WYKŁAD 7
PODSTAWOWE METODY NUMERYCZNE DLA LINIOWYCH UKŁADÓW ALGEBRAICZNYCH

Rozważmy układ n równań liniowych z niewiadomymi $x_1, x_2, ..., x_n$

$$\begin{cases} E_1: \; a_{1,1}x_1 + a_{1,2}x_2 + ... + a_{1,n-1}x_{n-1} + a_{1,n}x_n = b_1 \\ E_2: \; a_{2,1}x_1 + a_{2,2}x_2 + ... + a_{2,n-1}x_{n-1} + a_{2,n}x_n = b_2 \\ \vdots \\ E_{n-1}: a_{n-1,1}x_1 + a_{n-11,2}x_2 + ... + a_{n-1,n-1}x_{n-1} + a_{n-1,n}x_n = b_{n-1} \\ E_n: \; a_{n,1}x_1 + a_{n,2}x_2 + ... + a_{n,n-1}x_{n-1} + a_{n,n}x_n = b_n \end{cases}$$

W notacji macierzowo wektorowej powyższy układ równań ma postać

$$\mathbf{Ax} = \mathbf{b}$$

gdzie $\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & ... & a_{1,n-1} & a_{1,n} \\ a_{2,1} & a_{2,2} & ... & a_{2,n-1} & a_{2,n} \\ ... & ... & ... & ... & ... \\ a_{n-1,1} & a_{n-1,2} & ... & a_{n-1,n-1} & a_{n-1,n} \\ a_{n,1} & a_{n,2} & ... & a_{n,n-1} & a_{n,n} \end{bmatrix}$, $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$

Zakładamy, że rozwiązanie istnieje i jest jednoznaczne, tj. wyznacznik det(**A**) jest niezerowy.

Omówimy ogólną metodę rozwiązania takiego układu znaną pod nazwą Metody Eliminacji Gaussa (MEG). Podstawowa koncepcja tej metody jest prosta: manipulując w odpowiedni sposób równaniami sprowadzimy wyjściowy układ do równoważnego (tj. posiadającego to samo rozwiązanie) układu z **macierzą górną trójkątną**, a następnie w łatwy sposób rozwiążemy ten układ.

Rozważmy pierwszy krok procedury eliminacji. W kroku tym eliminujemy niewiadomą x_1 z równań E_2, E_3,.., E_n. W tym celu równanie E_1 jest mnożone przez odpowiednio dobrane liczby, a następnie jest odejmowane od kolejnych równań układu. Transformacja ta wygląda następująco:

$$\begin{cases} E_1: \ a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n = b_1 \\ E_2: \ a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,n}x_n = b_2 \\ \vdots \\ E_n: \ a_{n,1}x_1 + a_{n,2}x_2 + \ldots + a_{n,n}x_n = b_n \end{cases} \Rightarrow \begin{cases} E_1: \ a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n = b_1 \\ E_2^{(1)}: \quad a_{2,2}^{(1)}x_2 + \ldots + a_{2,n}^{(1)}x_n = b_2^{(1)} \\ \vdots \\ E_n^{(1)}: \quad a_{n,2}^{(1)}x_2 + \ldots + a_{n,n}^{(1)}x_n = b_n^{(1)} \end{cases}$$

gdzie

$$a_{i,j}^{(1)} = a_{i,j} - a_{1,j}l_{i,1} \quad , \quad l_{i,1} = \frac{a_{i,1}}{a_{1,1}}$$

$$b_i^{(1)} = b_i - b_1 l_{i,1} \quad , \quad i, j = 2, \ldots, n$$

Macierzowo-wektorowa postać układu liniowego otrzymanego w efekcie pierwszego etapu eliminacji to $\mathbf{A}^{(1)}\mathbf{x} = \mathbf{b}^{(1)}$, gdzie

$$\mathbf{A}^{(1)} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ 0 & a_{2,2}^{(1)} & \cdots & a_{2,n}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & a_{n,2}^{(1)} & \cdots & a_{n,n}^{(1)} \end{bmatrix} \qquad \mathbf{b}^{(1)} = \begin{bmatrix} b_1 \\ b_2^{(1)} \\ \vdots \\ b_n^{(1)} \end{bmatrix}$$

Procedura eliminacji kolejnych niewiadomych jest kontynuowana analogicznie aż do osiągnięcia docelowej formy układu. W szczególności, po (k-1)-szym kroku układ równań ma następującą formę

$$\begin{cases} E_1: & a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,k}x_k + \ldots + a_{1,n}x_n = b_1 \\ E_2^{(1)}: & \quad a_{2,2}^{(1)}x_2 + \ldots + a_{2,k}^{(1)}x_k + \ldots + a_{2,n}^{(1)}x_n = b_2^{(1)} \\ \vdots & \quad \ddots \quad \vdots \\ E_k^{(k-1)}: & \quad a_{k,k}^{(k-1)}x_k + \ldots + a_{k,n}^{(k-1)}x_n = b_k^{(k-1)} \\ E_{k+1}^{(k-1)}: & \quad a_{k+1,k}^{(k-1)}x_k + \ldots + a_{k+1,n}^{(k-1)}x_n = b_{k+1}^{(k-1)} \\ & \quad \vdots \\ E_n^{(k-1)}: & \quad a_{n,k}^{(k-1)}x_k + \ldots + a_{n,n}^{(k-1)}x_n = b_n^{(k-1)} \end{cases}$$

Górny indeks informuje ile razy modyfikowana była dana wielkość do tego momentu obliczeń. W następnym, tj. k-tym kroku, równanie $E_k^{(k-1)}$ będzie wykorzystane do eliminacji niewiadomej x_k z równań $E_{k+1}^{(k-1)}, \ldots, E_n^{(k-1)}$.

Formuły transformacji współczynników macierzy układu oraz elementów wektora prawych stron układu w kroku *k*-tym mają postać

$$a_{i,j}^{(k)} = a_{i,j}^{(k-1)} - a_{k,j}^{(k-1)} l_{i,k} \quad , \quad l_{i,k} = \frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}}$$

$$b_i^{(k)} = b_i^{(k)} - b_k^{(k-1)} l_{i,k} \quad , \quad i, j = k+1, \ldots, n$$

Macierzowo-wektorowa postać układu po *k* krokach eliminacji to $\mathbf{A}^{(k)}\mathbf{x} = \mathbf{b}^{(k)}$, gdzie

$$\mathbf{A}^{(k)} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,k+1} & \cdots & a_{1,n} \\ * & \ddots & \cdots & \cdots & \vdots \\ * & * & a_{k+1,k+1}^{(k)} & \cdots & a_{k+1,n}^{(k)} \\ * & * & \vdots & \ddots & \vdots \\ * & * & a_{n,k+1}^{(k)} & \vdots & a_{n,n}^{(k)} \end{bmatrix} \quad , \quad \mathbf{b}^{(k)} = [b_1, b_2^{(1)}, .., b_k^{(k-1)}, b_{k+1}^{(k)}, .., b_n^{(k)}]^T$$

Procedura eliminacji kończy się po n-1 krokach otrzymaniem układu równań o następującej strukturze

$$\begin{cases} E_1: & a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,k}x_k + \ldots + a_{1,n}x_n = b_1 \\ E_2^{(1)}: & \quad a_{2,2}^{(1)}x_2 + \ldots + a_{2,k}^{(1)}x_k + \ldots + a_{2,n}^{(1)}x_n = b_2^{(1)} \\ & \qquad \ddots \\ E_{n-1}^{(n-2)}: & \qquad\qquad a_{n-1,n-1}^{(n-2)}x_{n-1} + a_{n-1,n}^{(n-2)}x_n = b_{n-1}^{(n-2)} \\ E_n^{(n-1)}: & \qquad\qquad\qquad a_{n,n}^{(n-1)}x_n = b_n^{(n-1)} \end{cases}$$

W zapisie macierzowo-wektorowym mamy $\mathbf{Ux} = \hat{\mathbf{b}}$, gdzie

$$\mathbf{U} \equiv \mathbf{A}^{(n-1)} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n-1} & a_{1,n} \\ * & a_{2,2}^{(1)} & \cdots & a_{2,n-1}^{(1)} & a_{2,n}^{(1)} \\ * & * & \ddots & \ddots & \vdots \\ * & * & * & a_{n-1,n-1}^{(n-2)} & a_{n-1,n}^{(n-2)} \\ * & * & * & * & a_{n,n}^{(n-1)} \end{bmatrix}, \quad \hat{\mathbf{b}} \equiv \mathbf{b}^{(n-1)} = [b_1, b_2^{(1)}, .., b_{n-1}^{(n-2)}, b_n^{(n-1)}]^T$$

Otrzymany układ z **macierzą górną trójkątną U** można rozwiązać łatwo „od końca”. Istotnie, ostatnie równanie układu zawiera tylko ostatnią niewiadomą, którą można natychmiast obliczyć. Wówczas przedostatnie równanie jest efektywnie równaniem również z jedną niewiadomą (czyli x_{n-1}) ponieważ x_n jest już znana. Ogólnie, możemy zapisać następujący algorytm rozwiązania układu z macierzą **U** następująco

$$x_n = b_n^{(n-1)} / a_{n,n}^{(n-1)}$$

$$x_i = \frac{1}{a_{i,i}^{(i-1)}} \left[b_i^{(i-1)} - \sum_{j=i+1}^{n} a_{i,j}^{(i-1)} x_j \right] \quad , \quad i = n-1, n-2, \ldots, 2, 1$$

Zauważmy, że pętla realizowana jest w kierunku malejącego indeksu (wstecz). Pseudokod kompletnego algorytmu eliminacji Gaussa przedstawiamy na następnej stronie.

Metoda Eliminacji Gaussa (wariant podstawowy)

% *Etap 1 – Transformacja do układu trójkątnego*

for $k = 1 : n-1$

  for $j = k+1 : n$

    $l(j,k) \leftarrow a(j,k) / a(k,k)$

    $a(j, k+1:n) \leftarrow a(j, k+1:n) - l(j,k) \cdot a(k, k+1:n)$

    $b(j) \leftarrow b(j) - l(j,k) \cdot b(k)$

  end

end

$\rightarrow$

% *Etap 2 – Rozwiązanie*

$x(n) \leftarrow b(n) / a(n,n)$

for $j = n-1 : 1 : -1$

  $x(j) \leftarrow b(j)$

  for $k = j+1 : n$

    $x(j) \leftarrow x(j) - a(j,k) \cdot x(k)$

  end

  $x(j) \leftarrow x(j) / a(j,j)$

end

UWAGA: algorytm w opisanej formie może zawieść nawet jeśli został zastosowany do macierzy nieosobliwej ($\det(\mathbf{A}) \neq 0$). O tym dalej …

Faktoryzacja LU

Pokażemy, że ważnym „produktem ubocznym” eliminacji Gaussa **rozkład wyjściowej macierzy układu do postaci iloczynu dwóch macierzy: dolnej i górnej trójkątnej**. Rozkład taki nazywany jest faktoryzacją (przedstawieniem w postaci iloczynu) LU.

Rozważmy ponownie 1-szy krok eliminacji. Wprowadziliśmy uprzednio liczby $l_{2,1}^{(1)}, l_{3,1}^{(1)}, ..., l_{n,1}^{(1)}$. Użyjemy ich teraz do skonstruowania specjalnej macierzy $\tilde{\mathbf{L}}^{(1)}$, a mianowicie

$$\tilde{\mathbf{L}}^{(1)} = \begin{bmatrix} 1 & * & * & * \\ -l_{2,1}^{(1)} & 1 & * & * \\ \vdots & * & \ddots & * \\ -l_{n,1}^{(1)} & * & * & 1 \end{bmatrix}$$

Symbol „gwiazdka” oznacza elementy zerowe. Zauważmy teraz (**ćwiczenie dla Czytelnika!**), że pierwszy etap eliminacji transformuje oryginalny układ równań do układu z macierzą $\mathbf{A}^{(1)} = \tilde{\mathbf{L}}^{(1)}\mathbf{A}$!

Ogólnie, w k-tym etapie eliminacji zdefiniowaliśmy liczby $l_{j,k}^{(k)}$, $j = k+1,..,n$, za pomocą których możemy skonstruować macierz $\tilde{\mathbf{L}}^{(k)}$ postaci

$$\tilde{\mathbf{L}}^{(k)} = \begin{bmatrix} 1 & * & * & * & * & * \\ * & \ddots & * & * & * & * \\ * & * & 1 & * & * & * \\ * & * & -l_{k+1,k}^{(k)} & 1 & * & * \\ * & * & \vdots & * & \ddots & * \\ * & * & -l_{n,k}^{(k)} & * & * & 1 \end{bmatrix}$$

Ponownie, macierz współczynników układu po *k*-tym kroku eliminacji może być obliczona ze wzoru $\mathbf{A}^{(k)} = \tilde{\mathbf{L}}^{(k)}\mathbf{A}^{(k-1)}$.

Z powyższego wynika natychmiast, że docelowa macierz górna trójkątna U może być przedstawiona jako iloczyn oryginalnej macierzy A oraz macierzy typu $\tilde{\mathbf{L}}$ z kolejnych kroków eliminacji, a mianowicie

$$\mathbf{U} \equiv \mathbf{A}^{(n-1)} = \tilde{\mathbf{L}}^{(n-1)}\mathbf{A}^{(n-2)} = \underbrace{\tilde{\mathbf{L}}^{(n-1)}\tilde{\mathbf{L}}^{(n-2)}...\tilde{\mathbf{L}}^{(2)}\tilde{\mathbf{L}}^{(1)}}_{\tilde{\mathbf{L}}}\mathbf{A} = \tilde{\mathbf{L}}\mathbf{A}$$

Kapitalną własnością macierzy $\tilde{\mathbf{L}}$ jest to, że macierz do niej odwrotna wyraża się zaskakująco prosto, a mianowicie

$$\mathbf{L} \equiv \tilde{\mathbf{L}}^{-1} = \begin{bmatrix} 1 & * & * & * & * & * \\ l_{2,1}^{(1)} & 1 & * & * & * & * \\ l_{3,1}^{(1)} & l_{3,2}^{(2)} & \ddots & * & * & * \\ \vdots & \vdots & \ddots & 1 & * & * \\ \vdots & \vdots & \ddots & l_{n-1,n-2}^{(n-2)} & 1 & * \\ l_{n,1}^{(1)} & l_{n,2}^{(2)} & \cdots & l_{n,n-2}^{(n-2)} & l_{n,n-1}^{(n-1)} & 1 \end{bmatrix}$$

Dowód tego (nieoczywistego) faktu można znaleźć w dobrych podręcznikach numerycznej algebry liniowej np. w znakomitej książce Trefethena i Bau pt. “Numerical Linear Algebra”. Dla ambitnych godna polecenia jest również monografia „Analiza Numeryczna” Kincaida i Cheneya (WNT, Warszawa 2006).

Otrzymaliśmy zatem następującą formułę faktoryzacyjną dla macierzy (nieosobliwej!) **A**

$$\mathbf{A} = \mathbf{LU}$$

Pseudokod podstawowej wersji faktoryzacji LU wygląda następująco:

Faktoryzacja **LU** *– wariant podstawowy*

for $k = 1 : n-1$ *do*

  for $j = k+1 : n$ *do*

    $a(j,k) \leftarrow a(j,k) / a(k,k)$

    $a(j,k+1:n) \leftarrow a(j,k+1:n) - a(j,k) \cdot a(k,k+1:n)$

  end

end

Uwaga **:**

1. *Oryginalna zawartosc* **A** *zostala zniszczona*!
2. *Glowna diagonala i gorny trojkat* **A** *zawiera niezerowe elementy macierzy* **U**
3. *Dolny trojkat* **A** *zawiera niezerowe elementy macierzy* **L**. *Nie ma potrzeby przechowywania elementow diagonalnych L* (*są to jedynki*).

Co można powiedzieć o koszcie numerycznym MEG i faktoryzacji LU? Analizując pseudokody tych algorytmów można wyciągnąć następujące wnioski:

1. W etapie eliminacji mamy trzy zagnieżdżone pętle. Wynika z tego, że **całkowita liczba operacji zmiennoprzecinkowych jest proporcjonalna do n^3**, gdzie n jest rozmiarem układu. Dokładniejsza analiza pokazuje, że dla dużych wartości n dominujący koszt wynika z operacji mnożenia i dodawania/odejmowania oraz, że liczba tych operacji jest bliska $\frac{2}{3}n^3$.

2. W etapie rozwiązania układu z macierzą górną trójkątną mamy jedynie dwie zagnieżdżone pętle, wobec czego koszt numeryczny będzie (z grubsza) proporcjonalny n^2. Wnioskujemy, że **dla układu o dużym rozmiarze koszt numeryczny etapu rozwiązania jest znikomy w porównaniu z kosztem sprowadzenia tego układu do postaci trójkątnej**.

3. **Koszt numeryczny faktoryzacji LU (zastosowanej do ogólnej macierzy kwadratowej macierzy) jest proporcjonalny do trzeciej potęgi jest rozmiaru.**

Kiedy opłaca się zastosować faktoryzację LU zamiast „zwyczajnej” MEG? Rozważmy sytuację, w której chcemy wyznaczyć rozwiązania szeregu m układów liniowych o tej samej macierzy **A**, ale różnych wektorach prawych stron

$$\mathbf{A}\mathbf{x}^{(j)} = \mathbf{b}^{(j)} \quad , \quad j = 1,..,m$$

Używając MEG, całkowity koszt numeryczny rozwiązania tych układów będzie proporcjonalny (z grubsza) do $m \cdot n^3$. Taka estymacja kosztu wynika z faktu, że każdorazowo przeprowadzana jest kompletna procedura eliminacji (kosztem $\sim n^3$) **pracującą na tej samej macierzy**. Niepotrzebnego powtarzania tych samych operacji można jednak uniknąć stosując faktoryzację LU. Idea polega na jednorazowym obliczeniu faktorów **U** i **L** (kosztem $\sim n^3$), a następnie rozwiązaniu m układów równań kosztem proporcjonalnym jedynie do n^2 (oba faktory są trójkątne). Oto jak to działa:

1) *Wyznacz faktory* **L** *i* **U** *macierzy* **A** ($koszt \sim n^3$)

2) *dla* $j = 1,..,m$ *wykonaj*

$$\mathbf{L}\mathbf{U}\mathbf{x}^{(j)} = \mathbf{b}^{(j)} \Rightarrow \begin{cases} \mathbf{L}\mathbf{y} = \mathbf{b}^{(j)} \\ \mathbf{U}\mathbf{x}^{(j)} = \mathbf{y} \end{cases}$$

Pseudokod procedury rozwiązania układu liniowego przy pomocy faktorów L i U

Rozwiazanie ukladu (LU)x = b

% Ly = b

```
y(1) ← b(1)
for j = 2 : n
    y(j) ← b(j)
    for k = 1 : j − 1
        y(j) ← y(j) − a(j,k) · y(k)
    end
end
```

⟹

% Ux = y

```
x(n) ← y(n) / a(n,n)
for j = n − 1 : 1 : −1
    x(j) ← y(j)
    for k = j + 1 : n
        x(j) ← x(j) − a(j,k) · x(k)
    end
    x(j) ← x(j) / a(j, j)
end
```

Trudności z podstawowym wariantem MEG. Wybór elementu głównego.

Przedstawiony wyżej podstawowy wariant MEG może zawieść nawet gdy macierz **A** jest nieosobliwa. Istotnie, jeśli w k-1-szym etapie eliminacji z równania $E_k^{(k-1)}$ zniknęła również niewiadoma x_k , to współczynnik $a_{k,k}^{(k-1)}$ jest równy zeru i w kroku k-tym nie mam możliwości obliczenia żadnej z liczb $l^{(k)}$ (mamy dzielenia przez zero)!

W skrajnym przypadku, pierwsze z równań może w ogóle nie zawierać niewiadomej x_1 i proces eliminacji nie może być rozpoczęty! Oto banalny przykład nieodpowiedniej macierzy

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

To jednak nie koniec problemów. Rozważmy mianowicie układ równań z następującą macierzą współczynników

$$\mathbf{A} = \begin{bmatrix} 10^{-10} & 1 \\ 1 & 1 \end{bmatrix}$$

Jej faktory **L** i **U** to (sprawdzić – ćwiczenie!): $\mathbf{L} = \begin{bmatrix} 1 & 0 \\ 10^{10} & 1 \end{bmatrix}$, $\mathbf{U} = \begin{bmatrix} 10^{-10} & 1 \\ 0 & 1-10^{10} \end{bmatrix}$

Jak wiemy, arytmetyka komputera nie jest dokładna. Zobaczymy co się stanie jeśli do rozwiązania układu liniowego z powyższą macierzą zastosujemy komputer, którego arytmetyka jest dziesiętna i ma dokładność 7-miu miejsc znaczących (w mantysie). W uwagi na prostotę układu obliczenia takiego hipotetycznego komputera możemy z łatwością zasymulować … ręcznie!

Zacznijmy od tego, że element u_{22} faktora **U** będzie miał reprezentację przybliżoną, a mianowicie

$$10^{10} - 1 = 1 \cdot 10^{10} - 0.0000000001 \cdot 10^{10} = (1 - \underbrace{0.0000000}_{\text{tylko 7 cyfr jest uwzględnionych}} 001) \cdot 10^{10} \cong 10^{10}$$

Zatem, dokładny faktor U będzie w naszym komputerze zastąpiony faktorem przybliżonym postaci

$$\tilde{\mathbf{U}} = \begin{bmatrix} 10^{-10} & 1 \\ 0 & -10^{10} \end{bmatrix}$$

Zauważmy, że faktor **L** będzie natomiast reprezentowany ściśle ($\tilde{\mathbf{L}} \equiv \mathbf{L}$).

Jeśli pomnożymy faktory macierzy **A** tak, jak widzi je nasz komputer to otrzymamy

$$\tilde{\mathbf{A}} = \mathbf{L}\tilde{\mathbf{U}} = \begin{bmatrix} 10^{-10} & 1 \\ 1 & 0 \end{bmatrix} \neq \mathbf{A}$$

Jak widać, różnica pomiędzy **A** and $\tilde{\mathbf{A}}$ jest niebagatelna!

Zobaczmy jaki wpływ na rozwiązanie układu liniowego mają błędy zaokrągleń w naszym przykładzie. Rozważmy układ równań postaci

$$\mathbf{Ax} = \mathbf{b}\ , \quad \mathbf{b} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Jego dokładne rozwiązanie to

$$\mathbf{x} = \begin{bmatrix} -(1-10^{-10})^{-1} \\ (1-10^{-10})^{-1} \end{bmatrix} \approx \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Zastosujemy metodę faktoryzacji LU uwzględniając fakt przybliżonej reprezentacji faktora U i błędy zaokrągleń popełnione w trakcie obliczeń. Mamy wówczas

$$\underset{\mathbf{y}}{\mathbf{L}\,\tilde{\mathbf{U}}\mathbf{x}} = \begin{bmatrix}1\\0\end{bmatrix} \Rightarrow \begin{bmatrix}1 & 0\\ 10^{10} & 1\end{bmatrix}\begin{bmatrix}y_1\\y_2\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix} \Rightarrow \mathbf{y} = \begin{bmatrix}1\\-10^{10}\end{bmatrix}$$

$$\begin{bmatrix}10^{-10} & 1\\ 0 & -10^{10}\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} = \begin{bmatrix}1\\-10^{10}\end{bmatrix} \Rightarrow \tilde{\mathbf{x}} = \begin{bmatrix}0\\1\end{bmatrix} \overset{!!!}{\neq} \mathbf{x}$$

Ponownie, otrzymane rozwiązanie znacząco odbiega od rozwiązania dokładnego! Widać również, że „winnym" zaobserwowanego efektu jest wielki co do wartości mnożnik $l_{2,1}^{(1)}$, równy aż 10^{10}!

Okazuje się, że prostym lekarstwem na opisany problem jest **zmiana kolejności równań**, czyli następująca transformacja

$$\mathbf{A} = \begin{bmatrix}10^{-10} & 1\\ 1 & 1\end{bmatrix},\ \mathbf{b} = \begin{bmatrix}1\\0\end{bmatrix} \quad \underset{\substack{\textit{zmiana}\\ \textit{kolejnosci}\\ \textit{rownan}}}{\Longrightarrow} \quad \hat{\mathbf{A}} = \begin{bmatrix}1 & 1\\ 10^{-10} & 1\end{bmatrix},\ \hat{\mathbf{b}} = \begin{bmatrix}0\\1\end{bmatrix}$$

Poniższy rachunek pokazuje, że tym razem metoda faktoryzacji LU działa znakomicie!

$$\hat{\mathbf{L}} = \begin{bmatrix} 1 & 0 \\ 10^{-10} & 1 \end{bmatrix} \quad , \quad \hat{\mathbf{U}} = \begin{bmatrix} 1 & 1 \\ 0 & 1-10^{-10} \end{bmatrix} \cong \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \tilde{\hat{\mathbf{U}}}$$

$$\hat{\mathbf{L}}\tilde{\hat{\mathbf{U}}}\mathbf{x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 0 \\ 10^{-10} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow \hat{\mathbf{y}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow \tilde{\hat{\mathbf{x}}} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \approx \mathbf{x}$$

Zauważmy, że tym razem mnożnik $l_{2,1}^{(1)} = 10^{-10}$, tj. jest bardzo mały i nie powoduje „katastrofalnego błędu obcięcia”. Ogólnie mówiąc, należy unikać dużych wartości mnożników *l*!

Ogólna strategia stosowana w celu otrzymania algorytmu eliminacji odpornego na opisane wyżej efekty nosi nazwę **Metody Eliminacji Gaussa z wyborem elementu głównego**.

Wybór elementu głównego w *k*-tym kroku eliminacji polega na:

- znalezieniu takiego równania $E_{EG}^{(k-1)}$ wśród równań $E_j^{(k-1)}$, $j \in \{k, k+1, .., n\}$, w którym współczynnik przy niewiadomej x_k jest największy co do modułu,
- przestawieniu w układzie równania $E_{EG}^{(k-1)}$ i równania $E_k^{(k-1)}$.

W ten sposób **wartości bezwzględne wszystkich mnożników *l* nigdy nie przekraczają jedności i** katastrofalna utrata dokładności nie ma miejsca (chyba, że macierz **A** jest fatalnie uwarunkowana, o czym przy innej okazji …).

Pseudokod Metody Eliminacji Gaussa z wyborem elementu głównego

for $k = 1 : n-1$

Znajdz $i \geq k$ *taki, ze* $|a_{i,k}|$ *jest maksymalny*

$a(k, k:n) \leftrightarrow a(i, k:n)$ *przestaw wiersze* $i-ty$ *z* $k-tym$

$b(k) \leftrightarrow b(i)$ *przestaw elementy wektora* **b**

for $j = k+1 : n$

$l(j,k) \leftarrow a(j,k) / a(k,k)$

$a(j, k+1:n) \leftarrow a(j, k+1:n) - l(j,k) \cdot a(k, k+1:n)$

$b(j) \leftarrow b(j) - l(j,k) \cdot b(k)$

end

end

$\Rightarrow$

$x(n) \leftarrow b(n) / a(n,n)$

for $j = n-1 : 1 : -1$

$x(j) \leftarrow b(j)$

for $k = j+1 : n$

$x(j) \leftarrow x(j) - a(j,k) \cdot x(k)$

end

$x(j) \leftarrow x(j) / a(j,j)$

end

Wybór elementu głównego a faktoryzacja LU

Zrozumienie w jaki sposób wybór elementu głównego wpływa na postać faktoryzacji **LU** wymaga wprowadzenia tzw. **macierzy permutującej**. Ogólnie, macierz permutująca otrzymywana jest w wyniku przestawienia wierszy (lub - równoważnie – kolumn) macierzy jednostkowej **I**. Na przykład:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \mathbf{P} := \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Co się stanie jeśli dana macierz **A** zostanie pomnożona przez macierz permutującą? Oto odpowiedź:

$$\mathbf{PA} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} 7 & 8 & 9 \\ 4 & 5 & 6 \\ 1 & 2 & 4 \end{bmatrix}$$ – *przestawienie wierszy 1 i 3 w macierzy* **A**

$$\mathbf{AP} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 3 & 2 & 1 \\ 6 & 5 & 4 \\ 9 & 8 & 7 \end{bmatrix}$$ – *przestawienie kolumn 1 i 3 w macierzy* **A**

Rozważmy teraz proces eliminacji z wyborem EG. Ponieważ wynikające z wyboru EG przestawienie równań układu poprzedza eliminacje kolejnej niewiadomej, zatem transformacja macierzy $\mathbf{A}^{(k-1)}$ do macierzy $\mathbf{A}^{(k)}$ może być zapisana w postaci następującej

$$\mathbf{A}^{(0)} \equiv \mathbf{A} \quad , \quad \mathbf{A}^{(k)} = \tilde{\mathbf{L}}^{(k)}\mathbf{P}^{(k)}\mathbf{A}^{(k-1)} \quad , \quad k = 1,2,..,n-1$$

W konsekwencji

$$\mathbf{U} \equiv \mathbf{A}^{(n-1)} = \tilde{\mathbf{L}}^{(n-1)}\mathbf{P}^{(n-2)}\mathbf{A}^{(n-2)} = \tilde{\mathbf{L}}^{(n-1)}\mathbf{P}^{(n-1)}\tilde{\mathbf{L}}^{(n-2)}\mathbf{P}^{(n-2)}...\tilde{\mathbf{L}}^{(1)}\mathbf{P}^{(1)}\mathbf{A}$$

Fenomenalna własność macierzy permutujących **P** i faktorów **L** polega na tym, że ich naprzemienny iloczyn da się zapisać następująco

$$\tilde{\mathbf{L}}^{(n-1)}\mathbf{P}^{(n-1)}\tilde{\mathbf{L}}^{(n-2)}\mathbf{P}^{(n-2)}...\tilde{\mathbf{L}}^{(1)}\mathbf{P}^{(1)} = \left[\tilde{\tilde{\mathbf{L}}}^{(n-1)}\tilde{\tilde{\mathbf{L}}}^{(n-2)}...\tilde{\tilde{\mathbf{L}}}^{(1)}\right]\cdot\left[\mathbf{P}^{(n-1)}\mathbf{P}^{(n-2)}...\mathbf{P}^{(1)}\right]$$

gdzie oznaczyliśmy

$$\tilde{\tilde{\mathbf{L}}}^{(k)} = \mathbf{P}^{(n-1)}...\mathbf{P}^{(k+1)}\tilde{\mathbf{L}}^{(k)}[\mathbf{P}^{(k+1)}]^{-1}...[\mathbf{P}^{(n-1)}]^{-1}$$

Co więcej, macierz odwrotna $\mathbf{L} = \left[\tilde{\tilde{\mathbf{L}}}^{(n-1)}\tilde{\tilde{\mathbf{L}}}^{(n-2)}...\tilde{\tilde{\mathbf{L}}}^{(1)}\right]^{-1}$ istnieje i jest dolna trójkątna!

Wprowadzając zatem globalną macierz permutującą $\mathbf{P} = \mathbf{P}^{(n-1)}\mathbf{P}^{(n-2)}...\mathbf{P}^{(1)}$ otrzymujemy następującą postać faktoryzacji LU

$$\mathbf{PA} = \mathbf{LU}$$

Faktoryzacja LU z wyborem EG może być wykorzystana do rozwiązania liniowego układu równań z następujący sposób

$$\mathbf{Ax} = \mathbf{b} \Rightarrow \mathbf{PAx} = \mathbf{Pb} \Rightarrow \mathbf{LUx} = \mathbf{Pb} \Rightarrow \begin{cases} \mathbf{Ly} = \mathbf{Pb} \\ \mathbf{Ux} = \mathbf{y} \end{cases}$$

Jak widać, jedyna komplikacja polega na tym, że w programie komputerowym należy zapamiętać historię przestawień kolejności równań dokonanych w procesie eliminacji. W praktycznej implementacji załatwia się do za pomocą dodatkowego wektora zawierającego odpowiednio poprzestawiane liczby naturalne od 1 do n = dim(**A**).

Inne użyteczne zastosowania faktoryzacji LU:

1. Obliczanie wyznacznika macierzy **A**

$$\mathbf{PA} = \mathbf{LU} \quad \Rightarrow \quad \det(\mathbf{PA}) = \det(\mathbf{LU})$$

$$\Downarrow$$

$$\underbrace{\det(\mathbf{P})}_{\substack{1\,(\textit{even permut.})\\ \textit{or } -1\,(\textit{odd perm.})}} \det(\mathbf{A}) = \underbrace{\det(\mathbf{L})}_{\substack{=1\,(\textit{product of}\\ \textit{diagonal elem.})}} \det(\mathbf{U})$$

$$\Downarrow$$

$$\det(\mathbf{A}) = \pm\det(\mathbf{U}) = u_{11}u_{22}...u_{nn}$$

2. Obliczanie macierzy odwrotnej (na ogół należy go unikać!)

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I} \quad \Rightarrow \quad \mathbf{A}\mathbf{x}_i = \mathbf{e}_i \quad , \quad \mathbf{e}_i \leftarrow i-ta\ kolumna\ macierzy\ \mathbf{I} \quad , \quad i = 1,2,..,n$$

Ostatecznie $\mathbf{A}^{-1} = [\mathbf{x}_1 \,|\, \mathbf{x}_2 \,|\, ... \,|\, \mathbf{x}_n]$

Uwaga: Powyższe układy mogą być rozwiązane łącznym kosztem proporcjonalnym do n^3. Naiwne zastosowanie eliminacji Gaussa dałoby koszt całkowity proporcjonalny do n^4.

UWAGI KOŃCOWE

- Istnieją inne użyteczne warianty faktoryzacji macierzy. W szczególności każda macierz może być przedstawiona w formie iloczynu macierzy ortogonalnej Q i górnej trójkątnej R (faktoryzacja QR). Dla macierzy symetrycznych i dodatnio określonych ($\mathbf{A} = \mathbf{A}^T$, $\mathbf{x}^T \mathbf{A}\mathbf{x} > 0$ *dla każdego* $\mathbf{x} \neq \mathbf{0}$) istnieje faktoryzacja Choleskiego ($\mathbf{A} = \mathbf{L}\mathbf{L}^T$). Metody te omawiane są m.in. w trakcie kursu „Metody Numeryczne" prowadzonym na WMEiL.

- Metoda eliminacji Gaussa i faktoryzacja LU należą do tzw. „metod dokładnych", tj. algorytmów, które dają dokładne (przynajmniej w teorii) rozwiązanie układu po skończonej, z góry określonej liczbie operacji arytmetycznych. Dla układów o bardzo dużych rozmiarach koszt numeryczny i wymagania co do pamięci komputera są niepraktyczne i/lub nierealistyczne. Wielkie układy równań (o wymiarach rzędu $n \sim 10^4$ i więcej) rozwiązuje się **metodami przybliżonymi, tj. metodami w których rozwiązanie osiągane jest na drodze kolejnych iteracji**. Niektóre z tych metod omawiane są w kursie „Metody Numeryczne", a także w innych prowadzonych na Wydziale MEiL kursach poświęconych zastosowaniom metod komputerowych w rozmaitych działach techniki.