)$.

Jeżeli jednak wszystkie prawe strony nie są z góry znane — co jest sytuacją typową w obliczeniach iteracyjnych — eliminacja Gaussa jest nieefektywna, gdyż trzebaby ją niepotrzebnie przeprowadzać dla każdej prawej strony z osobna, co podnosiłoby koszt numeryczny do $O(MN^3) + O(MN^2)$.

Równania macierzowe

Zauważmy, że korzystając z własności mnożenia macierzy, równania (13) można zapisać w postaci

$$\mathbf{A}\mathbf{X} = \mathbf{B} \tag{14}$$

gdzie $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{X}, \mathbf{B} \in \mathbb{R}^{N\times M}$, przy czym

$$\mathbf{X} = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(M)} \\ x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(M)} \\ \dots & \dots & \dots & \dots \\ x_N^{(1)} & x_N^{(2)} & \dots & x_N^{(M)} \end{bmatrix}$$

i analogicznie dla $\mathbf{B}$. Innymi słowy, macierzowy układ równań (14) jest równoważny układowi równań liniowych (13) z M niezależnymi prawymi stronami. Oczywiście do rozwiązywania układów równań macierzowych postaci (14) można używać nie tylko eliminacji Gaussa, ale wszystkich innych metod właściwych dla układów równań liniowych.

Uwaga — jawna konstrukcja macierzy odwrotnej

Z powyższych uwag widać, że problem jawnej konstrukcji macierzy odwrotnej

$$\mathbf{A} \cdot \mathbf{A}^{-1} = \mathbb{I} \tag{15}$$

jest problemem postaci (14), a więc kolejne kolumny macierzy odwrotnej uzyskujemy rozwiązując kolejne układy (13) dla $i = 1, 2, \ldots, N$, przy czym $\mathbf{b}^{(1)} = [1, 0, 0, \ldots]^T$, $\mathbf{b}^{(2)} = [0, 1, 0, \ldots]^T$ itd. Rozwiązanie układu równań $\mathbf{A}\mathbf{x} = b$ poprzez jawną konstrukcję macierzy odwrotnej, $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, wymaga rozwiązania N układów równań liniowych, co oznacza koszt $O(2N^3)$, podczas gdy koszt bezpośredniego rozwiązania równania $\mathbf{A}\mathbf{x} = \mathbf{b}$ to $O(N^3)$.

Pojawiający się często we wzorach napis $\mathbf{A}^{-1}\mathbf{b}$ **zawsze** rozumiemy jako wezwanie do znalezienie wektora $\mathbf{z}$ takiego, że $\mathbf{Az} = \mathbf{b}$, **nigdy** nie jako polecenie jawnego skonstruowania macierzy $\mathbf{A}^{-1}$.

Przykład

Wyrażenie

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{J}^{-1}\mathbf{f}_n \tag{16}$$

interpretujemy jako

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{z} \tag{17a}$$

gdzie

$$\mathbf{Jz} = \mathbf{f}_n \tag{17b}$$

Kiedy eliminacja Gaussa nie wystarcza

Eliminacji Gaussa warto używać do rozwiązywania pojedynczego równania

$$\mathbf{A}\mathbf{x} = \mathbf{b}\,, \quad \det \mathbf{A} \neq 0 \tag{18}$$

lub do kilku takich równań, o ile ich wszystkie prawe strony są z góry znane. (Przykładem takiej sytuacji jest rozwiązywanie liniowych równań macierzowych, w tym **niezwykle egzotyczna** sytuacja, w której należy znaleźć jawną odwrotność danej macierzy.) Widzimy wszkaże, że kolejne kroki eliminacji Gaussa zależą wyłącznie od elementów macierzy $\mathbf{A}$ — prawe strony, choć trzeba je przekształcać, są niejako “biernymi” uczestnikami procesu. Należy pomyśleć o algorytmach, które zajmują się samą macierzą, odkładając przekształcanie prawych stron (kolmun wyrazów wolnych) do czasu, gdy będzie to *naprawdę* potrzebne.

Takimi algorytmami są *faktoryzacje*, czyli przedstawienie macierzy $\mathbf{A}$ jako iloczynu dwu macierzy w jakimś sensie prostszych:

$$\mathbf{A} = \mathbf{Y} \cdot \mathbf{Z} \,. \tag{19}$$

Faktoryzacja *LU*

Przypuśćmy, że udało nam się znaleźć faktoryzację

$$\mathbf{A} = \mathbf{L} \cdot \mathbf{U} \,, \tag{20}$$

gdzie macierz $\mathbf{U}$ jest trójkątna górna (wszystkie elementy poniżej głównej przekątnej są zerami), natomiast $\mathbf{L}$ jest trójkatna dolna; dodatkowo przyjmujemy, że jej wszystkie elementy diagonalne są równe 1, $l_{ii} = 1$. Taką faktoryzację nazywamy *faktoryzacją LU*.

Jeżeli faktoryzacja *LU* jest znana, równanie

$$\mathbf{A}\mathbf{x} \equiv \mathbf{L}\,\underbrace{\mathbf{U}\mathbf{x}}_{\mathbf{y}} = \mathbf{b} \tag{21}$$

rozwiązujemy jako

$$\begin{aligned} \mathbf{L}\mathbf{y} &= \mathbf{b} && \text{(22a)} \\ \mathbf{U}\mathbf{x} &= \mathbf{y} && \text{(22b)} \end{aligned}$$

Pierwsze z tych równań rozwiązujemy metodą *forward substitution*, drugie — metodą *back substitution*. Ponieważ są to równania z macierzami trójkątnymi, koszt obliczeniowy rozwiązania każdego z nich wynosi $O(N^2)$, a zatem koszt rozwiązania (21) wynosi $O(2N^2)$.

Pozostaje jezcze “tylko” dokonać samej faktoryzacji.

Algorytm Doolittle'a

Aby dokonać faktoryzacji *LU*, należy obliczyć N^2 nieznanych elementów macierzy $\mathbf{L}, \mathbf{U}$. Rozpiszmy (20):

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1N} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2N} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3N} \\ \dots & \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & a_{N3} & \dots & a_{NN} \end{bmatrix}}_{\mathbf{A}} = \underbrace{\begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \dots & \dots & \dots & \dots & \\ l_{N1} & l_{N2} & l_{N3} & \dots & 1 \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1N} \\ & u_{22} & u_{23} & \dots & u_{2N} \\ & & u_{33} & \dots & u_{3N} \\ & & & \dots & \dots \\ & & & \dots & u_{NN} \end{bmatrix}}_{\mathbf{U}} \tag{23}$$

Okazuje się, że rozwiązywanie równań na poszczególne elementy l_{ij}, u_{pq} jest proste, jeżeli przeprowadza się je *we właściwej kolejności*, odpowiadającej kolejnym kolumnom macierzy $\mathbf{A}$.

Pierwsza kolumna: Aby znaleźć pierwszą kolumnę macierzy $\mathbf{A}$, mnożymy kolejne wiersze $\mathbf{L}$ przez pierwszą kolumnę macierzy $\mathbf{U}$. Ale ta kolumna ma tylko jeden element. Otrzymujemy

$$
\begin{array}{rcl}
u_{11} & = & a_{11} \\
l_{21}u_{11} & = & a_{21} \\
l_{31}u_{11} & = & a_{31} \\
\dots & \dots & \dots \\
l_{N1}u_{11} & = & a_{N1}
\end{array}
\tag{24}
$$

Z pierwszego z równań (24) obliczamy u_{11}, a następnie z kolejnych $l_{21}, l_{31}, \dots, l_{N1}$.

Druga kolumna: Wyrażenia na elementy drugiej kolumny macierzy $\mathbf{A}$ powstają z przemnożenia kolejnych wierszy $\mathbf{L}$ przez drugą kolumnę $\mathbf{U}$:

$$\begin{array}{ccrcc} & & u_{12} & = & a_{12} \\ l_{21}u_{12} & + & u_{22} & = & a_{22} \\ l_{31}u_{12} & + & l_{32}u_{22} & = & a_{32} \\ \dots & \dots & \dots & \dots & \dots \\ l_{N1}u_{12} & + & l_{N2}u_{22} & = & a_{N2} \end{array} \tag{25}$$

Z pierwszego z tych równań obliczamy u_{12}. W tym momencie u_{12} jest już znane, podobnie jak obliczone wcześniej $l_{\bullet 1}$, a zatem z drugiego z równań (25) obliczamy u_{22}, a z kolejnych $l_{32}, l_{42}, \dots, l_{N2}$.

I tak dalej.

Widać, że średni koszt obliczenia któregoś z nieznanych elementów l_{ij}, u_{pq} jest rzędu $O(N)$. Ponieważ elementów tych jest N^2, złożoność numeryczna algorytmu Doolittle'a wynosi $O(N^3)$. Całkowity koszt rozwiązania układu równań liniowych, a więc faktoryzacji *LU* i rozwiązania układów równań z macierzami trójkątnymi (22), jest taki sam, jak eliminacji Gaussa.

Przewaga faktoryzacji *LU* nad eliminacją Gaussa polega na tym, iż przy pomocy faktoryacji *LU* można rozwiązywać dowolnie wiele równań z takimi samymi lewymi stronami (macierzami), przy czym "kosztowną" część, a więc samą faktoryzację, oblicza się tylko raz.

Z uwagi na symetrię problemu i na kolejność wykonywanych obliczeń, faktoryzacja *LU* nie wymaga dodatkowej pamięci do zapamiętania obliczonych elementów faktoryzacji: elementy macierzy $\mathbf{L}$ (bez diagonali) zapamiętujemy w poddiagonalnym trójkącie macierzy $\mathbf{A}$, elementy macierzy $\mathbf{U}$ — na diagonali i w ponaddiagonalnym trójkącie $\mathbf{A}$.

Przykład

W celu dokonania faktoryzacji LU macierzy

$$\begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix} \tag{26}$$

musimy rozwiązać równania

$$\begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & \\ l_{21} & 1 & \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ & u_{22} & u_{23} \\ & & u_{33} \end{bmatrix} \tag{27}$$

ze względu na l_{ik}, u_{kj}. W tym celu zapiszmy indywidualne równania, na jakie rozpada się (27), w kolejności odpowiadające przeglądaniu macierzy (26) kolumnami.

Pierwsza kolumna macierzy (26) odpowiada

$$\begin{aligned} u_{11} &= 1 && (28a) \\ l_{21}u_{11} &= 2 && (28b) \\ l_{31}u_{11} &= 2 && (28c) \end{aligned}$$

skąd natychmiast otrzymujemy

$$u_{11} = 1\,, \quad l_{21} = 2\,, \quad l_{31} = 2\,. \tag{29}$$

Zwróćmy uwagę, iż pierwsze z równań (28) służy do wyliczenia elementu macierzy U, drugie i trzecie — do wyliczenia elementów macierzy L.

Druga kolumna odpowiada

$$\begin{aligned} u_{12} &= 2 && (30a) \\ l_{21}u_{12} + u_{22} &= 1 && (30b) \\ l_{31}u_{12} + l_{32}u_{22} &= 2 && (30c) \end{aligned}$$

Zauważmy, że jeśli równania (30) rozwiązywać w kolejności „naturalnej”, od góry do dołu, każde z nich okazuje się być równaniem z *jedną* niewiadomą. Pierwsze dwa służą do wyliczenia elementów macierzy U, trzecie do wyliczenia elementu macierzy L. Otrzymujemy

$$u_{12} = 2\,, \quad u_{22} = -3\,, \quad l_{32} = \frac{2}{3}\,. \tag{31}$$

Trzecia kolumna (26) daje

$$\begin{aligned} u_{13} &= 2 && \text{(32a)}\\ l_{21}u_{13} + u_{23} &= 2 && \text{(32b)}\\ l_{31}u_{13} + l_{32}u_{23} + u_{33} &= 1 && \text{(32c)} \end{aligned}$$

W tym wypadku wszystkie trzy równania (32) służą do obliczenia elementów macierzy U. Podobnie jak poprzednio, jeśli równania te rozwiązywać od góry do dołu, każde z nich jest równaniem z jedną niewiadomą. Jako rozwiązanie otrzymujemy

$$u_{13} = 2\,, \quad u_{23} = -2\,, \quad u_{33} = -\frac{5}{3}\,. \tag{33}$$

Ostatecznie

$$\begin{bmatrix} 1 & & \\ 2 & 1 & \\ 2 & \frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 2 \\ & -3 & -2 \\ & & -\frac{5}{3} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix}. \tag{34}$$

Równość w (34) można sprawdzić bezpośrednim rachunkiem.

Algorytm Crouta

Przedstawiony algorytm nie zawiera wyboru elementu podstawowego (pivotingu), ten zaś jest niezbędny dla stabilności całego procesu. Z uwagi na symetrie faktoryzacji, tylko częściowy wybór elementu podstawowego jest możliwy. Omówimy to na przykładzie. Rozwiązaujàc równania (25) począwszy od drugiego z nich, obliczamy

$$\begin{aligned} l_{22}u_{22} &= a_{22} - l_{21}u_{12} \qquad (l_{22} \equiv 1) \\ l_{32}u_{22} &= a_{32} - l_{31}u_{12} \\ &\cdots \\ l_{N2}u_{22} &= a_{N2} - l_{N1}u_{12} \end{aligned} \tag{35}$$

Porównujemy teraz wyliczone lewe strony równań (35) i wybieramy największą (na moduł) z nich; tę uznajemy za "właściwe" u_{22} — odpowiada to permutacji wierszy macierzy $\mathbf{A}$. Należy także spermutować już obliczone wiersze macierzy $\mathbf{L}$. W rezultacie otrzymujemy faktoryzację *LU* nie samej macierzy $\mathbf{A}$, ale macierzy różniącej się od niej pewną permutacją wierszy.

Przykład

Rozpatrzmy problem znalezienia następującej faktoryzacji:

$$\begin{bmatrix} 2 & 4 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 0 & 1 & 2 & -1 \\ -1 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}. \tag{36}$$

Faktoryzację znajdujemy przechodząc macierz $\mathbf{A}$ *kolumnami*, poczynając od lewego górnego rogu. Pierwsza kolumna daje zatem

$$
\begin{aligned}
a_{11}: &\quad u_{11} = 2 && \text{(37a)}\\
a_{21}: &\quad l_{21}u_{11} = 1 && \text{(37b)}\\
a_{31}: &\quad l_{31}u_{11} = 0 && \text{(37c)}\\
a_{41}: &\quad l_{41}u_{11} = -1 && \text{(37d)}
\end{aligned}
$$

Po przejrzeniu pierwszej kolumny faktoryzacja ma postać

$$
\begin{bmatrix} 2 & 4 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 0 & 1 & 2 & -1 \\ -1 & 1 & 0 & 1 \end{bmatrix}
=
\begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ 0 & l_{32} & 1 & 0 \\ -\frac{1}{2} & l_{42} & l_{43} & 1 \end{bmatrix}
\begin{bmatrix} 2 & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}. \tag{38}
$$

Przystępujemy do przeglądania drugiej kolumny:

$$a_{12}: \qquad u_{12} = 4 \tag{39a}$$

$$a_{22}: \qquad \frac{1}{2}\cdot 4 + u_{22} = 2 \implies u_{22} = 0 \tag{39b}$$

$$a_{32}: \qquad 0\cdot 4 + l_{32}u_{22} = 1 \implies l_{32}u_{22} = 1 \tag{39c}$$

$$a_{42}: \qquad -\frac{1}{2}\cdot 4 + l_{42}u_{22} = 1 \implies l_{42}u_{22} = 3 \tag{39d}$$

Widać, iż równań (39) nie da się rozwiązać ze względu na l_{32}, l_{42}. Dzieje się tak dlatego, że aktualny element diagonalny („element podstawowy”) jest zerem. Aby uniknąć tej sytuacji, należy przestawić drugi wiersz faktoryzowanej macierzy z pewnym innym wierszem leżącym *poniżej* drugiego; oczywiście należy także przestawić już obliczone elementy macierzy $\mathbf{L}$ odpowiadające przestawianym wierszom $\mathbf{A}$. Jako wiersz, który zajmie miejsce wiersza drugiego, wybieramy ten, który prowadzi do największej (na moduł) wartości po prawej stronie równań (39), jako że ta wartość stanie się nowym elementem diagonalnym, przez który będziemy dzielić. W naszym przykładzie jest to wiersz czwarty. Zatem

$$\begin{bmatrix} 2 & 4 & 1 & 1 \\ -1 & 1 & 0 & 1 \\ 0 & 1 & 2 & -1 \\ 1 & 2 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ 0 & l_{32} & 1 & 0 \\ \frac{1}{2} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix} . \tag{40}$$

Kolory wskazują co z czym było przestawiane. Podkreślam, iż w macierzy $\mathbf{L}$ przestawieniu podlegają tylko *już obliczone* elementy, a więc elementy leżące na lewo od aktualnie analizowanej kolumny. Ponieważ wiersze leżące powyżej aktualnie obliczanego elementu diagonalnego nie ulegają zmianie, obliczoną wartość u_{12} można już było wpisać do macierzy. Teraz z łatwością obliczamy

$$a_{22}: \quad -\frac{1}{2}\cdot 4 + u_{22} = 1 \implies u_{22} = 3 \tag{41a}$$

$$a_{32}: \quad 0\cdot 4 + l_{32}u_{22} = 1 \implies l_{32} = \frac{1}{3} \tag{41b}$$

$$a_{42}: \quad \frac{1}{2}\cdot 4 + l_{42}u_{22} = 2 \implies l_{42} = 0 \tag{41c}$$

a zatem

$$\begin{bmatrix} 2 & 4 & 1 & 1 \\ -1 & 1 & 0 & 1 \\ 0 & 1 & 2 & -1 \\ 1 & 2 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ 0 & \frac{1}{3} & 1 & 0 \\ \frac{1}{2} & 0 & l_{43} & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & u_{13} & u_{14} \\ 0 & 3 & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}. \tag{42}$$

Przystępujemy do przeglądania trzeciej kolumny.

$$a_{13}: \quad u_{13} = 1 \tag{43a}$$

$$a_{23}: \quad -\frac{1}{2} \cdot u_{13} + u_{23} = 0 \implies u_{23} = \frac{1}{2} \tag{43b}$$

$$a_{33}: \quad 0 \cdot u_{13} + \frac{1}{3} \cdot u_{23} + u_{33} = 2 \implies u_{33} = \frac{11}{6} \tag{43c}$$

$$a_{43}: \quad \frac{1}{2} \cdot u_{13} + 0 \cdot u_{23} + l_{43}u_{33} = 3 \implies l_{43}u_{33} = \frac{5}{2} \tag{43d}$$

W tym wypadku nie *musimy* permutować wierszy (równania (43) nie zawierają dzielenia przez zero), tym niemniej *powinniśmy* to zrobić, aby elementem diagonalnym był element o możliwie największym module. Ponieważ $5/2 > 11/6$, permutujemy trzeci i czwarty wiersz macierzy $\mathbf{A}$, przestawiając jednocześnie odpowiednie *już obliczone* elementy macierzy $\mathbf{L}$.

A zatem

$$\begin{bmatrix} 2 & 4 & 1 & 1 \\ -1 & 1 & 0 & 1 \\ \color{red}{1} & \color{red}{2} & \color{red}{3} & \color{red}{1} \\ \color{red}{0} & \color{red}{1} & \color{red}{2} & \color{red}{-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ \color{blue}{\frac{1}{2}} & \color{blue}{0} & 1 & 0 \\ \color{blue}{0} & \color{blue}{\frac{1}{3}} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & 1 & u_{14} \\ 0 & 3 & \frac{1}{2} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix} . \tag{44}$$

Jak poprzednio, kolory pokazują elementy, które zostały przestawione. Teraz z łatwością obliczamy najpierw brakujące elementy u_{33}, l_{43}, później zaś elementy ostatniej kolumny macierzy $\mathbf{U}$ — w tym przypadku nie trzeba (a nawet nie da się) wykonywać już żadnych „pivotów”. Ostatecznie otrzymujemy

$$\begin{bmatrix} 2 & 4 & 1 & 1 \\ -1 & 1 & 0 & 1 \\ 1 & 2 & 3 & 1 \\ 0 & 1 & 2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 1 & 0 \\ 0 & \frac{1}{3} & \frac{11}{15} & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & 1 & 1 \\ 0 & 3 & \frac{1}{2} & \frac{3}{2} \\ 0 & 0 & \frac{5}{2} & \frac{1}{2} \\ 0 & 0 & 0 & -\frac{28}{15} \end{bmatrix} . \tag{45}$$

Widać zatem, że

1. Faktoryzacja LU nie wymaga *de facto* rozwiązywania skomplikowanego układu równań, jako że każde z rozwiązywanych równań jest równaniem z jedną niewiadomą, jeśli tylko macierz $\mathbf{A}$ jest przeglądana we właściwej kolejności. Obliczenie jednego elementu wymaga $\sim N$ operacji, wszystkich elementów jest N^2, zatem koszt obliczeniowy faktoryzacji LU jest rzędu $O(N^3)$.

2. Macierz $\mathbf{A}$ można przeglądać kolumnami poczynając od lewego górnego rogu, lecz jeszcze bardziej naturalna jest następująca kolejność:

   (a) Zaczynamy od lewego górnego rogu.

(b) Przeglądając k–tą kolumnę od pozycji diagonalnej w dół obliczamy wszystkie iloczyny $l_{kk}u_{kk}$, $l_{k+1,k}u_{kk}$, ..., $l_{Nk}u_{kk}$ *bez wykonywania dzielenia* przez u_{kk}. Jako element podstawowy wybieramy ten z nich, który ma największą (na moduł) wartość — w tym celu przestawiamy odpowiednie wiersze $\mathbf{A}$ *oraz* odpowiednie elementy $\mathbf{L}$ stojące w już obliczonych kolumnach $(1, \ldots, k-1)$. Teraz wykonujemy dzielenie przez nowe u_{kk} $(l_{kk} = 1)$. Widać, że iloczynów $l_{sk}u_{kk}$, $s > k$, nie trzeba ponownie obliczać, ponieważ zostały policzone przed wybraniem elementu podstawowego.

(c) Po przejrzeniu k–tej kolumny przeglądamy k–ty wiersz poczynając od pozycji $k+1$ (poprzednie elementy tego wiersza zostały już obliczone przy okazji przeglądania poprzednich kolumn), jako że nie biorą one udziału w wyborze elementu podstawowego, wszystkie zaś elementy potrzebne do ich obliczenia są już w tym momencie znane.

3. Na skutek zastosowania wyboru elementu podstawowego dostajemy **nie** faktoryzację wyjściowej macierzy $\mathbf{A}$, lecz faktoryzację macierzy różniącej się od macierzy wyjściowej kolejnością wierszy (porównaj lewe strony (36) i (45)). Trzeba zapamiętać tę permutację wierszy, jako że przy rozwiązywaniu równania $\mathbf{Ax} = \mathbf{b}$ trzeba zastosować tę samą permutację elementów wektora $\mathbf{b}$.

Algorytm Thomasa

Można łatwo zauważyć, że faktoryzacji LU macierzy trójdiagonalnej można dokonać w czasie liniowym. Istotnie, gdy obliczamy elementy $l_{ij}, i > j$, widzimy, że dla takiej macierzy tylko $l_{n,n+1} \neq 0$ — pozostałych elementów nie trzeba więc obliczać, skoro z góry wiadomo, że znikają. Czynnik L jest dwudiagonalny, podobnie dwudiagonalny jest czynnik U, a zatem także forward substitution i backsubstitution można wykonać w czasie liniowym. Uwaga: Ze względu na konieczność zachowania kształtu macierzy trójdiagonalnej, niemożliwy jest przy tym wybór elementu podstawowego.

Złożoność obliczeniowa rozwiązywania układu równań z macierzą trójdiagonalną wynosi $O(N)$.

Algorytm faktoryzacji LU macierzy trójdiagonalnej, wraz z forward substitution i backsubstitution, nosi nazwę algorytmu Thomasa.