Miriam Jańczak
numer albumu: 229761

3 stycznia 2018
prowadzący: dr hab. Paweł Zieliński

Obliczenia naukowe

Lista 5

Opis problemu

Rozwiązanie układu równań liniowych:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{1}$$

dla danej macierzy $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ i wektora prawych stron $\boldsymbol{b} \in \mathbb{R}^n$, gdzie $n \geqslant 4$.

Macierz $\boldsymbol{A}$ jest rzadką macierzą blokową o następującej strukturze:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_1 & \boldsymbol{C}_1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{B}_2 & \boldsymbol{A}_2 & \boldsymbol{C}_2 & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}_3 & \boldsymbol{A}_3 & \boldsymbol{C}_3 & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{B}_{v-2} & \boldsymbol{A}_{v-2} & \boldsymbol{C}_{v-2} & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B}_{v-1} & \boldsymbol{A}_{v-1} & \boldsymbol{C}_{v-1} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B}_v & \boldsymbol{A}_v \end{pmatrix}, \tag{2}$$

$v = \frac{n}{\ell}$, zakładając że n jest podzielne przez ℓ, gdzie $\ell \geqslant 2$ jest rozmiarem wszystkich kwadratowych macierzy wewnętrznych (bloków) $\boldsymbol{A}_i$, $\boldsymbol{B}_i$, $\boldsymbol{C}_i$, $\boldsymbol{0}$.

Macierze $\boldsymbol{A}_i$, $\boldsymbol{B}_i$, $\boldsymbol{C}_i$, $\boldsymbol{0}$ są następującej postaci:

(i) $\boldsymbol{A}_i \in \mathbb{R}^{\ell\times\ell}$, $i = 1,\dots,v$ – macierze gęste,

(ii) $\boldsymbol{0} \in \mathbb{R}^{\ell\times\ell}$ – macierz zerowa,

(iii) $\boldsymbol{B}_i \in \mathbb{R}^{\ell\times\ell}$, $i = 2,\dots,v$ – macierze z niezerowymi dwoma ostatnimi kolumnami:

$$\boldsymbol{B}_i = \begin{pmatrix} 0 & \cdots & 0 & b^i_{1\,\ell-1} & b^i_{1\,\ell} \\ 0 & \cdots & 0 & b^i_{2\,\ell-1} & b^i_{2\,\ell} \\ \vdots & & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & b^i_{\ell\,\ell-1} & b^i_{\ell\,\ell} \end{pmatrix}, \tag{3}$$

(iv) $\boldsymbol{C}_i \in \mathbb{R}^{\ell\times\ell}$, $i = 1,\dots,v-1$ – macierze diagonalne:

$$\boldsymbol{C}_i = \begin{pmatrix} c^i_1 & 0 & 0 & \cdots & 0 \\ 0 & c^i_2 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & c^i_{\ell-1} & 0 \\ 0 & \cdots & 0 & 0 & c^i_\ell \end{pmatrix}. \tag{4}$$

W celu rozwiązania układu równań liniowych $\boldsymbol{Ax} = \boldsymbol{b}$ (1) należało zastosować dwie metody:

(a) metodę eliminacji Gaussa w wersji bez wyboru elementu głównego oraz z częściowym wyborem elementu głównego,

(b) obliczyć rozkład $\boldsymbol{LU}$ macierzy $\boldsymbol{A}$ w wersji bez wyboru elementu głównego oraz z częściowym wyborem elementu głównego, a następnie rozwiązać układ $\boldsymbol{LUx} = \boldsymbol{b}$.

Sposób przechowywania macierzy

Macierz $\boldsymbol{A}$ dana w zadaniu posiada tylko $(\ell+3)n-3\ell$ elementów nie będących zerami – $v\cdot\ell^2$ w blokach $\boldsymbol{A}_i$, $(v-1)\cdot 2\ell$ w blokach $\boldsymbol{B}_i$ i $(v-1)\cdot\ell$ w blokach $\boldsymbol{C}_i$, co świadczy o tym, że $\boldsymbol{A}$ jest macierzą rzadką. Przechowywanie macierzy $\boldsymbol{A}$ w standardowy sposób (tablica dwuwymiarowa $n \times n$) byłoby więc dość nieefektywne. Aby temu zapobiec użyta została specjalna struktura do przechowywania macierzy rzadkich `SparseMatrixCSC` z języka `Julia`, w której macierze przechowywane są w skompresowanym porządku kolumnowym. W celu optymalizacji czasowej dostępu do elementów tak przechowywanej macierzy pod kątem zaimplementowanych algorytmów używano macierzy transponowanej, co jednak nie miało wpływu na ogólną ich postać, dlatego zostanie pominięte w rozważaniach.

Opis algorytmów

Metoda eliminacji Gaussa jest algorytmem mającym szerokie zastosowanie w rozwiązywaniu podstawowych problemów algebry liniowej takich jak rozwiązywanie układów równań liniowych, obliczanie rzędu macierzy, jej wyznacznika, macierzy odwrotnej czy rozkładu $\boldsymbol{LU}$ macierzy. Wykorzystuje ona elementarne operacje na macierzy takie jak mnożenie wiersza przez skalar czy odejmowanie od siebie dwóch wierszy.

Rozwiązywanie układów równań metodą eliminacji Gaussa

Opis działania

Zasadą działania *metody eliminacji Gaussa* przy rozwiązywaniu układów równań jest stopniowa eliminacja niewiadomych przez odpowiednie kombinowanie równań tak, aby zastąpić dany układ $\boldsymbol{Ax} = \boldsymbol{b}$ równoważnym mu układem z macierzą trójkątną górną.
W pierwszym kroku zostaje wyeliminowana niewiadoma x_1 z $n-1$ równań poprzez odejmowanie dla $i = 2, \cdots, n$ odpowiedniej krotności pierwszego równania od i-tego równania, aby wyzerować w nim współczynnik przy x_1. Takie postępowanie powtarzane jest dla kolejnych niewiadomych x_k, gdzie dla $i = k+1, \cdots, n$ od i-tego równania odejmowana jest odpowiednia krotność k-tego równania.

Aby możliwe było wykonanie powyższej procedury każdy z elementów diagonalnych w macierzy musi być różny od zera. W momencie kiedy tak nie jest potrzebna jest modyfikacja algorytmu, a mianowicie zamiana wiersza z zerowym elementem na diagonali z innym który w tym miejscu nie posiada zera, w praktyce w i-tym kroku algorytmu wyszukuje się w i-tej kolumnie element (zwany *elementem głównym*) o największej co do modułu wartości i wiersz z takim elementem zamienia się miejscem z i-tym wierszem. Taka zamiana zawsze jest możliwa, gdyż w przeciwnym przypadku macierz byłaby osobliwa.

Ostatnim krokiem jest rozwiązanie powstałego układu z macierzą trójkątną górna za pomocą *algorytmu podstawiania wstecz*. Polega on na obliczeniu:

$$x_i = \frac{b_i - \sum_{j=i+1}^{n}}{a_{ii}}$$

dla wierszy i od n do 1.

Metoda eliminacji Gaussa ma złożoność $O(n^3)$, a algorytm podstawiania wstecz $O(n^2)$. Zatem, aby rozwiązać układ równań, trzeba wykonać łącznie $O(n^3)$ operacji.

Zastosowane modyfikacje

Jak już zostało wspomniane, macierz $\boldsymbol{A}$ jest macierzą rzadką ponadto posiada ona specyficzną blokowo-trójdiagonalną postać (2) co umożliwia zredukowanie w znacznym stopniu liczby wykonywanych operacji w stosunku do metody eliminacji Gaussa stosowanej dla macierzy gęstych.

Zauważyć można, że postać macierzy $\boldsymbol{A}$ zapewnia, że wiele elementów znajdujących się pod diagonalą będzie zerami i nie będzie konieczne ich zerowanie.

Rozpatrując pierwszych $\ell-2$ kolumn widać że elementy niezerowe mogą znajdować się jedynie w bloku $\boldsymbol{A}_1$, a więc tylko w ℓ pierwszych rzędach. Idąc dalej, dla kolejnych ℓ kolumn wszystkie niezerowe elementy będą znajdować się najniżej w bloku $\boldsymbol{B}_2$ albo w bloku $\boldsymbol{A}_3$ – czyli 2ℓ pierwszych rzędach, a dla jeszcze następnych ℓ kolumn w blokach $\boldsymbol{B}_3$ i $\boldsymbol{A}_4$ – czyli 3ℓ pierwszych rzędach. Biorąc pod uwagę następne kolumny schemat będzie się powtarzał dając możliwość wyprowadzenia ogólnego wzoru na indeks ostatniego niezerowego elementu $e_{non\ 0}$ w danej kolumnie k:

$$e_{non\ 0}(k) = \min\left\{\ell + \ell \cdot \left\lfloor \frac{k+1}{\ell} \right\rfloor, n\right\} \tag{5}$$

Również, poza ostatnimi ℓ wierszami, w każdym wierszu ostatnim niezerowym elementem jest element leżący na diagonali bloku $\boldsymbol{C}_i$. Można zauważyć, że owe elementy znajdują się zawsze w odległości ℓ od elementów na diagonali macierzy $\boldsymbol{A}$. Natomiast dla ostatnich ℓ rzędów najbardziej wysunięte na prawo elementy niezerowe leżą w n-tej kolumnie. Powyższa obserwacja pozwala na wyprowadzenie wzoru tym razem na indeks kolumny k_{last}, w której znajduje się ostatni niezerowy element w rzędzie r:

$$k_{last}(r) = \min\{r + \ell, n\}. \tag{6}$$

Oczywiście, jeżeli w danym kroku metody eliminacji Gaussa r-ty rząd odejmowany jest od rzędów pod nim, nie jest konieczne modyfikowanie elementów w kolumnach o większych od $k_{last}(r)$ indeksach.

Metoda eliminacji Gaussa prowadzi do układu z macierzą trójkątną górną, który rozwiązywany jest za pomocą algorytmu podstawiania wstecz, który w tym przypadku także poddawany jest drobnym modyfikacjom w celu ograniczenia liczby wykonywanych operacji.

Warto zauważyć tutaj, że w wyniku eliminacji Gaussa poza elementami pod diagonalą bloków $\boldsymbol{C}_i$ w macierzy $\boldsymbol{A}$ nie powstały żadne nowe elementy niezerowe. Wystarczy zatem dla każdego wiersza sumować elementy tylko do pewnej kolumny określonej wyprowadzonym wcześniej wzorem (6).

Metodę eliminacji Gaussa z opisanymi modyfikacjami przedstawia Algorytm 1.

Zakładając, że ℓ jest stałą, złożoność obliczeniowa zmodyfikowanej metody eliminacji Gaussa, wynosi $O(n)$. Zewnętrzna pętla eliminacji Gaussa wykonuje $n-1$ przebiegów, środkowa maksymalnie 2ℓ, natomiast wewnętrzna maksymalnie ℓ. Z kolei w algorytmie podstawiania wstecz zewnętrzna pętla wykonuje n przebiegów, natomiast wewnętrzna maksymalnie ℓ. Jest to znacząca poprawa względem standardowej metody eliminacji Gaussa.

Algorytm 1: Eliminacja Gaussa

Dane wejściowe:

- $\boldsymbol{A}$ – dana w zadaniu macierz postaci (2),
- $\boldsymbol{b}$ – wektor prawych stron,
- n – rozmiar macierzy $\boldsymbol{A}$,
- ℓ – rozmiar bloku macierzy $\boldsymbol{A}$.

Dane wyjściowe:

- $\boldsymbol{x}$ – wektor zawierający rozwiązania układu $\boldsymbol{Ax} = \boldsymbol{b}$.

```
function eliminacja_gaussa(A, b, n, ℓ)
    for k ← 1 to n − 1 do
        e_non 0 ← min(ℓ + ℓ · ⌊(k+1)/ℓ⌋, n)
        k_last ← min(k + ℓ, n)
        for i ← k + 1 to e_non 0 do
            if A[k][k] = 0 then
                error współczynnik na przekątnej równy zeru
            z ← A[i][k]/A[k][k]
            A[i][k] ← 0
            for j ← k + 1 to k_last do
                A[i][j] ← A[i][j] − z · A[k][j]
            b[i] ← b[i] − z · b[k]
    for i ← n downto 1 do
        k_last ← min(i + ℓ, n)
        for j ← k + 1 to k_last do
            suma ← suma + x[i] · A[i][j]
        x[i] ← (b[i] − suma)/A[i][i]
    return x
```

Powyżej został rozpatrzony wariant metody eliminacji Gaussa bez wyboru elementu głównego, czasami jednak lepiej sprawdza się algorytm z tzw. częściowym wyborem (umożliwia rozwiązanie układu kiedy na diagonali macierzy pojawiają się elementy zerowe), w tym wypadku oznacza to wybranie wiersza, dla którego element w eliminowanej kolumnie i ma największą co do modułu wartość i zamienienie go z i-tym wierszem (po zamianie eliminacja jest kontynuowana w zwykły sposób).

W praktyce taka zamiana wierszy bywa kosztowna, szczególnie kiedy operacje wykonywane są na dużych macierzach, dlatego przy metodzie eliminacji Gaussa z wyborem elementu głównego pierwszą wprowadzoną zmianą jest stworzenie wektora permutacji wierszy (p), w którym pamiętane jest na jakiej aktualnie pozycji w macierzy znajduje się dany wiersz. Wpływ tego zabiegu na algorytm jest taki, że zamiast odwołania do konkretnego wiersza zostaje wykonane odwołanie do jego pozycji w wektorze permutacji.

Wybór elementu głównego sprawia również, że niemożliwe jest zachowanie wyliczonych wartości k_{last}, gdyż odejmowanie wierszy w innej kolejności, może doprowadzić do powstania nowych elementów niezerowych. Konieczne jest zatem nowe, szersze oszacowanie k_{last}. Zauważyć można, że w czasie eliminowania współczynników z $\ell - 2$ pierwszych kolumn najdalszy niezerowy element można stworzyć w kolumnie z indeksem 2ℓ – poprzez odejmowanie ℓ-tego wiersza, który w tej kolumnie posiada niezerowy element. Podczas eliminowania współczynników z kolejnych ℓ kolumn najdalszy niezerowy element można stworzyć w kolumnie z indeksem 3ℓ, analogicznie poprzez odejmowanie 2ℓ-tego wiersza, który w tej kolumnie posiada niezerowy element. Stosowanie powyższego rozumowania dla dalszych kolumn prowadzi do uzyskania nowego wzoru na k_{last}, mianowicie:

$$k_{last}(k) = \min\left\{2\ell + \ell \cdot \left\lfloor \frac{k+1}{\ell} \right\rfloor, n\right\}. \tag{7}$$

Podobne ograniczenie zastosowane jest również podczas wykonywania algorytmu podstawiania wstecz – nie powstają żadne nowe elementy niezerowe poza tymi już uwzględnionymi, jedyną zmianą jest uwzględnienie permutacji wiersza, co jednak w zasadzie nie wpływa na szacowaną wartość.

Metodę eliminacji Gaussa z częściowym wyborem elementu głównego przedstawia Algorytm 2

Złożoność obliczeniowa zmodyfikowanej metody eliminacji Gaussa z częściowym wyborem elementu głównego jest gorsza niż bez wyboru elementu głównego z powodu zastosowanych szerszych ograniczeń na k_{last}, jednak przy założeniu, że ℓ jest stałą nie wpływa to na ogólną złożoność $O(n)$.

Rozkład LU

Opis działania

Układy równań liniowych z niektórymi typami macierzy da się rozwiązać w sposób stosunkowo łatwy, takimi macierzami są np. macierze trójkątne – górna i dolna. Idea rozkładu $\boldsymbol{LU}$ macierzy $\boldsymbol{A}$ jest taka, żeby przedstawić ją za pomocą iloczynu

$$\boldsymbol{A} = \boldsymbol{LU}, \tag{8}$$

macierzy trójkątnej dolnej $\boldsymbol{L}$ z elementami na przekątnej równymi 1 i macierzy trójkątnej górnej $\boldsymbol{U}$, za pomocą których układ równań da się rozwiązać w stosunkowo łatwy sposób.

Taki rozkład można uzyskać za pomocą znanej już metody eliminacji Gaussa. Metoda ta przekształca macierz $\boldsymbol{A}$ do macierzy trójkątnej górnej, która stanie się macierzą $\boldsymbol{U}$. Macierz $\boldsymbol{L}$ zostaje stworzona poprzez zapamiętanie mnożników użytych do eliminacji kolejnych współczynników macierzy $\boldsymbol{A}$ i tak mnożnik użyty do wyzerowania elementu a_{ij} zapisujemy w i-tym wierszu i j-tej kolumnie macierzy $\boldsymbol{L}$. Cały rozkład $\boldsymbol{LU}$ można przeprowadzić bezpośrednio na macierzy $\boldsymbol{A}$ oszczędzając w ten sposób pamięć.

Złożoność obliczeniowa wyznaczenia rozkładu $\boldsymbol{LU}$ to $O(n^3)$, umożliwia on jednak stosunkowo szybkie rozwiązywanie wielu układów równań w których macierz jest taka sama, a zmienia się wektor prawych stron. W tym wypadku eliminacja Gaussa o dużej złożoności wykonywana jest tylko raz, a rozwiązywanie układów dzieli się na dwa etapy:

$$\begin{cases} \boldsymbol{Lz} = \boldsymbol{b} \\ \boldsymbol{Ux} = \boldsymbol{z}, \end{cases} \tag{9}$$

co dzięki postaci macierzy $\boldsymbol{L}$ i $\boldsymbol{U}$ (macierze trójkątne) można wykonać w $O(n^2)$ operacji.

Algorytm 2: Eliminacja Gaussa z częściowym wyborem elementu głównego

Dane wejściowe:

$\boldsymbol{A}$ – dana w zadaniu macierz postaci (2),
$\boldsymbol{b}$ – wektor prawych stron.
n – rozmiar macierzy $\boldsymbol{A}$,
ℓ – rozmiar bloku macierzy $\boldsymbol{A}$.

Dane wyjściowe:

$\boldsymbol{x}$ — wektor zawierający rozwiązania układu $\boldsymbol{Ax} = \boldsymbol{b}$.

function `eliminacja_gaussa_z_elementem_głównym(`$\boldsymbol{A}$, $\boldsymbol{b}$, n, ℓ`)`
  $\mathsf{p} \leftarrow \{\mathsf{i} : \mathsf{i} \in \{1, \ldots, n\}\}$
  for $\mathsf{k} \leftarrow 1$ **to** $n - 1$ **do**
    $e_{non\ 0} \leftarrow \min\left(\ell + \ell \cdot \left\lfloor \frac{\mathsf{k}+1}{\ell} \right\rfloor, n\right)$
    $k_{last} \leftarrow \min\left(2\ell + \ell \cdot \left\lfloor \frac{\mathsf{k}+1}{\ell} \right\rfloor, n\right)$
    for $\mathsf{i} \leftarrow \mathsf{k} + 1$ **to** $e_{non\ 0}$ **do**
      $r_{\max} \leftarrow \mathsf{m}$ takie, że: $\boldsymbol{A}[\mathsf{p}[\mathsf{m}]][\mathsf{k}] = \max(|\boldsymbol{A}[\mathsf{p}[\mathsf{q}]][\mathsf{k}]| : \mathsf{q} \in \{\mathsf{i}, \ldots, e_{non\ 0}\})$
      if $\mathsf{p}[r_{\max}] = 0$ **then**
        error macierz osobliwa
      $\mathtt{swap}\ (\mathsf{p}[\mathsf{k}], \mathsf{p}[r_{\max}])$
      $\mathsf{z} \leftarrow \boldsymbol{A}[\mathsf{p}[\mathsf{i}]][\mathsf{k}] / \boldsymbol{A}[\mathsf{p}[\mathsf{k}]][\mathsf{k}]$
      $\boldsymbol{A}[\mathsf{p}[\mathsf{i}]][\mathsf{k}] \leftarrow 0$
      for $\mathsf{j} \leftarrow \mathsf{k} + 1$ **to** k_{last} **do**
        $\boldsymbol{A}[\mathsf{p}[\mathsf{i}]][\mathsf{j}] \leftarrow \boldsymbol{A}[\mathsf{p}[\mathsf{i}]][\mathsf{j}] - \mathsf{z} \cdot \boldsymbol{A}[\mathsf{p}[\mathsf{k}]][\mathsf{j}]$
      $\boldsymbol{b}[\mathsf{p}[\mathsf{i}]] \leftarrow \boldsymbol{b}[\mathsf{p}[\mathsf{i}]] - \mathsf{z} \cdot \boldsymbol{b}[\mathsf{p}[\mathsf{k}]]$
  for $\mathsf{i} \leftarrow n$ **downto** 1 **do**
    $k_{last} \leftarrow \min\left(2\ell + \ell \cdot \left\lfloor \frac{\mathsf{p}[\mathsf{i}]+1}{\ell} \right\rfloor, n\right)$
    for $\mathsf{j} \leftarrow \mathsf{k} + 1$ **to** k_{last} **do**
      $\mathsf{suma} \leftarrow \mathsf{suma} + \boldsymbol{x}[\mathsf{j}] \cdot \boldsymbol{A}[\mathsf{p}[\mathsf{i}]][\mathsf{j}]$
    $\boldsymbol{x}[\mathsf{i}] \leftarrow (\boldsymbol{b}[\mathsf{p}[\mathsf{i}]] - \mathsf{suma}) / \boldsymbol{A}[\mathsf{p}[\mathsf{i}]][\mathsf{i}]$
  return $\boldsymbol{x}$

Zastosowane modyfikacje

Rozkład $\boldsymbol{LU}$ dla macierzy $\boldsymbol{A}$ w postaci 2 jest wyznaczany w sposób bardzo podobny do metody eliminacji Gaussa. Jednak zamiast zerowania elementów a_{ik} podstawiane są mnożniki $z = a_{ik}/a_{kk}$, które stanowią elementy macierzy $\boldsymbol{L}$.

Złożoność obliczeniowa wyznaczenia takiego rozkładu jest w oczywisty sposób taka sama, co złożoność dla metody eliminacji Gaussa bez wyboru elementu głównego (Algorytm 1) czy z częściowym jego wyborem (Algorytm 2), a więc, przy założeniu, że ℓ jest stałą, wynosi $O(n)$.

W celu rozwiązania układu równań liniowych $\boldsymbol{Ax} = \boldsymbol{b}$, gdzie $\boldsymbol{A} = \boldsymbol{LU}$ czyli $\boldsymbol{LUx} = \boldsymbol{b}$ należy podzielić obliczenia na dwa etapy 9. Drugi etap obliczeń czyli rozwiązanie układu $\boldsymbol{Ux} = \boldsymbol{z}$ nie różni się w zasadzie niczym od algorytmu podstawiania wstecz, nie ulega także zmianie wartość k_{last}, która wskazuje na to do której kolumny w danym wierszy należy sumować. Aby rozwiązać układ $\boldsymbol{Lz} = \boldsymbol{b}$ należy zastosować algorytm podstawiania w przód, który jest podobny do algorytmu podstawiania wstecz, jednak zaczyna się od pierwszego wiersza i sumowane są elementy z kolumn coraz dalszych, a nie coraz wcześniejszych. Algorytm obliczania $\boldsymbol{Lz} = \boldsymbol{b}$ został oczywiście odpowiednio zoptymalizowany ze względu na specyficzną postać macierzy. Warto zauważyć że elementy zerowe w macierzy $\boldsymbol{L}$ są na tych samych indeksach co te w macierzy

$\boldsymbol{A}$. Niepotrzebne jest więc rozpoczynanie sumowania od pierwszej kolumny dla każdego wiersza. W zasadzie takie sumowanie ma miejsce tylko dla ℓ pierwszych wierszy. Dla kolejnych ℓ wystarczy sumować od kolumny $\ell - 1$, a dla jeszcze dalszych ℓ kolumny $2\ell - 1$, itd. Tę zależność można przedstawić za pomocą następującego ogólnego wzoru:

$$k_{from}(r) = \max\left\{\ell \cdot \left\lfloor \frac{r-1}{\ell} \right\rfloor - 1, 1\right\}. \tag{10}$$

Metodę rozwiązania układu równań liniowych za pomocą rozkładu $\boldsymbol{LU}$ macierzy $\boldsymbol{A}$ prezentuje Algorytm 3.

Złożoność obliczeniowa rozwiązywania układu równań liniowych z rozkładu $\boldsymbol{LU}$ wynosi $O(n)$, ponieważ rozwiązanie $\boldsymbol{Ux} = \boldsymbol{z}$ ma taką samą złożoność jak w metodzie Gaussa, a rozwiązanie $\boldsymbol{Lz} = \boldsymbol{b}$ w oczywisty sposób ma także zbliżoną do tej złożoność.

Algorytm 3: Rozwiązywanie układu równań przy użyciu rozkładu $\boldsymbol{LU}$.

Dane wejściowe:

$\boldsymbol{A}$ – macierz (2) po przekształceniu do postaci, gdzie nad przekątną znajdują się elementy macierzy $\boldsymbol{U}$, a pod przekątną $\boldsymbol{L}$,
$\boldsymbol{b}$ – wektor prawych stron.
n – rozmiar macierzy $\boldsymbol{A}$,
ℓ – rozmiar bloku macierzy $\boldsymbol{A}$.

Dane wyjściowe:

$\boldsymbol{x}$ – wektor zawierający rozwiązania układu $\boldsymbol{Ax} = \boldsymbol{b}$.

function `rozwiązanie_LU`($\boldsymbol{A}$, $\boldsymbol{b}$, n, ℓ)
  for i $\leftarrow 1$ **to** n **do**
    suma $\leftarrow 0$
    $k_{from} \leftarrow \min\left(\ell \cdot \left\lfloor \frac{i-1}{\ell} \right\rfloor, n\right)$
    for j $\leftarrow k_{from}$ **to** i $- 1$ **do**
      suma $\leftarrow$ suma + z[j] $\cdot$ $\boldsymbol{A}$[i][j]
    z[i] = $\boldsymbol{b}$[i] $-$ suma
  for i $\leftarrow n$ **downto** 1 **do**
    suma $\leftarrow 0$
    $k_{last} \leftarrow \min(i + \ell, n)$
    for j $\leftarrow$ i + 1 **to** k_{last} **do**
      suma $\leftarrow$ suma + $\boldsymbol{x}$[j] $\cdot$ $\boldsymbol{A}$[i][j]
    $\boldsymbol{x}$[i] $\leftarrow$ (z[i] $-$ suma)/$\boldsymbol{A}$[i][i]
  return $\boldsymbol{x}$

Wyniki

Wnioski