

Metody numeryczne

Faktoryzacje

P. F. Góra

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

2020

Kiedy eliminacja Gaussa nie wystarcza

Eliminacji Gaussa warto używać do rozwiązywania pojedynczego równania

$$Ax = b, \quad \det A \neq 0 \quad (1)$$

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 wszakże, że kolejne kroki eliminacji Gaussa zależą wyłącznie od elementów macierzy 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 (kolumn wyrazów wolnych) do czasu, gdy będzie to *naprawdę* potrzebne.

Takimi algorytmami są *faktoryzacje*, czyli przedstawienie macierzy A jako iloczynu dwu macierzy w jakimś sensie prostszych:

$$A = Y \cdot Z. \quad (2)$$

Faktoryzacja LU

Przypuśćmy, że udało nam się znaleźć faktoryzację

$$\mathbf{A} = \mathbf{L} \cdot \mathbf{U}, \quad (3)$$

gdzie macierz \mathbf{U} jest trójkątna górna (wszystkie elementy poniżej głównej przekątnej są zerami), natomiast \mathbf{L} jest trójkątna 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

$$\mathbf{Ax} \equiv \mathbf{L} \underbrace{\mathbf{Ux}}_{\mathbf{y}} = \mathbf{b} \quad (4)$$

rozwiązujemy jako

$$\mathbf{Ly} = \mathbf{b} \quad (5a)$$

$$\mathbf{Ux} = \mathbf{y} \quad (5b)$$

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 (4) wynosi $O(2N^2)$.

Pozostaje jeszcze “tylko” dokonać samej faktoryzacji.

Algorytm Doolittle'a

Aby dokonać faktoryzacji LU , należy obliczyć N^2 nieznanych elementów macierzy L , U . Rozpiszmy (3):

$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1N} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2N} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3N} \\ \dots & \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & a_{N3} & \dots & a_{NN} \end{bmatrix}}_A = \underbrace{\begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \dots & \dots & \dots & \dots & \\ l_{N1} & l_{N2} & l_{N3} & \dots & 1 \end{bmatrix}}_L \underbrace{\begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1N} \\ & u_{22} & u_{23} & \dots & u_{2N} \\ & & u_{33} & \dots & u_{3N} \\ & & & \dots & \dots \\ & & & & u_{NN} \end{bmatrix}}_U \quad (6)$$

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 A .

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

$$\begin{aligned} u_{11} &= a_{11} \\ l_{21}u_{11} &= a_{21} \\ l_{31}u_{11} &= a_{31} \\ \dots &\dots \dots \\ l_{N1}u_{11} &= a_{N1} \end{aligned} \tag{7}$$

Z pierwszego z równań (7) obliczamy u_{11} , a następnie z kolejnych $l_{21}, l_{31}, \dots, l_{N1}$.

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

$$\begin{array}{rcccccc}
 & & & & u_{12} & = & a_{12} \\
 l_{21}u_{12} & + & & & u_{22} & = & a_{22} \\
 l_{31}u_{12} & + & l_{32}u_{22} & = & a_{32} & & \\
 \dots & \dots & \dots & \dots & \dots & & \\
 l_{N1}u_{12} & + & l_{N2}u_{22} & = & a_{N2} & &
 \end{array} \tag{8}$$

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 (8) obliczamy u_{22} , a z kolejnych $l_{32}, l_{42}, \dots, l_{N2}$.

I tak dalej.

Widać, że średni koszt obliczenia któregoś z nieznanymi elementami 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 (5), jest taki sam, jak eliminacji Gaussa.

Przewaga faktoryzacji LU nad eliminacją Gaussa polega na tym, iż **przy pomocy faktoryzacji 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 L (bez diagonal) zapamiętujemy w poddiagonalnym trójkącie macierzy A , elementy macierzy U — na diagonal i w ponaddiagonalnym trójkącie A .

Przykład

W celu dokonania faktoryzacji LU macierzy

$$\begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix} \quad (9)$$

musimy rozwiązać równania

$$\begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & & \\ l_{21} & 1 & \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ & u_{22} & u_{23} \\ & & u_{33} \end{bmatrix} \quad (10)$$

ze względu na l_{ik} , u_{kj} . W tym celu zapiszmy indywidualne równania, na jakie rozpada się (10), w kolejności odpowiadające przeglądaniu macierzy (9) kolumnami.

Pierwsza kolumna macierzy (9) odpowiada

$$u_{11} = 1 \quad (11a)$$

$$l_{21}u_{11} = 2 \quad (11b)$$

$$l_{31}u_{11} = 2 \quad (11c)$$

skąd natychmiast otrzymujemy

$$u_{11} = 1, \quad l_{21} = 2, \quad l_{31} = 2. \quad (12)$$

Zwróćmy uwagę, iż pierwsze z równań (11) służy do wyliczenia elementu macierzy U , drugie i trzecie — do wyliczenia elementów macierzy L .

Druga kolumna odpowiada

$$u_{12} = 2 \quad (13a)$$

$$l_{21}u_{12} + u_{22} = 1 \quad (13b)$$

$$l_{31}u_{12} + l_{32}u_{22} = 2 \quad (13c)$$

Zauważmy, że jeśli równania (13) 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

$$u_{12} = 2, \quad u_{22} = -3, \quad l_{32} = \frac{2}{3}. \quad (14)$$

Trzecia kolumna (9) daje

$$u_{13} = 2 \quad (15a)$$

$$l_{21}u_{13} + u_{23} = 2 \quad (15b)$$

$$l_{31}u_{13} + l_{32}u_{23} + u_{33} = 1 \quad (15c)$$

W tym wypadku wszystkie trzy równania (15) 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

$$u_{13} = 2, \quad u_{23} = -2, \quad u_{33} = -\frac{5}{3}. \quad (16)$$

Ostatecznie

$$\begin{bmatrix} 1 & & & \\ 2 & 1 & & \\ 2 & \frac{2}{3} & 1 & \end{bmatrix} \begin{bmatrix} 1 & 2 & 2 \\ & -3 & -2 \\ & & -\frac{5}{3} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix}. \quad (17)$$

Równość w (17) 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 symetrię faktoryzacji, tylko częściowy wybór elementu podstawowego jest możliwy. Omówimy to na przykładzie. Rozwiązując równania (8) począwszy od drugiego z nich, obliczamy

$$\begin{aligned}l_{22}u_{22} &= a_{22} - l_{21}u_{12} && (l_{22} \equiv 1) \\l_{32}u_{22} &= a_{32} - l_{31}u_{12} \\&\dots \\l_{N2}u_{22} &= a_{N2} - l_{N1}u_{12}\end{aligned}\tag{18}$$

Porównujemy teraz wyliczone lewe strony równań (18) i wybieramy największą (na moduł) z nich; tę uznajemy za “właściwe” u_{22} — odpowiada to permutacji wierszy macierzy A . **Należy także spermutować już obliczone wiersze macierzy L .** W rezultacie otrzymujemy faktoryzację LU nie

samej macierzy A , ale macierzy różniącej się od niej pewną permutacją wierszy.

Przykład

Rozpatrzmy problem znalezienia następującej faktoryzacji:

$$\begin{bmatrix} 2 & 4 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 0 & 1 & 2 & -1 \\ -1 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}. \quad (19)$$

Faktoryzację znajdujemy przechodząc macierz A *kolumnami*, poczynając od lewego górnego rogu. Pierwsza kolumna daje zatem

$$a_{11} : u_{11} = 2 \quad (20a)$$

$$a_{21} : l_{21}u_{11} = 1 \quad (20b)$$

$$a_{31} : l_{31}u_{11} = 0 \quad (20c)$$

$$a_{41} : l_{41}u_{11} = -1 \quad (20d)$$

Po przejściu pierwszej kolumny faktoryzacja ma postać

$$\begin{bmatrix} 2 & 4 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 0 & 1 & 2 & -1 \\ -1 & 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ 0 & l_{32} & 1 & 0 \\ -\frac{1}{2} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} 2 & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix} \cdot \quad (21)$$

Przystępujemy do przeglądania drugiej kolumny:

$$a_{12} : \quad u_{12} = 4 \quad (22a)$$

$$a_{22} : \quad \frac{1}{2} \cdot 4 + u_{22} = 2 \implies u_{22} = 0 \quad (22b)$$

$$a_{32} : \quad 0 \cdot 4 + l_{32}u_{22} = 1 \implies l_{32}u_{22} = 1 \quad (22c)$$

$$a_{42} : \quad -\frac{1}{2} \cdot 4 + l_{42}u_{22} = 1 \implies l_{42}u_{22} = 3 \quad (22d)$$

Widać, iż równań (22) 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 (22), jako że ta wartość stanie

się nowym elementem diagonalnym, przez który będziemy dzielić. W naszym przykładzie jest to wiersz czwarty. Zatem

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

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

$$a_{22} : -\frac{1}{2} \cdot 4 + u_{22} = 1 \implies u_{22} = 3 \quad (24a)$$

$$a_{32} : 0 \cdot 4 + l_{32}u_{22} = 1 \implies l_{32} = \frac{1}{3} \quad (24b)$$

$$a_{42} : \frac{1}{2} \cdot 4 + l_{42}u_{22} = 2 \implies l_{42} = 0 \quad (24c)$$

a zatem

$$\begin{bmatrix} 2 & 4 & 1 & 1 \\ -1 & 1 & 0 & 1 \\ 0 & 1 & 2 & -1 \\ 1 & 2 & 3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ 0 & \frac{1}{3} & 1 & 0 \\ \frac{1}{2} & 0 & l_{43} & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & u_{13} & u_{14} \\ 0 & 3 & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}. \quad (25)$$

Przystępujemy do przeglądania trzeciej kolumny.

$$a_{13} : u_{13} = 1 \quad (26a)$$

$$a_{23} : -\frac{1}{2} \cdot u_{13} + u_{23} = 0 \implies u_{23} = \frac{1}{2} \quad (26b)$$

$$a_{33} : 0 \cdot u_{13} + \frac{1}{3} \cdot u_{23} + u_{33} = 2 \implies u_{33} = \frac{11}{6} \quad (26c)$$

$$a_{43} : \frac{1}{2} \cdot u_{13} + 0 \cdot u_{23} + l_{43}u_{33} = 3 \implies l_{43}u_{33} = \frac{5}{2} \quad (26d)$$

W tym wypadku nie *musimy* permutować wierszy (równania (26) 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} , przedstawiając jednocześnie *już obliczone* elementy macierzy \mathbf{L} .

A zatem

$$\begin{bmatrix} 2 & 4 & 1 & 1 \\ -1 & 1 & 0 & 1 \\ 1 & 2 & 3 & 1 \\ 0 & 1 & 2 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & 0 & 1 & 0 \\ 0 & \frac{1}{3} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & 1 & u_{14} \\ 0 & 3 & \frac{1}{2} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}. \quad (27)$$

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

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

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 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 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 $l_{kk}u_{kk}, l_{k+1,k}u_{kk}, \dots, l_{Nk}u_{kk}$ *bez wykonywania dzielenia przez u_{kk}* . Jako element podstawowy wybieramy ten z nich, który ma największą (na moduł) wartość — w tym celu przedstawiamy odpowiednie wiersze A oraz odpowiednie elementy L stojące w już obliczonych kolumnach $(1, \dots, k-1)$. 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 przejrzaniu 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 A , lecz faktoryzację macierzy różniącej się od macierzy wyjściowej kolejnością wierszy (porównaj lewe strony (19) i (28)). Trzeba zapamiętać tę permutację wierszy, jako że przy rozwiązywaniu równania $Ax = b$ trzeba zastosować tę samą permutację elementów wektora b .

Algorytm Thomasa

Można łatwo zauważyć, że faktoryzacji LU macierzy trójdzielnej 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. Czynniki L jest dwudzielny, podobnie dwudzielny jest czynnik U , a zatem także forward substitution i backsubstitution można wykonać w czasie liniowym. Uwaga: Ze względu na konieczność zachowania kształtu macierzy trójdzielnej, niemożliwy jest przy tym wybór elementu podstawowego.

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

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

Faktoryzacja Cholesky'ego

Niech $\mathbf{A} \in \mathbb{R}^{N \times N}$ będzie symetryczna, $\mathbf{A}^T = \mathbf{A}$, i dodatnio określona:

$$\forall \mathbf{x} \in \mathbb{R}^N, \mathbf{x} \neq 0: \mathbf{x}^T \mathbf{A} \mathbf{x} > 0. \quad (29)$$

Wówczas istnieje alternatywa dla faktoryzacji LU : faktoryzacja postaci

$$\mathbf{A} = \mathbf{C} \mathbf{C}^T, \quad (30)$$

gdzie \mathbf{C} jest macierzą trójkątną dolną o elementach diagonalnych większych od zera. Znalezienie faktoryzacji Cholesky'ego jest mniej więcej o połowę szybsze, niż znalezienie faktoryzacji LU tej samej macierzy.

Najprostszy algorytm jest bardzo podobny do algorytmu Doolittle'a:

$$\underbrace{\begin{bmatrix} c_{11} & & & & \\ c_{21} & c_{22} & & & \\ c_{31} & c_{32} & c_{33} & & \\ c_{41} & c_{42} & c_{43} & c_{44} & \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_C \underbrace{\begin{bmatrix} c_{11} & c_{21} & c_{31} & c_{41} & \dots \\ & c_{22} & c_{32} & c_{42} & \dots \\ & & c_{33} & c_{43} & \dots \\ & & & c_{44} & \dots \\ & & & & \ddots \end{bmatrix}}_{C^T} = \underbrace{\begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} & \dots \\ a_{21} & a_{22} & a_{32} & a_{42} & \dots \\ a_{31} & a_{32} & a_{33} & a_{43} & \dots \\ a_{41} & a_{42} & a_{43} & a_{44} & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_A \quad (31)$$

Pierwsza kolumna macierzy A daje

$$\begin{aligned}
 c_{11}^2 &= a_{11} \\
 c_{21}c_{11} &= a_{21} \\
 c_{31}c_{11} &= a_{31} \\
 c_{41}c_{11} &= a_{41} \\
 &\dots\dots\dots
 \end{aligned} \quad (32)$$

Z pierwszego z tych równań obliczamy c_{11} , z kolejnych c_{21}, c_{31} itd.

Druga kolumna daje

$$\begin{aligned}c_{11}c_{21} &= a_{21} \\c_{21}^2 + c_{22}^2 &= a_{22} \\c_{31}c_{21} + c_{32}c_{22} &= a_{32} \\c_{41}c_{21} + c_{42}c_{22} &= a_{42} \\&\dots\dots\dots\end{aligned}\tag{33}$$

Pierwsze z równań (33) jest identyczne z drugim z równań (32). Drugie z równań (33) pozwala na wyliczenie c_{22} . Dalsze równania pozwalają wyliczyć c_{32} , c_{42} itd.

I tak dalej.

Z uwagi na symetrię problemu, przy obliczaniu faktoryzacji Cholesky'ego nie jest możliwy wybór elementów podstawowych. Z uwagi na kolejność obliczeń, obliczone czynniki Cholesky'ego można przechowywać w tym samym miejscu, co elementy pierwotnej macierzy A .

Faktoryzacja LDL

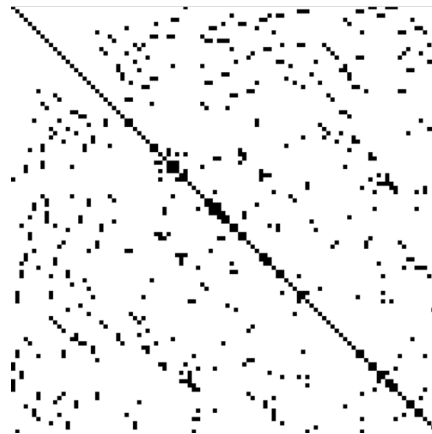
Jeżeli macierz spełnia założenia potrzebne do przeprowadzenia faktoryzacji Cholesky'ego, można także znaleźć jej inną faktoryzację:

$$\mathbf{A} = \mathbf{L} \mathbf{D} \mathbf{L}^T, \quad (34)$$

gdzie \mathbf{L} jest macierzą trójkątną dolną o tej własności, że $\forall i: l_{ii} = 1$, natomiast \mathbf{D} jest macierzą diagonalną o dodatnich elementach. Zaletą faktoryzacji LDL w stosunku do faktoryzacji Cholesky'ego jest to, iż do znalezienia LDL nie potrzeba pierwiastkowań.

Macierze rzadkie

W wielu praktycznych zastosowaniach występują **macierze rzadkie**, to znaczy takie, w których liczba elementów niezerowych rośnie wolniej niż N^2 , gdzie N jest wymiarem macierzy. Na przykład w macierzy trójdzielnej liczba niezerowych elementów skaluje się jak $O(3N)$, a w macierzy pasmowej o P dodatkowych diagonalach jak $O((2P + 1)N)$. Możliwe są także inne struktury macierzy rzadkich.



Macierze rzadkie i efektywność numeryczna

Dla efektywności numerycznej jest niesłuchanie ważne, aby zastosowany **algorytm** uwzględniał strukturę macierzy, tak, aby nie trzeba było wykonywać redundantnych mnożeń przez zero i dodawań zera, a nawet żeby nie przechodzić przez zerowe elementy.

- Dla macierzy trójdzielnej faktoryzacji LU dokonujemy w czasie liniowym, $O(N)$, ale za to niemożliwy jest wybór elementu podstawowego.
- Jeżeli możliwa jest faktoryzacja Cholesky'ego macierzy M -diagonalnej, także jej czynnik Cholesky'ego będzie M -diagonalny. Może jednak pojawić się niekorzystne zjawisko, zwane **wypełnieniem**: Jeżeli sama macierz ma zera “wewnątrz” pasma, jej czynnik Cholesky'ego nie musi ich mieć, co może bardzo niekorzystnie wpłynąć na wydajność numeryczną.

Przykład

Czynnik Cholesky'ego następującej macierzy *rzadkiej*

$$\begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet & \dots \\ \bullet & \bullet & & & & \\ \bullet & & \bullet & & & \\ \bullet & & & \bullet & & \\ \bullet & & & & \bullet & \\ \vdots & & & & & \ddots \end{bmatrix}$$

(35)

(niewypełnione elementy są zerami) będzie macierzą *pełną*.

Minimum Degree Algorithm*

Niech $\mathbf{A} \in \mathbb{R}^{N \times N}$ będzie macierzą posiadającą faktoryzację Cholesky'ego, dla której zachodzi niebezpieczeństwo pojawienia się wypełnienia. Wypełnienie zależy od struktury macierzy, nie od wartości jej poszczególnych elementów. Zamiast równania

$$\mathbf{Ax} = \mathbf{b} \quad (36a)$$

możemy rozwiązywać równanie

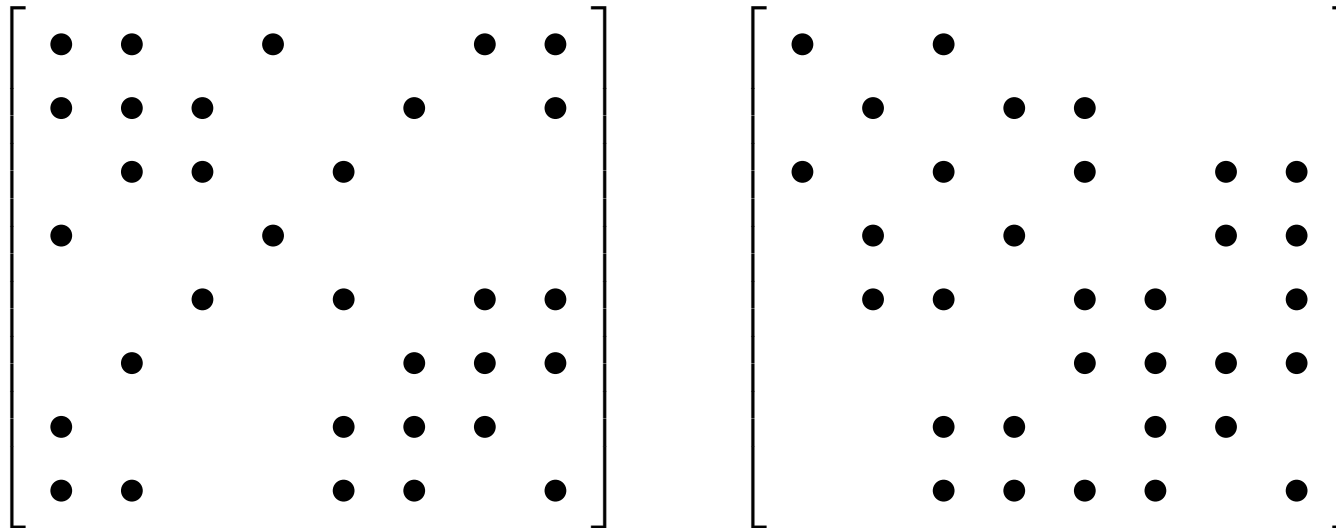
$$(\mathbf{PAP}^T) (\mathbf{Px}) = \mathbf{Pb} \quad (36b)$$

gdzie \mathbf{P} jest macierzą permutacji. (Jest to macierz ortogonalna.) Jeśli \mathbf{A} jest symetryczna i dodatnio określona, także macierz \mathbf{PAP}^T jest symetryczna i dodatnio określona, a więc posiada ona swoją faktoryzację Cholesky'ego. Macierz \mathbf{P} staramy się dobrać tak, aby wypełnienie w czynniku Cholesky'ego spemutowanej macierzy było możliwie małe. Problem

znalezienia permutacji takiej, aby wypełnienie było najmniejsze z możliwych jest NP-zupełny, w praktyce do poszukiwania P posługujemy się algorytmami heurystycznymi. Z historycznych powodów, z uwagi na związek pomiędzy macierzami symetrycznymi a grafami (struktura zerowych/niezerowych elementów pozadiagonalnych macierzy symetrycznej odpowiada macierzy sąsiedztwa pewnej klasy grafów nieskierowanych), algorytmy te nazywa się *minimum degree algorithms*. Przegląd tych algorytmów wykracza, niestety, poza zakres wykładu ze *wstępu* do metod numerycznych.

Jeżeli znalezienie efektywnej permutacji nie wydaje się tanie i wygodne, można rozważyć użycie zupełnie innej klasy algorytmów, na przykład algorytmów iteracyjnych.

Przykład



Macierz po prawej stronie stanowi optymalną permutację macierzy po lewej. W obu macierzach 32 elementy (dokładnie połowa) jest pusta. Gdyby zastosować faktoryzację Cholesky'ego do macierzy po lewej, czynnik trójkątny miałby tylko 4 (zamiast 16) elementów zerowych. Czynnik Cholesky'ego macierzy po prawej będzie miał 13 elementów zerowych (trzy elementy się wypełnią).

Faktoryzacja QR

Innym rodzajem faktoryzacji jest faktoryzacja QR : daną macierz $A \in \mathbb{R}^{N \times N}$ przedstawiam w postaci iloczynu

$$A = QR, \quad (37)$$

gdzie Q jest macierzą ortogonalną, a R jest macierzą trójkątną górną. Złożoność obliczeniowa faktoryzacji QR wynosi dla macierzy pełnych $O(N^3)$, czyli tyle samo, co faktoryzacji LU , jednak współczynnik przy wyrazie wiodącym jest gorszy niż dla LU . QR nie jest więc metodą “z wyboru” rozwiązywania układów równań liniowych. Jeśli jednak z jakichś względów

faktoryzację QR możemy łatwo (lub musimy) obliczyć, układ równań liniowych rozwiązujemy jak następuje:

$$\mathbf{Ax} = \mathbf{b} \quad (38a)$$

$$\mathbf{QRx} = \mathbf{b} \quad (38b)$$

$$\mathbf{Rx} = \mathbf{Q}^T \mathbf{b} \quad (38c)$$

Koszt obliczeniowy przejścia od (38b) do (38c) wynosi $O(N^2)$. Równanie (38c) rozwiązujemy metodą *backsubstitution*, co także kosztuje $O(N^2)$. Jest to więc koszt mały w porównaniu z dokonaniem samej faktoryzacji.

Pozostaje pytanie: Jak dokonać tej faktoryzacji?

Transformacja Householdera

Niech $\mathbf{u} \in \mathbb{R}^N$, $\mathbf{u} \neq 0$. Tworzymy macierz

$$\mathbf{P} = \mathbb{I} - 2 \frac{\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|^2}. \quad (39)$$

W sposób oczywisty $\mathbf{P}^T = \mathbf{P}$. Obliczmy

$$\begin{aligned} \mathbf{P}^2 &= \left(\mathbb{I} - 2 \frac{\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|^2} \right) \left(\mathbb{I} - 2 \frac{\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|^2} \right) \\ &= \mathbb{I} - 2 \frac{\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|^2} - 2 \frac{\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|^2} + 4 \frac{\overbrace{\mathbf{u}\mathbf{u}^T\mathbf{u}\mathbf{u}^T}^{\|\mathbf{u}\|^2}}{\|\mathbf{u}\|^4} \\ &= \mathbb{I} - 4 \frac{\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|^2} + 4 \frac{\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|^2} \\ &= \mathbb{I} \end{aligned} \quad (40)$$

Skoro $\mathbf{P}^T = \mathbf{P}$ oraz $\mathbf{P}^2 = \mathbb{I}$, $\mathbf{P}^T = \mathbf{P}^{-1}$: macierz (39) jest macierzą symetryczną, rzeczywistą oraz ortogonalną. Macierz taką nazywamy ortogonalną macierzą rzutową.

Niech teraz w (39)

$$\mathbf{u} = \mathbf{x} \mp \|\mathbf{x}\| \hat{\mathbf{e}}_1, \quad (41)$$

gdzie $\hat{\mathbf{e}}_1$ jest pierwszym wektorem jednostkowym. Macierz (39) wraz z (41) nazywam *macierzą Householdera*. Obliczam

$$\mathbf{P}\mathbf{x} = \mathbf{x} - \frac{2\mathbf{u}\mathbf{u}^T\mathbf{x}}{\|\mathbf{u}\|^2}. \quad (42a)$$

Zauważmy, że $\mathbf{u}^T\mathbf{x} = \mathbf{x}^T\mathbf{x} \mp \|\mathbf{x}\| \hat{\mathbf{e}}_1^T\mathbf{x} = \|\mathbf{x}\|^2 \mp \|\mathbf{x}\| x_1$, gdzie x_1 jest pierwszą składową wektora \mathbf{x} .

Analogicznie $\|\mathbf{u}\|^2 = (\mathbf{x} \mp \|\mathbf{x}\| \hat{\mathbf{e}}_1)^T (\mathbf{x} \mp \|\mathbf{x}\| \hat{\mathbf{e}}_1) = \mathbf{x}^T \mathbf{x} \mp \|\mathbf{x}\| x_1 \mp \|\mathbf{x}\| x_1 + \|\mathbf{x}\|^2 = 2 (\|\mathbf{x}\|^2 \mp \|\mathbf{x}\| x_1)$. Wobec tego

$$\mathbf{P}_\mathbf{x} = \mathbf{x} - \frac{2\mathbf{u} (\|\mathbf{x}\|^2 \mp \|\mathbf{x}\| x_1)}{2 (\|\mathbf{x}\|^2 \mp \|\mathbf{x}\| x_1)} = \mathbf{x} - \mathbf{u} = \pm \|\mathbf{x}\| \hat{\mathbf{e}}_1. \quad (42b)$$

Efektem działania macierzy Householdera na wskazany wektor jest wyzerowanie wszystkich jego składowych, poza pierwszą, i “przelanie” całej jego długości na pierwszą składową. Złożoność obliczeniowa tej procedury wynosi $O(N)$.

Faktoryzacja QR

Niech $A \in \mathbb{R}^{N \times N}$. Niech P_1 oznacza transformację Householdera zbudowaną na pierwszej kolumnie macierzy A . Otrzymujemy

$$P_1 A = A_1 \quad (43)$$
$$P_1 \begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \cdots \\ \bullet & \bullet & \bullet & \bullet & \cdots \\ \bullet & \bullet & \bullet & \bullet & \cdots \\ \bullet & \bullet & \bullet & \bullet & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} = \begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \cdots \\ & \bullet & \bullet & \bullet & \cdots \\ & \bullet & \bullet & \bullet & \cdots \\ & \bullet & \bullet & \bullet & \cdots \\ & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Transformacja Householdera P_1 wyzerowała pierwszą kolumnę macierzy A , za wyjątkiem elementu diagonalnego. Złożoność obliczeniowa tego kroku wynosi $O(N^2)$ (transformacja Householdera działa na N kolumn).

Niech teraz

$$\mathbf{P}_2 = \left[\begin{array}{c|cccc} 1 & 0 & 0 & \dots & 0 \\ \hline 0 & & & & \\ 0 & & & & \\ 0 & & & & \\ 0 & & & & \end{array} \right] \quad (44)$$

$(N-1)\mathbf{P}_2$

gdzie $(N-1)\mathbf{P}_2 \in \mathbb{R}^{(N-1) \times (N-1)}$ jest transformacją Householdera, zbudowaną na drugiej kolumnie macierzy \mathbf{A}_1 , poczynając od elementu diagonalnego w dół. Otrzymujemy

$$\mathbf{P}_2\mathbf{P}_1\mathbf{A} = \mathbf{P}_2\mathbf{A}_1 = \mathbf{A}_2 = \left[\begin{array}{ccccc} \bullet & \bullet & \bullet & \bullet & \dots \\ & \bullet & \bullet & \bullet & \dots \\ & & \bullet & \bullet & \dots \\ & & \bullet & \bullet & \dots \\ & & \vdots & \vdots & \ddots \end{array} \right] \quad (45)$$

Następnie definiuję

$$\mathbf{P}_3 = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ \hline 0 & 0 & & & & \\ 0 & 0 & & & & \\ 0 & 0 & & & & \\ 0 & 0 & & & & \end{array} \right] \quad (46)$$

gdzie $(N-2)\mathbf{P}_3 \in \mathbb{R}^{(N-2) \times (N-2)}$ jest transformacją Householdera, zbudowaną na trzeciej kolumnie macierzy \mathbf{A}_2 , poczynając od elementu diagonalnego w dół. Stojąca w lewym górnym rogu macierz jednostkowa służy do tego, żeby nie zepsuć struktury, którą osiągnęliśmy w poprzednich kro-

kach. Otrzymujemy

$$\mathbf{P}_3\mathbf{P}_2\mathbf{P}_1\mathbf{A} = \mathbf{P}_3\mathbf{A}_2 = \mathbf{A}_3 = \begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \cdots \\ & \bullet & \bullet & \bullet & \cdots \\ & & \bullet & \bullet & \cdots \\ & & & \bullet & \cdots \\ & & & \vdots & \ddots \end{bmatrix} \quad (47)$$

Widać, że po $N-1$ krokach osiągnę

$$\mathbf{R} = \begin{bmatrix} \bullet & \bullet & \bullet & \cdots & \bullet \\ & \bullet & \bullet & \cdots & \bullet \\ & & \bullet & \cdots & \bullet \\ & & & \ddots & \vdots \\ & & & & \bullet \end{bmatrix} = \underbrace{\mathbf{P}_{N-1}\mathbf{P}_{N-2}\cdots\mathbf{P}_1}_{\mathbf{Q}^T} \mathbf{A} \quad (48)$$

\mathbf{R} jest macierzą trójkątną górną. Ponieważ macierze \mathbf{P}_i są ortogonalne, ich iloczyn, oznaczony przez \mathbf{Q}^T , także jest macierzą ortogonalną. Nie

musimy zapamiętywać poszczególnych macierzy P_i , wystarczy zapamiętać ich iloczyn.

Otrzymaliśmy zatem dla dowolnej macierzy kwadratowej *faktoryzację na macierz ortogonalną i trójkątną górną*:

$$A = QR, \quad (49)$$

czyli poszukiwaną faktoryzację QR .

Nie jest to jedyny algorytm pozwalający uzyskać faktoryzację QR , pokazuje jednak, że jest ona w ogólnym (czyli dowolnym) wypadku możliwa.

Obroty Givensa

Transformacja Householdera służy do zerowania wielu składowych jakiegoś wektora. Jeżeli chcemy selektywnie wyzerować jakieś składowe — lub jeśli interesujący nas wektor ma jakąś szczególną postać — bardziej efektywne od transformacji Householdera będą *obroty Givensa*.

Niech x będzie pewnym wektorem i niech $y = G(i, j)x$. Składowe wektora y wynoszą

$$y_k = \begin{cases} cx_i + sx_j & k = i \\ -sx_i + cx_j & k = j \\ x_k & \text{poza tym} \end{cases} \quad (51)$$

Zażądajmy, aby $y_j = 0$. Widać, że musi zachodzić

$$c = \frac{x_i}{\sqrt{x_i^2 + x_j^2}}, \quad s = \frac{x_j}{\sqrt{x_i^2 + x_j^2}}. \quad (52)$$

Obrót Givensa (50) wraz z warunkami (52) zeruje j -tą składową wybranego wektora. Składowa i -ta przybiera wartość $\sqrt{x_i^2 + x_j^2}$.

Faktoryzacja QR macierzy trójdzielnej symetrycznej

Rozpatrzmy macierz $\mathbf{A} \in \mathbb{R}^{N \times N}$, trójdzielną symetryczną

$$\mathbf{A} = \begin{bmatrix} a & b & & & & \\ b & d & e & & & \\ & e & f & g & & \\ & & g & h & l & \\ & & & \ddots & \ddots & \ddots \end{bmatrix} \quad (53)$$

Zadziałajmy na nią macierzą Givensa taką, aby zerowała drugi element pierwszej kolumny

$$\mathbf{G}_1 = \begin{bmatrix} c_1 & s_1 & & & & \\ -s_1 & c_1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & \ddots \end{bmatrix} \quad (54)$$

Macierz A_1 mnożymy przez macierz Givensa

$$G_2 = \begin{bmatrix} 1 & & & & & \\ & c_2 & s_2 & & & \\ & -s_2 & c_2 & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & \ddots \end{bmatrix} \quad (56)$$

dobraną tak, aby zerowała trzeci element drugiej kolumny macierzy A_1 . Pierwszy wiersz i pierwsza kolumna nie zmieniają się, podobnie jak wiersze począwszy od czwartego. W rezultacie macierz $A_2 = G_2 A_1 = G_2 G_1 A$ ma zera w poddiagonalnych miejscach dwu pierwszych kolumn. Ten krok także wymaga stałej, niezależnej od rozmiaru macierzy, liczby operacji.

W kolejnym kroku macierz A_2 mnożymy przez taką macierz Givensa, która wyzeruje czwarty element trzeciej kolumny. I tak dalej.

W ten sposób, po $N-1$ krokach, ponosząc koszt numeryczny $O(N)$ (stały koszt na krok, $\sim N$ kroków), otrzymujemy

$$\mathbf{G}_{N-1} \cdots \mathbf{G}_2 \mathbf{G}_1 \mathbf{A} = \mathbf{R} = \begin{bmatrix} \bullet & \bullet & \bullet & & & & \\ & \bullet & \bullet & \bullet & & & \\ & & \bullet & \bullet & \bullet & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \bullet & \bullet & \\ & & & & & \bullet & \bullet \\ & & & & & & \bullet \end{bmatrix}, \quad (57a)$$

czyli

$$\mathbf{A} = \underbrace{\mathbf{G}_1^T \mathbf{G}_2^T \cdots \mathbf{G}_{N-1}^T}_{\mathbf{Q}} \mathbf{R} \quad (57b)$$

Macierz \mathbf{Q} jest ortogonalna. Macierz \mathbf{R} jest trójkątna górna (tak naprawdę ma ona tylko dwie niezerowe diagonale nad diagonalą główną). Widzimy, że (57b) jest faktoryzacją QR macierzy trójdzielnej symetrycznej.

Zastosowanie do rozwiązywania układu równan liniowych

Jeżeli chcemy użyć obrotów Givensa do rozwiązania układu równań liniowych

$$\mathbf{Ax} = \mathbf{b}, \quad (58)$$

gdzie \mathbf{A} jest trójdagonalną macierzą symetryczną, postępując jak poprzednio otrzymujemy kolejno

$$\mathbf{G}_1 \mathbf{Ax} = \mathbf{G}_1 \mathbf{b} \quad (59a)$$

$$\mathbf{G}_2 \mathbf{G}_1 \mathbf{Ax} = \mathbf{G}_2 \mathbf{G}_1 \mathbf{b} \quad (59b)$$

...

$$\mathbf{G}_{N-1} \cdots \mathbf{G}_2 \mathbf{G}_1 \mathbf{Ax} \equiv \mathbf{Rx} = \mathbf{G}_{N-1} \cdots \mathbf{G}_2 \mathbf{G}_1 \mathbf{b}. \quad (59c)$$

Oczywiście istotne jest tylko równanie (59c) — lewych stron poprzednich równań nie musimy wyliczać. Każde kolejne mnożenie po stronie prawej

wykonujemy w stałym czasie, a więc do postaci $\mathbf{R}\mathbf{x} = \tilde{\mathbf{b}}$ dochodzimy w czasie $O(N)$. To równanie rozwiązujemy metodą *backsubstitution*, co, z uwagi na szczególną postać macierzy \mathbf{R} , także da się wykonać w czasie liniowym.

Przykład ten pokazuje, że możemy odnieść duży zysk na złożoności obliczeniowej, jeśli tylko dobierzemy odpowiedni algorytm odpowiadający strukturze — w tym wypadku rzadkości i symetryczności — macierzy.

Uwaga: Skumulowanej macierzy Givensa \mathbf{Q} nie musimy wyliczać w sposób jawny — gdybyśmy to chcieli zrobić, wymagałoby to $O(N^2)$ operacji.

Wzór Shermana-Morrisona

Twierdzenie: Niech $\mathbf{A} \in \mathbb{R}^{N \times N}$, $\det \mathbf{A} \neq 0$ oraz $\mathbf{u}, \mathbf{v} \in \mathbb{R}^N$. Niech $\mathbf{A}_1 = \mathbf{A} + \mathbf{u}\mathbf{v}^T$. Wówczas

$$\mathbf{A}_1^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}}. \quad (60)$$

Zauważmy, że ponieważ $\det \mathbf{A} \neq 0$, macierz \mathbf{A}^{-1} istnieje. Ponadto wyrażenie $\mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}$ jest *liczbą* (skalarem).

Wzór ten jest przydatny w przypadku, gdy chcemy wyliczyć \mathbf{A}_1^{-1} , gdzie \mathbf{A}_1 jest pewną szczególną modyfikacją macierzy \mathbf{A} , a odwrotność \mathbf{A}^{-1} znamy.

Przykład

Niech $\mathbf{u} = \mathbf{v} = [1, 0, 0, 0, 1]^T$. Wówczas

$$\mathbf{uv}^T = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} [1 \ 0 \ 0 \ 0 \ 1] = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (61)$$

Niech teraz

$$\mathbf{A} = \begin{bmatrix} 3 & 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 & 3 \end{bmatrix}, \quad \mathbf{A}_1 = \mathbf{A} + \mathbf{uv}^T = \begin{bmatrix} 4 & 1 & 0 & 0 & 1 \\ 1 & 4 & 1 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 1 & 4 & 1 \\ 1 & 0 & 0 & 1 & 4 \end{bmatrix}. \quad (62)$$

Dowód.

$$\begin{aligned}
 & (\mathbf{A} + \mathbf{u}\mathbf{v}^T) \left(\mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}} \right) \\
 = & \mathbf{A}\mathbf{A}^{-1} - \frac{1}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}} \mathbf{A}\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^T\mathbf{A}^{-1} + \mathbf{u}\mathbf{v}^T\mathbf{A}^{-1} \\
 & - \frac{1}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}} \mathbf{u} \underbrace{\mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}}_{\text{to jest liczba!}} \mathbf{v}^T\mathbf{A}^{-1} \\
 = & \mathbb{I} - \frac{1}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}} \mathbf{u}\mathbf{v}^T\mathbf{A}^{-1} + \mathbf{u}\mathbf{v}^T\mathbf{A}^{-1} - \frac{\mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}} \mathbf{u}\mathbf{v}^T\mathbf{A}^{-1} \\
 = & \mathbb{I} + \left(1 - \frac{1}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}} - \frac{\mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}}{1 + \mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}} \right) \mathbf{u}\mathbf{v}^T\mathbf{A}^{-1} = \mathbb{I}. \quad (63)
 \end{aligned}$$

□

Algorytm Shermana-Morrisona

Wzór Shermana-Morrisona (60) pozwala skonstruować odwrotność macierzy A_1 jeśli znamy odwrotność A . Jednak w praktyce prawie **nigdy nie konstruujemy jawnej odwrotności macierzy!** Jak więc zastosować ten wzór?

Zauważmy, że **zapewne** chcemy obliczyć jakieś $A_1^{-1}b$, gdzie b jest znanym wektorem, przy założeniu, że **łatwo** potrafimy obliczyć $A^{-1}b$. Interesuje nas znalezienie

$$w = A_1^{-1}b = \left(A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1}u} \right) b \quad (64)$$

Algorytm wygląda następująco:

(a) Rozwiąż równanie

$$\mathbf{Az} = \mathbf{b} \quad (65a)$$

(b) Rozwiąż równanie

$$\mathbf{Aq} = \mathbf{u} \quad (65b)$$

(c) Oblicz

$$\mathbf{w} = \mathbf{z} - \frac{\mathbf{v}^T \mathbf{z}}{1 + \mathbf{v}^T \mathbf{q}} \mathbf{q}. \quad (65c)$$

Problem sprowadza się więc do rozwiązania dwu równań (65a),(65b) z taką samą macierzą, które umiemy szybko rozwiązać, gdyż — na przykład — znamy faktoryzację macierzy \mathbf{A} . Zauważmy, że $\mathbf{v}^T \mathbf{A}^{-1} \mathbf{b} = \mathbf{v}^T \mathbf{z}$ jest *liczbą*.

Przykład (c.d.)

Pierwsza z macierzy (62) jest macierzą symetryczną, dodatnio określoną i trójdziagonalną, a więc jej czynnik Cholesky'ego ma tylko dwie niezerowe diagonale, a koszt jego wyliczenia jest rzędu $O(N)$. Czynnik Cholesky'ego drugiej z tych macierzy jest pełną macierzą trójkątną (nastąpi *wypełnienie*) i koszt jego wyliczenia jest rzędu $O(N^3)$ (wyobraźmy sobie, że zamiast o macierzach 5×5 , mówimy o macierzach 1000×1000). Zastosowanie algorytmu (65) redukuje problem do znalezienia i dwukrotnego zastosowania rzadkiego czynnika Cholesky'ego pierwszej z macierzy (62). Da się to zrobić w czasie liniowym.