

2 Rozdział

Strona: [LeIA](#)

Kurs: Metody numeryczne (2025Z)

Książka: 2 Rozdział

Wydrukowane przez użytkownika: Kinga Kondraciuk

Data: niedziela, 30 listopada 2025, 14:01

Spis treści

- 1. Normy wektorów**
- 2. Numeryczna algebra liniowa**
- 3. Normy macierzowe**
- 4. Współczynnik uwarunkowania**
- 5. Metoda eliminacji Gaussa**
- 6. Wsteczne podstawienie**
- 7. Eliminacja Gaussa-Jordana**
- 8. Rozkład na czynniki - faktoryzacja macierzy**
 - 8.1. Metoda Doolittle'a
 - 8.2. Metoda eliminacji Gaussa

1. Normy wektorów

W algebrze liniowej, analizie funkcyjnej i pokrewnych dziedzinach matematyki, norma to funkcja przyporządkowująca dodatnią wartość liczbową określającą długość lub wielkość wektora, lub przestrzeni wektorowej gdy obliczamy normę macierzy. W ogólności normy są wykorzystywane do określania odległości między punktami wskazywanymi przez wektory oraz porównywania ich długości. Istnieje wiele różnych rodzajów norm (różnych funkcji), które mogą spełnić to zadanie. Aby funkcja przekształcająca wektor lub macierz w liczbę mogła być nazywana normą, to musi ona posiadać następujące właściwości:

1. $\|av\| = |a|\|v\|$ (skalarność)
2. $\|u + v\| \leq \|u\| + \|v\|$ (nierówność trójkątna)
3. $\|v\| \geq 0$ (niewjemność)
4. Jeżeli $\|v\| = 0$ to $v = 0$, czyli v jest wektorem zerowym (jednoznaczność),

gdzie a to dowolna wartość skalarna, i to dowolne wektory. W dalszej części przedstawimy kilka najczęściej wykorzystywanych funkcji, które charakteryzują się przed chwilą wymienionymi właściwościami, zatem są normami macierzy:

Norma euklidesowa (L2-norm)

To najbardziej intuicyjna norma wśród norm wektorowych, która stanowi euklidesową długość wektora, czyli pierwiastek sumy kwadratów jego składowych x_i

$$\|x\|_2 := \sqrt{x_1^2 + \dots + x_n^2} \quad (2.1.4)$$

p-norma

p -norma to funkcja zwracająca wartość skalarną zdefiniowaną jako pierwiastek p -tego stopnia sumy składowych wektora podniesionych do p -tej potęgi. W tym kontekście, zdefiniowana wcześniej norma euklidesowa jest szczególnym przypadkiem p -normy dla $p = 2$.

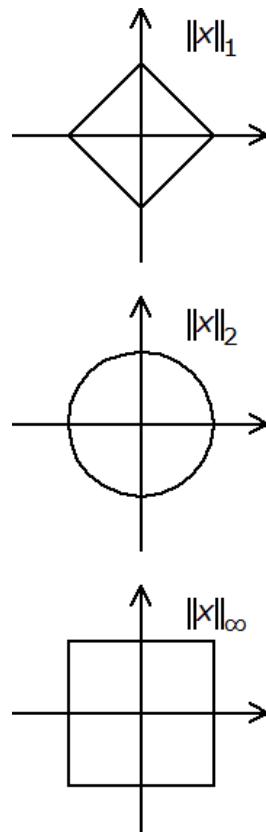
$$\|x\|_p := \left(\sum_{i=0}^n |x_i|^p \right)^{1/p} \quad (2.1.5)$$

Norma nieskończoność

Przykład klasycznej normy, w której funkcja przyporządkowuje wektorowi wartość maksymalnej składowej x_i . Zaletą tej normy jest jest bardzo prosta i szybka implementacja oraz brak wrażliwości na błędy operacji arytmetycznych.

$$\|x\|_\infty := \max_i |x_i|. \quad (2.1.6)$$

Na rysunku 2.2 przedstawiona została interpretacja graficzna wymienionych norm indukowanych w przestrzeni wektorów dwuwymiarowych. Przedstawiono na nim „koła jednostkowe”, stanowiące zbiory punktów znajdujących się na końcach wszystkich wektorów, które dla poszczególnych zdefiniowanych norm uzyskują wartość równą jeden. Wektory są zaczepione w początku układu współrzędnych.



Rys. 2.2. "Koła jednostkowe" dla trzech definicji norm: L_1 dla p -normy i $p = 1$, L_2 dla normy euklidesowej oraz L_∞ dla normy nieskończoność.

[Rysunek dla normy L_1 przedstawia obrócony o 45° kwadrat, którego wierzchołki są na osiach współrzędnych ox i oy , a jego środek ciężkości w początku układu współrzędnych. Środkowy rysunek dla normy L_2 przedstawia okrąg ze środkiem w początku układu współrzędnych. Trzeci, dolny rysunek dla normy L_∞ przedstawia kwadrat ze środkiem ciężkości w początku układu współrzędnych i bokami równoległymi do osi ox i oy .]

2. Numeryczna algebra liniowa

Algebra liniowa jest często osobnym kursem ujętym w programie typowych studiów inżynierskich. Niemniej istnieją pewne specjalne techniki, nieuwzględnione w standardowym kursie, które są związane ze specyfiką rozwiązywania zagadnień algebry liniowej na komputerach. Niniejszy rozdział przedstawia kilka wybranych takich metod.

Typowym zagadnieniem z zakresu algebry liniowej wykorzystywanym w ramach problemów inżynierskich jest rozwiązanie układu równań liniowych. Mamy tutaj na myśli nie układy z trzema czy czterema niewiadomymi, które standardowo są rozwiązywane analitycznie, lecz układy z setkami czy nawet dziesiątkami tysięcy niewiadomych. Tego typu problemy wymagają specjalnych technik, które nie tylko gwarantują znalezienie rozwiązania, ale również znajdują je w sposób minimalizujący nakłady oraz błędy obliczeniowe. Wśród metod rozwiązywania układów równań liniowych wyróżniamy metody bezpośrednie oraz metody iteracyjne. Do metod bezpośrednich zaliczamy takie jak: eliminacja Gaussa, faktoryzacje LU czy QR, faktoryzacje SVD. Do metod iteracyjnych zaliczamy Jacobiego, SOR, Gradientów sprężonych, GMRES, i inne. W niniejszym kursie skupimy się tylko na wybranych metodach bezpośrednich.

Drugim klasycznym zagadnieniem algebry liniowej, który ma duże znaczenie z punktu widzenia inżynierskiego jest wyznaczanie wartości własnych macierzy. Należy zaznaczyć, że wyznaczanie wartości własnych macierzy metodami analitycznymi jest niezmiernie czasochłonne i często sprowadza się do rozwiązywania bardzo źle uwarunkowanego wielomianowego równania charakterystycznego. Wśród metod numerycznych pozwalających wyznaczyć wartości i wektory własne są metody pozwalające wyznaczyć wartości własne rzeczywiste i ujęte, maksymalną wartość własną, minimalną wartość własną lub wszystkie wartości. W niniejszym podręczniku przedstawimy jedynie podstawowe metody wyznaczania wartości własnych minimalnej i maksymalnej. Pozostałe metody wykraczają poza program studiów inżynierskich.

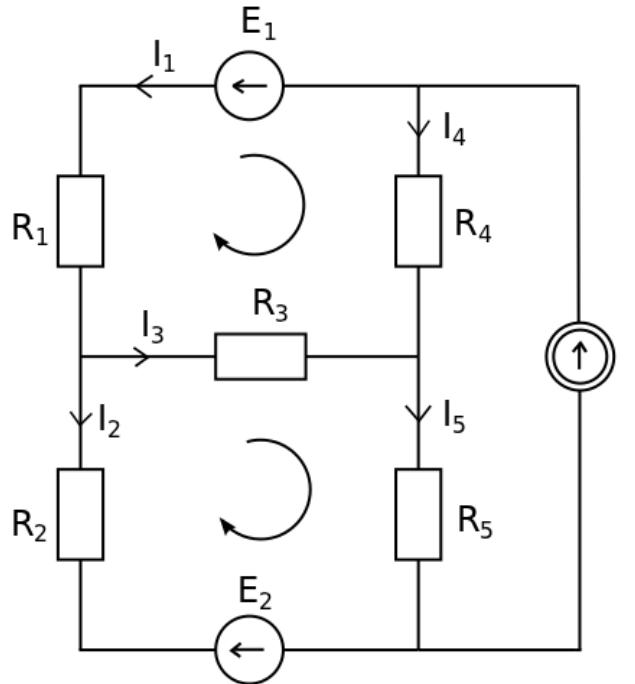
Na początku wprowadźmy podstawowe oznaczenia. Wiele zagadnień naukowych oraz inżynierskich prowadzi do układu równań liniowych $Ax = b$, który w formie macierzowej przyjmuje postać:

$$Ax = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (2.1.1)$$

Przykład 2.1

Przykład

Rozważmy przykład techniczny obwodu elektrycznego zaprezentowanego na rysunku 2.1. Naszym zadaniem jest przedstawić w zapisie macierzowym układ równań pozwalający znaleźć rozkład prądów w obwodzie. Należy wykorzystać prawa Kirchhoffa. Parametry obwodu to: $I = 10[A]$, $E1 = 5[V]$, $E2 = 8[V]$, $R1 = 5[\Omega]$, $R2 = 5[\Omega]$, $R3 = 3[\Omega]$, $R4 = 7[\Omega]$, $R5 = 2[\Omega]$



Rysunek 2.1. Obwód elektryczny zbudowany z trzech oczek, zawierający pięć rezystancji, dwa źródła napięcia E_1 , E_2 oraz jedno źródło prądu I .

Z równań Kirchhoffa otrzymujemy pięć równań. Pierwsze trzy równania przedstawiają bilans prądów w węzłach, a pozostałe dwa bilans napięć w oczkach. Układ pięciu niezależnych liniowo równań zawiera pięć niewiadomych i posiada jednoznaczne rozwiązańe.

$$\begin{aligned}
 I_1 + I_4 &= I \\
 I_3 + I_4 &= I_5 \\
 I_2 + I_3 &= I_1 \\
 R_1 I_1 + R_3 I_3 - R_4 I_4 &= E_1 \\
 R_2 I_2 - R_3 I_3 - R_5 I_5 &= -E_2
 \end{aligned} \tag{2.1.2}$$

Powyższy układ równań zapisany algebraicznie możemy przekształcić do postaci macierzowej:

$$\left[\begin{array}{ccccc} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & -1 \\ -1 & 1 & 1 & 0 & 0 \\ R_1 & 0 & R_3 & -R_4 & 0 \\ 0 & R_2 & -R_3 & 0 & -R_5 \end{array} \right] \left[\begin{array}{c} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{array} \right] = \left[\begin{array}{c} I \\ 0 \\ 0 \\ E_1 \\ -E_2 \end{array} \right] \tag{2.1.3}$$

Ostatecznie po podstawieniu wartości liczbowych otrzymujemy układ równań przedstawiony w równaniu 2.1.3.

$$\left[\begin{array}{ccccc} 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & -1 \\ -1 & 1 & 1 & 0 & 0 \\ 5 & 0 & 3 & -7 & 0 \\ 0 & 5 & -3 & 0 & -2 \end{array} \right] \left[\begin{array}{c} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \end{array} \right] = \left[\begin{array}{c} 10 \\ 0 \\ 0 \\ 5 \\ -8 \end{array} \right] \tag{2.1.3}$$

3. Normy macierzowe

Macierze często są nazywane operatorami liniowymi, które przekształcają wektory, np. za pomocą operacji $x' = Ax$. Przekształceniem może być wydłużenie lub skrócenie wektora. W tym kontekście, normy macierzowe są funkcjami, które przyporządkowują danej macierzy liczbę skalarną wyrażającą zdolność macierzy do wydłużania wektorów. Każda norma wektorowa pozwala nam zdefiniować normę macierzową, która wyraża maksymalne wydłużenie wektora jednostkowego w danej normie po przekształceniu przez macierz A . Takie normy nazywa się normami indukowanymi. Normy indukowane, zatem charakteryzują jak dana macierz A rozciąga / przekształca wektory jednostkowe w odniesieniu do danej normy. Wyraża to równanie 2.1.3, które normie przyporządkowuje maksymalne wydłużenie dowolnego wektora x przez macierz A .

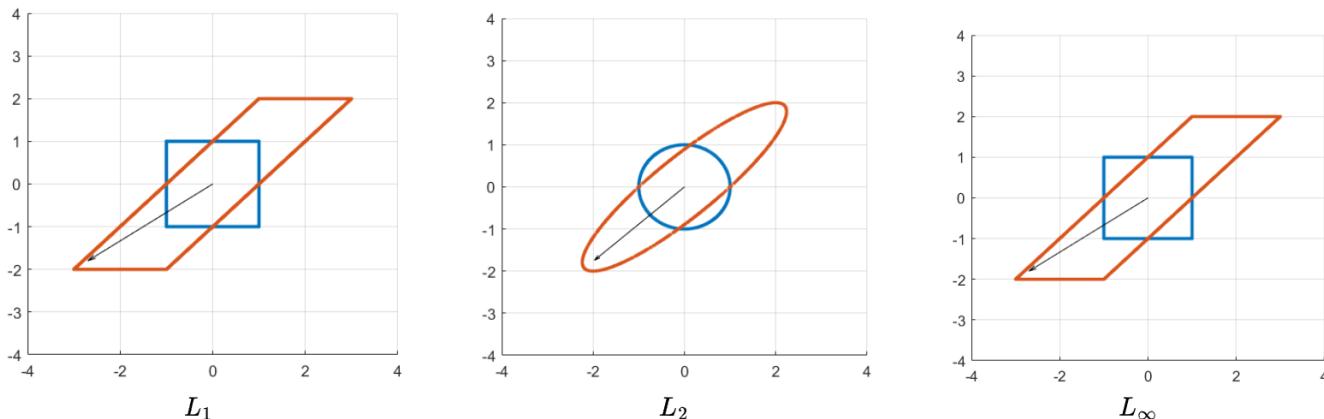
$$\|A\|_p = \max_{\|x\|_p \neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \max_{\|x\|_p=1} \|Ax\|_p \quad (2.3.1)$$

Podobnie do norm wektorowych, normy macierzowe spełniają podstawowe kryteria pozwalające na ich wykorzystanie do porównywania macierzy. Dla skalarów α w K oraz wszystkich macierzy A oraz B w przestrzeni $K^{m \times n}$,

- $\|\alpha A\| = |\alpha| \|A\|$ (skalarność)
- $\|A + B\| \leq \|A\| + \|B\|$ (nierówność trójkątna)
- $\|A\| \geq 0$ (nieujemność)
- $\|A\| = 0$ jeżeli $A = 0_{m,n}$ (jednoznaczność)

Na rysunku 2.3 przedstawiono graficzną interpretację przekształcenia zbioru "kół jednostkowych" dla norm L_1 , L_2 , L_∞ przez przykładową macierz:

$$A = \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix}$$



Rys. 2.3. "Koła jednostkowe" dla trzech definicji norm: L_1 dla p -normy i $p = 1$, L_2 dla normy euklidesowej oraz L_∞ dla normy nieskończoność.

Normy macierzowe są naturalnym rozszerzeniem notacji norm dedykowanej dla wektorów. Na początku wymienimy p -normy indukowane.

Norma L_1 - maksimum

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=0}^n |a_{ij}| \quad (2.3.2)$$

Norma L_∞ - nieskończoność

$$\|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=0}^n |a_{ij}| \quad (2.3.3)$$

Norma L_2

$$\sigma_{\max}(A) = \sqrt{\max \lambda_i} \quad (2.3.4)$$

gdzie σ_{\max} to maksymalna wartość osobienna macierzy A oraz λ_i oznacza wartości własne macierzy $A^H A$ (gdzie A^H to macierz Hermitowska, czyli transponowana macierz sprzężona spełniająca warunek $[a_{ij}] = [a_{ji}]$.

Norma Frobeniusa

$$\|A\|_2 = \sigma_{\max}(A) \leq \left(\sum_{i=0}^n \sum_{j=0}^n |a_{ij}^2| \right)^{1/2} = \|A\|_F \quad (2.3.5)$$

4. Współczynnik uwarunkowania

Powszechnym zastosowaniem norm macierzowych jest obliczanie współczynnika uwarunkowania macierzy $\text{cond}(A)$ informującego o błędzie rozwiązania układu równań $Ax = b$. Współczynnik uwarunkowania określany jest jako stosunek wartości wektora prawych stron b do wartości wektora prawych stron $A^{-1}b$. Założymy, że e jest błędem wektora prawych stron b układu równań $Ax = b \pm e$, wówczas rozwiązanie jest również obarczone tym błędem i wyraża się równaniem:

$$x + \Delta x = A^{-1}b + A^{-1}e$$

zatem stosunek względnego błędu rozwiązania $\frac{\|A^{-1}b\|}{\|A^{-1}e\|}$ do względnego błędu b $\frac{\|e\|}{\|b\|}$ wyraża się:

$$\frac{\frac{\|A^{-1}b\|}{\|e\|}}{\frac{\|b\|}{\|b\|}} = \frac{\|A^{-1}b\| / \|A^{-1}e\|}{\|e\| / \|b\|} = \frac{\|A^{-1}e\|}{\|e\|} \cdot \frac{\|b\|}{\|A^{-1}b\|}$$

Wartość maksymalna tego stosunku może więc postrzegać się jako iloczyn dwóch norm, zgodnie z poniższym przekształceniem:

$$\begin{aligned} \max_{e, b \neq 0} \left\{ \frac{\|A^{-1}e\|}{\|e\|} \cdot \frac{\|b\|}{\|A^{-1}b\|} \right\} &= \max_{e \neq 0} \left\{ \frac{\|A^{-1}e\|}{\|e\|} \right\} \max_{b \neq 0} \left\{ \frac{\|b\|}{\|A^{-1}b\|} \right\} \\ &= \max_{e \neq 0} \left\{ \frac{\|A^{-1}e\|}{\|e\|} \right\} \max_{x \neq 0} \left\{ \frac{\|Ax\|}{\|x\|} \right\} \\ &= \|A^{-1}\| \|A\|. \end{aligned}$$

Wartość maksymalna jest określana właśnie współczynnikiem uwarunkowania:

$$\text{cond}(A) = \|A^{-1}\| \|A\| \geq \|A^{-1}A\| = 1$$

Współczynnik uwarunkowania o wartości 1 nie wprowadza zatem żadnego dodatkowego błędu rozwiązania. Rozwiązanie jest zatem nie gorsze od samych danych wejściowych. Wartość współczynnika uwarunkowania zależy od przyjętych norm. Najczęściej stosowane definicje to:

- współczynnik uzależniony od wartości osobliwych gdy zastosujemy normę L_2

$$\text{cond}(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

- współczynnik uzależniony od wartości minimalnych i maksymalnych gdy macierz jest trójkątna i zastosujemy normę L_∞

$$\text{cond}(A) \geq \frac{\max_i(|a_{ii}|)}{\min_i(|a_{ii}|)}$$

Przykład 2.2

Przykład

Przeanalizujmy wpływ niedokładności danych wejściowych dla przykładowego układu równań wraz z obliczeniem współczynnika uwarunkowania. Rozważamy układ równań $Ax = b$

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 3,999 \end{bmatrix}; b = \begin{bmatrix} 4 \\ 7,999 \end{bmatrix}$$

Współczynnik uwarunkowania tej macierzy możemy obliczyć wyznaczając go z normy L_∞ w następujących krokach:

$$cond(A) = ||A^{-1}|| \cdot ||A||$$

$$||A||_{\infty} = \max_i \sum_j |a_{ij}| = 5,999$$

$$A^{-1} = \frac{1}{\det(A)} A^D = \frac{1}{3.999 - 2 \cdot 2} \begin{bmatrix} 3.999 & -2 \\ -2 & 1 \end{bmatrix} = \begin{bmatrix} -3999 & 2000 \\ 2000 & -1000 \end{bmatrix}$$

$$\| A^{-1} \|_{\infty} = \max_i \sum_j |a_{ij}| = 5999$$

$$cond(A) = ||A^{-1}|| \cdot ||A|| = 5,999 * 5999 = 35988$$

Współczynnik uwarunkowania dla tej macierzy o wymiarach 2×2 jest stosunkowo bardzo duży. Oznacza to, że spodziewamy się znacznego wpływu niedokładności na wynik rozwiązania. Sprawdźmy to.

Obliczmy rozwiązanie układu równań przy założeniu oryginalnego wektora $\begin{bmatrix} 4 \\ 7,999 \end{bmatrix}$. Wówczas rozwiązanie układu równań możemy obliczyć odejmując pierwszy wiersz od drugiego:

$$\begin{array}{rcl}
 x_1 + 2x_2 & = & 4 \\
 2x_1 + 3,999x_2 & = & 7,999 \\
 -2x_1 - 4x_2 & = & -8 \\
 \hline
 2x_1 + 3,999x_2 & = & 7,999 \\
 + - - - - \\
 -0,001x_2 & = & -0,001 \\
 \hline
 & & \rightarrow x_2 = 1
 \end{array}$$

następnie

$$x_1 = 4 - 2x_2 = 4 - 2 \cdot 1 = 2$$

otrzymujemy zatem rozwiązanie:

$$x = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Zmieńmy teraz nieznacznie wektor prawych stron b i przyjmijmy: $b = \begin{bmatrix} 4 \\ 8 \end{bmatrix}$. Wówczas analogicznie przeprowadzająca obliczenia jak powyżej otrzymamy wynik:

$$x = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$$

Jak można zauważyć, niewielka zmiana o wartość 0,001 wektora b spowodowała bardzo dużą zmianę rozwiązania. **Mówimy o takim układzie równań, że jest źle uwarunkowany.**

Zmieńmy współczynniki macierzy A :

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}; b = \begin{bmatrix} 4 \\ 7,999 \end{bmatrix}$$

Gdybyśmy powtórzyli obliczenia współczynnika uwarunkowania dla tej macierzy to otrzymalibyśmy wartość $cond(A) = 25$, która jest mniejsza od poprzedniej o ponad 100 razy (dwa rzedy). Widać, że macierz ta jest znacznie lepiej uwarunkowana. Powtórzmy serie obliczeń

dla oryginalnego $b = \begin{bmatrix} 4 \\ 7,999 \end{bmatrix}$ oraz zmodyfikowanego $b = \begin{bmatrix} 4 \\ 8 \end{bmatrix}$ i porównajmy wyniki.

dla $b = \begin{bmatrix} 4 \\ 7,999 \end{bmatrix}$ otrzymamy wynik $x = \begin{bmatrix} 3,998 \\ 0,001 \end{bmatrix}$,

$$a \text{ dla } b = \begin{bmatrix} 4 \\ 8 \end{bmatrix} \text{ otrzymamy } x = \begin{bmatrix} 4 \\ 0 \end{bmatrix}.$$

Jak widać, tym razem wyniki rozwiązania różnią się nieznacznie.

5. Metoda eliminacji Gaussa

Eliminacja Gaussa jest podstawowym sposobem rozwiązywania układu równań liniowych. W eliminacji Gaussa wykorzystujemy operacje wierszowe dodawania do siebie wierszy układu równań. Wykonujemy jednak te operacje w ścisłe określony sposób i kolejności. Zasadniczym celem eliminacji Gaussa jest przekształcenie układu równań do postaci górnopróbkowej. To znaczy, że wszystkie współczynniki macierzy $\langle A \rangle$ układu równań poniżej diagonali są zerowane. Uważny czytelnik zauważał zapewne, że mówimy o eliminacji elementów macierzy $\langle A \rangle$, ale operacje wykonujemy na całych wierszach układu równań. Nie możemy zatem zapomnieć o wektorze prawych stron $\langle b \rangle$. W celu uproszczenia implementacji algorytmu zazwyczaj tworzy się zazwyczaj macierz rozszerzoną, która powstaje poprzez dołączenie wektora prawych stron $\langle b \rangle$ dodatkowej kolumny do macierzy $\langle A \rangle$.

$$\langle [A|b] \rangle$$

Następnie przekształcamy macierz $\langle Ag \rangle$ do postaci górnopróbkowej za pomocą dodawania wierszy wierszy 'diagonalnych' wymnożonych przez stosowne współczynniki skalujące (odejmujemy wiersze 'diagonalne' od wierszy poniżej). Kolejność operacji (zaprezentowana na rysunku 2.4) jest następująca:

- najpierw odejmujemy pierwszy wiersz macierzy $\langle Ag \rangle$ od drugiego wiersza wymnożony przez współczynnik, który spowoduje po odjęciu wyzerowanie elementu $\langle a_{21} \rangle$, równy $\langle l_{21} = \frac{a_{21}}{a_{11}} \rangle$,
- w kolejnych krokach odejmuję pierwszy wiersz macierzy wymnożony przez stosowne współczynniki, aż wyzerowane zostaną wszystkie elementy poniżej diagonali,
- dalej, przechodzimy do zerowania elementów poniżej diagonali w drugiej kolumnie,
- itd., aż wyzerujemy wszystkie elementy poniżej diagonali.

Powyższy schemat został zilustrowany na rysunku 2.4.

$$\begin{aligned} -l_{21} &= -\left(\frac{a_{21}}{a_{11}}\right) \times \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ \cancel{a_{21}} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \\ -l_{31} &= -\left(\frac{a_{31}}{a_{11}}\right) \times \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ \cancel{a_{31}} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b_3 \end{bmatrix} \\ -l_{32} &= -\left(\frac{a'_{32}}{a'_{22}}\right) \times \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & \cancel{a'_{32}} & a'_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \end{bmatrix} \rightarrow \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b'_2 \\ b''_3 \end{bmatrix} \end{aligned}$$

Rysunek 2.4. Ilustracja przebiegu eliminacji Gaussa

Przykład 2.3

Przykład

Przeprowadź eliminację Gaussa dla poniższego układu równań z trzema niewiadomymi.

$$\langle x_1+3x_2+4x_3=2 \rangle \langle -2x_1+2x_2+3x_3=-1 \rangle \langle x_1+x_2+2x_3=3 \rangle$$

Powyższy układ równań możemy zapisać w postaci macierzowej:

$$\langle \left(\begin{matrix} 1 & 3 & 4 & 2 \\ -2 & 2 & 3 & -1 \\ 1 & 1 & 2 & 3 \end{matrix} \right) \cdot \left(\begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} \right) = \left(\begin{matrix} 2 \\ -1 \\ 3 \end{matrix} \right) \rangle$$

Następnie możemy odjąć pierwszy wiersz od kolejno drugiego i trzeciego:

$$\langle \left(\begin{matrix} 1 & 3 & 4 & 2 \\ -2 & 2 & 3 & -1 \\ 1 & 1 & 2 & 3 \end{matrix} \right) \cdot \left(\begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} \right) = \left(\begin{matrix} 2 \\ -1 \\ 3 \end{matrix} \right) \rangle$$

$$\{1\}) 2\backslash end\{matrix\}\right] \backslash)$$

W wyniku otrzymamy:

$$\left(\begin{pmatrix} 1 & 3 & 4 & 0 & 8 & 11 & 0 & -2 & -2 \end{pmatrix} \cdot \begin{pmatrix} x_1 & x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 & 3 & 1 \end{pmatrix} \right)$$

Kolejnym krokiem jest odjęcie drugiego wiersza od trzeciego w celu eliminacji elementu a_{32} :

$$\left(\begin{pmatrix} 1 & 3 & 4 & 0 & 8 & 11 & 0 & -2 & -2 \end{pmatrix} - \frac{-2}{8} \begin{pmatrix} 2 & 3 & 1 \end{pmatrix} \right) = \begin{pmatrix} 2 & 3 & 1 & 0 & 0 & 11 & 0 & -2 & -2 \end{pmatrix}$$

Ostatecznie otrzymuje górną-trójkątną postać układu równań:

$$\left(\begin{pmatrix} 1 & 3 & 4 & 0 & 8 & 11 & 0 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 & x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 & 3 & 1 & \frac{1}{8} & \frac{1}{8} & \frac{11}{8} \end{pmatrix} \right)$$

W celu implementacji komputerowej algorytmu eliminacji Gaussa warto posłużyć się zapisem algorytmicznym procedury. Zakładając, że operujemy na macierzy rozszerzonej $\begin{pmatrix} A & g \end{pmatrix}$ dla $i, j = 1, 2, \dots, n$ o wymiarach $(n \times n+1)$,

$$a_{ij}^k = a_{ij}^{(k-1)} - \frac{a_{ij}^{(k-1)} \cdot a_{kk}^{(k-1)}}{a_{jj}^{(k-1)}}$$

dla

$\forall (k=1, 2, \dots, n-1)$ (kolejne kolumny bez ostatniej)

$\forall (i=k+1, k+2, \dots, n)$ (kolejne wiersze poniżej k-tego)

$\forall (j=k, k+1, \dots, n+1)$ (wszystkie kolumny z pominięciem już wcześniej wyzerowanych)

Tak sformułowany algorytm pozwala nam wygodnie zaimplementować funkcje w MATLAB, która przyjmuje jako argument macierz A oraz wektor prawych stron b , a zwraca macierz rozszerzoną $\begin{pmatrix} Ag \end{pmatrix}$, która jest górną-trójkątna. Proszę zwrócić uwagę, na bardzo podobne oznaczenia, które ułatwiają interpretację kodu.

```
function Ag = gaussian(A,b)
Ag = [A b];
n = size(Ag,1);
for k=1:n-1
    for i = k+1:n
        l = Ag(i,k) / Ag(k,k);
        for j = k:n+1
            Ag(i,j) = Ag(i,j) - l * Ag(k,j);
        end
    end
end
end
```

Jednym z bardzo ważnych elementów, których nie możemy pominąć jest wrażliwość algorytmu na wystąpienie w dowolnym momencie procesu zero na diagonali, które będzie skutkowało dzieleniem przez zero podczas obliczania współczynników wykorzystywanego do eliminacji. W celu eliminacji tego problemu stosuje się **selekcję elementu głównego**.

Selekcja elementu głównego polega na takiej zamianie wierszy macierzy rozszerzonej $\begin{pmatrix} Ag \end{pmatrix}$, aby w kolejnym kroku algorytmu na diagonali znalazła się **maksymalny co modułu element**, jednocześnie pozostawiając już wyzerowane elementy nadal zerowymi.

Występują trzy rodzaje selekcji:

1. w kolumnie (selekcja częściowa)- najprostsza wymaga tylko zamiany równań (rysunek 2.5a),
2. w wierszu (selekcja częściowa) - zamieniamy kolejność niewiadomych w wektorze x (rysunek 2.5b),
3. podmacierz $A[r:end, c:end]$ (selekcja pełna) - zamieniamy zarówno wiersze jak i kolejność zmiennych w wektorze x (rysunek 2.5c).

$$\begin{array}{c} \text{a)} \\ \left[\begin{array}{ccccc} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{array} \right] \end{array}$$

$$\begin{array}{c} \text{b)} \\ \left[\begin{array}{ccccc} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{array} \right] \end{array}$$

$$\begin{array}{c} \text{c)} \\ \left[\begin{array}{ccccc} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{array} \right] \end{array}$$

Rys. 2.5. Elementy macierzy, w których poszukuje się wartości maksymalnych co do modułu w przypadku procedury selekcji elementu głównego: a) dla selekcji częściowej w kolumnie, b) dla selekcji częściowej w wierszu, c) dla selekcji pełnej w bloku macierzy.

Przykład 2.4

Przykład

Rozwiążemy układ równań stosując skończoną precyzję sztucznie zaokrąglając wyniki obliczeń arytmetycznych aby pokazać wpływ selekcji elementu głównego również na dokładność rozwiązania.

Na początku rozwiążmy układ z następującym układem wierszy.

$$\begin{aligned}
 & \left(\begin{matrix} 0.003 & 59.1 & 5.291 & -6.13 \end{matrix} \right) \left(\begin{matrix} x_1 & x_2 \end{matrix} \right) = \left(\begin{matrix} 59.17 & 46.78 \end{matrix} \right) \\
 & \frac{5.291}{0.003} = 1763.666... \approx 1764.0 \\
 & \left(\begin{matrix} 0 & -104200 \end{matrix} \right) \left(\begin{matrix} x_1 & x_2 \end{matrix} \right) = \left(\begin{matrix} 59.17 & -104300 \end{matrix} \right) \\
 & x_2 = \frac{-104300}{-104200} \approx 1.001 \quad x_1 = \frac{59.17 - 59.1}{-104200} \approx 3.633
 \end{aligned}$$

Teraz, powtórzmy obliczenia, ale wcześniejszej zamieniając miejscami wiersz pierwszy i drugi układu równań.

$$\begin{aligned}
 & \left(\begin{matrix} 5.291 & -6.13 & 0.003 & 59.1 \end{matrix} \right) \left(\begin{matrix} x_1 & x_2 \end{matrix} \right) = \left(\begin{matrix} 46.78 & 59.17 \end{matrix} \right) \\
 & \frac{0.003}{5.291} = 0.000567 \\
 & \left(\begin{matrix} 46.78 & 59.14 \end{matrix} \right) \left(\begin{matrix} x_1 & x_2 \end{matrix} \right) = \frac{46.78 - (-6.13)}{5.291} \approx 10.001
 \end{aligned}$$

Widzimy, że zmiana wierszy (zaznaczmy, że przy zastosowaniu sztucznie zawyżonego błędu zaokrągleń) dała nam wynik znacznie inny od poprzedniego. Który wynik jest poprawny? Ten drugi, ponieważ na diagonali znajdował się największy co do modułu w danej kolumnie element.

Wykonanie tych obliczeń w MATLABie (nawet z oryginalnym układem wierszy) daje nam wynik drugi. Po pierwsze, MATLAB zamienia wiersze automatycznie, po drugie operacje wykonane są z dużo większą precyzją ($\text{eps} \approx 10^{-14}$).

```

A = [0.003 59.1;
      5.291 -6.13];
b = [59.17
      46.78];
A\b

ans =
    10.0008
    1.0007
  
```

6. Wsteczne podstawienie

Jeżeli macierz rozszerzona reprezentująca nasz układ równań jest już w postaci trójkątnej, to wynik rozwiązania bardzo łatwo znaleźć stosując procedurę wstecznego podstawienia. Przyjrzyjmy się przykładowemu układowi równań w postaci górnopróbkątnej:

$$\left(\begin{matrix} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22} & a_{23} & b_2 \\ 0 & 0 & a_{33} & b_3 \end{matrix} \right)$$

Mając taką postać, procedura kolejno znajduje wartości zaczynając od ostatniego, czyli dla $\forall i=n, n-1, \dots, 1$. Otrzymujemy zatem:

$$\begin{aligned} x_3 &= \frac{b_3 - a_{23}x_2 - a_{33}x_1}{a_{33}} \\ x_2 &= \frac{b_2 - a_{12}x_3 - a_{23}x_1}{a_{22}} \\ x_1 &= \frac{b_1 - a_{11}x_3 - a_{12}x_2}{a_{11}} \end{aligned}$$

Zauważmy, że w kolejnych wierszach wykorzystujemy wartości x_i obliczone wcześniej. Powyższą procedurę możemy uogólnić dla macierzy górnopróbkątnej o dowolnym rozmiarze i przedstawić matematycznie:

$$\begin{aligned} x_i &= \frac{c_i - \sum_{j=i+1}^n (u_{ij} \cdot x_j)}{u_{ii}} \quad \text{dla } i = n, n-1, \dots, 1 \quad (\text{2.6.1}) \\ \text{a w przypadku macierzy dolnopróbkątnej, odwrotnie będziemy kolejno obliczać wartości zaczynając tym razem od pierwszego wiersza:} \\ x_i &= \frac{c_i - \sum_{j=1}^{i-1} (l_{ij} \cdot x_j)}{l_{ii}} \quad \text{dla } i = 1, 2, \dots, n \quad (\text{2.6.2}) \end{aligned}$$

Oto dodatkowo dwie funkcje implementujące oba podstawienia.

```
% U - macierz górnopróbkątna, c - wektor prawych stron
function x = wsteczne_gornopróbkątnie(U,c)
    n = size(U,1);
    x = zeros(n,1);
    for i = n:-1:1
        s = 0;
        for j = i+1:n
            s = s + U(i,j)*x(j);
        end
        x(i) = (c(i) - s) / U(i,i);
    end
end

% U - macierz dolnopróbkątna, c - wektor prawych stron
function x = wsteczne_dolnopróbkątnie(L,c)
    n = size(L,1);
    x = zeros(n,1);
    for i = 1:n
        s = 0;
        for j = 1:i-1
            s = s + L(i,j)*x(j);
        end
        x(i) = (c(i) - s) / L(i,i);
    end
end
```

7. Eliminacja Gaussa-Jordana

Eliminacja Gaussa-Jordana jest rozwinięciem eliminacji Gaussa o dodatkowe kroki. W eliminacji tej oprócz zerowania elementów poniżej diagonali układu równań, eliminujemy również elementy powyżej diagonali. Dodatkowo, skalujemy wartości we wszystkich wierszach, tak aby po eliminacji na diagonali były same jedynki. Pamiętając, że operacje, które wykonujemy są operacjami wierszowymi to rozwiązania układu równań oryginalnego i układu po eliminacji są takie same. Łatwo zauważyć, że po eliminacji Gaussa-Jordana, ponieważ w części macierzy A będziemy mieli wyzerowane wszystkie elementy poniżej i powyżej diagonali oraz na diagonali będą wartości , to ostatnia kolumna macierzy rozszerzonej będzie rozwiązaniem układu równań. Nie potrzebujemy zatem wstecznego podstawienia. Pozornie, algorytm wydaje się być szybszy obliczeniowo, ale z uwagi na konieczność wyzerowania elementów powyżej diagonali jego złożoność jest praktycznie taka sama jak eliminacji Gaussa połączonej z wstecznym podstawieniem. Niemniej występują sytuacje, że zastosowanie eliminacji Gaussa-Jordana jest korzystne. Jedną z nich jest np. konieczność obliczenia macierzy odwrotnej. Przebieg algorytmu eliminacji Gaussa-Jordana jest następujący:

1. Inicjalizacja (definicja macierzy rozszerzonej)

$$\forall (a_{ij})^{\{0\}} = a_{ij} \quad \text{dla } (i=1,2,\dots,n; j=1,2,\dots,n)$$

$$\forall (a_{ij})^{\{0\}} = b_i \quad \text{dla } (i=1,2,\dots,n; j=n+1)$$

2. Normalizacja (znormalizować element diagonalny do wartości 1)

$$\forall (a_{kj})^{\{0\}} = \frac{a_{kj}}{\sqrt{a_{kk}^{\{0\}}}} \quad \text{dla } (j=k, k+1, \dots, n+1)$$

3. Redukcja (zredukować wszystkie elementy pozadiagonalne w kolumnie k)

$$\forall (a_{ij})^{\{0\}} = a_{ij}^{\{k-1\}} - a_{ik}^{\{k-1\}} \cdot a_{kj}^{\{k\}} \quad \text{for } j=k, k+1, \dots, n+1; i=1, 2, \dots, n \text{ (and } i \neq k \text{)}$$

Kroki 2-3 muszą zostać wykonane dla wszystkich kolumn $(k=1,2,\dots,n)$.

Przykład 2.5

Przykład

Znajdź rozwiązanie układu równań przedstawionego w postaci macierzy rozszerzonej.

1. W pierwszym kroku podzielimy wszystkie elementy pierwszego wiersza przez wartość elementu diagonalnego $\sqrt{2}$.
 2. W drugim kroku odejmujemy pierwszy wiersz od drugiego wymnożony przez $\sqrt{3}$.
 3. W trzecim kroku odejmujemy pierwszy wiersz od trzeciego wymnożony przez $\sqrt{5}$.
 4. W czwartym kroku podzielimy elementy w drugim wierszu przez wartość diagonalną równą $\sqrt{5}$.
 5. W piątym kroku odejmujemy drugi wiersz od pierwszego wymnożony przez $\sqrt{7}$.
 6. W szóstym kroku odejmujemy drugi wiersz od trzeciego wymnożony przez $\sqrt{7}$.
 7. ... kontynuując obliczenia otrzymamy ostateczny wynik w czwartej kolumnie macierzy rozszerzonej: $x=[1,2,4]$.
- $$\left(\begin{array}{cccc|c} 2 & 1 & 4 & 3 & 1 \\ 1 & 0 & 1 & 2 & 0 \\ 1 & 1 & 0 & 1 & 2 \\ 1 & 2 & 0 & 0 & 1 \end{array} \right) \xrightarrow{\text{R1} \rightarrow \frac{1}{\sqrt{2}} \text{R1}} \left(\begin{array}{cccc|c} 1 & 0 & 2 & 1.5 & 0.5 \\ 1 & 0 & 1 & 2 & 0 \\ 1 & 1 & 0 & 1 & 2 \\ 1 & 2 & 0 & 0 & 1 \end{array} \right) \xrightarrow{\text{R2} \rightarrow \text{R2} - \sqrt{3}\text{R1}} \left(\begin{array}{cccc|c} 1 & 0 & 2 & 1.5 & 0.5 \\ 0 & 0 & 1 & 2-\sqrt{3} & 0 \\ 1 & 1 & 0 & 1 & 2 \\ 1 & 2 & 0 & 0 & 1 \end{array} \right) \xrightarrow{\text{R3} \rightarrow \text{R3} - \sqrt{5}\text{R1}} \left(\begin{array}{cccc|c} 1 & 0 & 2 & 1.5 & 0.5 \\ 0 & 0 & 1 & 2-\sqrt{3} & 0 \\ 0 & 1 & -2 & 1-\sqrt{5} & 1-\sqrt{5} \\ 1 & 2 & 0 & 0 & 1 \end{array} \right) \xrightarrow{\text{R4} \rightarrow \text{R4} - \sqrt{7}\text{R1}} \left(\begin{array}{cccc|c} 1 & 0 & 2 & 1.5 & 0.5 \\ 0 & 0 & 1 & 2-\sqrt{3} & 0 \\ 0 & 1 & -2 & 1-\sqrt{5} & 1-\sqrt{5} \\ 0 & 2 & -4 & 0.5-\sqrt{7} & 0.5-\sqrt{7} \end{array} \right) \xrightarrow{\text{R2} \rightarrow \text{R2} - \sqrt{7}\text{R3}} \left(\begin{array}{cccc|c} 1 & 0 & 2 & 1.5 & 0.5 \\ 0 & 0 & 1 & 2-\sqrt{3} & 0 \\ 0 & 1 & -2 & 1-\sqrt{5} & 1-\sqrt{5} \\ 0 & 0 & 0 & 0.5-\sqrt{7} & 0.5-\sqrt{7} \end{array} \right) \xrightarrow{\text{R3} \rightarrow \text{R3} + 2\text{R2}} \left(\begin{array}{cccc|c} 1 & 0 & 2 & 1.5 & 0.5 \\ 0 & 0 & 1 & 2-\sqrt{3} & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right)$$

Jak wcześniej wspomiano, eliminację Gaussa-Jordana można skutecznie wykorzystać do obliczania macierzy odwrotnej o niewielkim rozmiarze. Aby wyprowadzić ten sposób przypomnijmy definicję macierzy odwrotnej:

$$(AA^{-1}) = \left(\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & a_{21} & a_{22} & \dots & a_{2n} \\ a_{21} & a_{22} & \dots & a_{2n} & a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & a_{m1} & a_{m2} & \dots & a_{mn} \end{array} \right) \xrightarrow{\text{R1} \rightarrow \text{R1} - a_{11} \text{R1}, \text{R2} \rightarrow \text{R2} - a_{21} \text{R1}, \dots, \text{Rm} \rightarrow \text{Rm} - a_{m1} \text{R1}} \left(\begin{array}{cccc|c} 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 0 \end{array} \right) \quad (2.7.1)$$

Zauważmy, że w tym równaniu pierwszą i kolejne kolumny macierzy (A^{-1}) możemy oznaczyć jako zmienne (x_{ij})

$$\begin{aligned} & \left(\begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{matrix} \right) \cdot \left(\begin{matrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{matrix} \right) = \left(\begin{matrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{matrix} \right) \end{aligned}$$

Przy takim zapisie, okazuje się, że aby znaleźć kolejne kolumny macierzy odwrotnej (A^{-1}) wystarczy rozwiązać (n) niezależnych układów równań, w których niewiadomymi będą kolumny macierzy (A^{-1}) a wektorami prawych stron kolejne kolumny macierzy jednostkowej.

$$\begin{aligned} & \left(\begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{matrix} \right) \cdot \left(\begin{matrix} x_{11} \\ x_{21} \\ \vdots \\ x_{m1} \end{matrix} \right) = \left(\begin{matrix} 1 \\ 0 \\ \vdots \\ 0 \end{matrix} \right) \\ & \left(\begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{matrix} \right) \cdot \left(\begin{matrix} x_{12} \\ x_{22} \\ \vdots \\ x_{m2} \end{matrix} \right) = \left(\begin{matrix} 0 \\ 1 \\ \vdots \\ 0 \end{matrix} \right) \end{aligned}$$

i tak dalej.

Możemy jednak podejść do tego zagadnienia skuteczniej i zamiast rozwiązywać osobnych układów równań rozwiążemy jeden ale z wszystkimi wektorami prawych stron (całą macierzą jednostkową) doklejoną do macierzy (A) .

$$\left(\begin{matrix} a_{11} & a_{12} & \dots & a_{1n} & 1 & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & a_{2n} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & 0 & 0 & \dots & 1 \end{matrix} \right)$$

Wykorzystamy do tego metodę eliminacji Gaussa-Jordana, w wyniku której przekształcimy układ do postaci

$$\left(\begin{matrix} 1 & 0 & \dots & 0 & a'_{11} & a'_{12} & \dots & a'_{1n} & 0 & 1 & \dots & 0 \\ 0 & 1 & \dots & 0 & a'_{21} & a'_{22} & \dots & a'_{2n} & 0 & 0 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & a'_{m1} & a'_{m2} & \dots & a'_{mn} & 0 & 0 & \dots & 1 \end{matrix} \right)$$

gdzie współczynniki (a'_{ij}) stanowią współczynniki szukanej macierzy odwrotnej.

Przykład 2.6

Przykład

Wykorzystując eliminację Gaussa-Jordana znajdź macierz odwrotną do (A) :

$$(A = \left[\begin{matrix} 2 & 1 & 4 & 5 \end{matrix} \right])$$

Rozwiązanie

$$\begin{aligned} & \left(\begin{matrix} 2 & 1 & 4 & 5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 2 & 2.5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \\ & \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \\ & \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \rightarrow \left(\begin{matrix} 1 & 0 & 0 & -0.5 \end{matrix} \right) \end{aligned}$$

Odpowiedź

$$(A^{-1} = \left[\begin{matrix} \frac{5}{6} & -\frac{1}{6} & 0 & 0 \\ -\frac{1}{6} & \frac{5}{6} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{1}{2} \end{matrix} \right])$$

8. Rozkład na czynniki - faktoryzacja macierzy

Innym sposobem rozwiązywania układów równań jest wstępny rozkład na czynniki macierzy (A) . W algebrze liniowej stosuje się powszechnie kilka rodzajów faktoryzacji:

1. Faktoryzacja LU: $(A=L \cdot U)$, macierz L jest macierzą dolno-trójkątną, a macierz U górnou-trójkątną,
2. Faktoryzacja QR: $(A=Q \cdot R)$, macierz Q jest macierzą ortonormalną $(Q^T Q = 1 \rightarrow Q^{-1} = Q^T)$, macierz jest macierzą górnou-trójkątną,
3. Faktoryzacja SVD: $(A=S \cdot V)$, macierze (S) , (V) są ortogonalne, a (D) diagonalna.

W ramach niniejszego podręcznika przyjrzymy się wyłącznie faktoryzacji (LU) . Zakładając w właściwości górnou i dolno-trójkątne macierzy (L) i (U) oryginalny układ równań możemy zapisać następująco:

$$(Ax=b)$$

$$(LUx=b)$$

Rozwiązanie możemy uzyskać dwukrotnie stosując wsteczne podstawienie. Najpierw podstawimy $(Ux=y)$, wówczas uzyskamy

$$(Ly=b)$$

Macierz (L) jest dolno-trójkątna więc rozwiązywanie tego równani moźemy szybko znaleźć stosując wsteczne podstawienie od góry. Znając wynik (y) , moźemy przejść do drugiego etapu, korzystając z wprowadzonego wcześniej podstawienia:

$$(Ux=y)$$

Wykorzystując ponownie wsteczne podstawienie uzyskamy ostateczne rozwiązanie.

8.1. Metoda Doolittle'a

Pozostaje pytanie jak skutecznie znaleźć rozkład LU. Pierwszym podejściem jest zastosowanie **metody Doolittle'a**. Aby wyprowadzić tą metodę posłużmy się pełnym zapisem faktoryzacji LU:

$$\backslash(A=LU)$$

$$\backslash(\left(\begin{matrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{matrix} \right) = \left(\begin{matrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \dots & \dots & \dots & 0 \\ l_{n1} & l_{n2} & \dots & 1 \end{matrix} \right) \left(\begin{matrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & u_{nn} \end{matrix} \right))$$

Zapis ten pozwoli nam wyprowadzić procedurę Doolittle'a bezpośrednio z definicji faktoryzacji. W metodzie tej kluczowa jest kolejność w jakiej będziemy wyznaczać współczynniki kolejno parami poszczególnych wierszy macierzy $\backslash(U)$ i kolumn macierzy $\backslash(L)$.

- Najpierw wyznaczamy współczynniki pierwszego wiersza macierzy $\backslash(U)$. Zgodnie ze schematem na rysunku 2.6 wartości współczynników macierzy $\backslash(A)$ w pierwszym są równe:

$$\backslash(\begin{aligned} a_{11} &= 1 u_{11} + 0 \cdot u_{12} + \dots + 0 \cdot u_{1n} \\ a_{12} &= 1 u_{21} + 0 \cdot u_{22} + \dots + 0 \cdot u_{2n} \\ \vdots &= \vdots u_{n1} + 0 \cdot u_{n2} + \dots + 0 \cdot u_{nn} \end{aligned})$$

Stąd błyskawicznie (z uwagi na większość zer), możemy wyznaczyć $\backslash(u_{1i}=a_{1i})$ dla $\backslash(i=1,2,\dots,n)$.

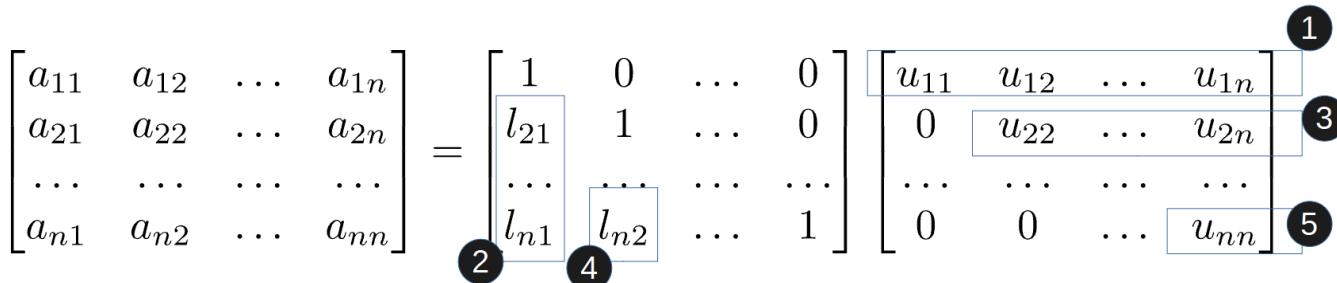
W kolejnym drugim kroku wyznaczamy współczynniki pierwszej kolumny macierzy $\backslash(L)$:

$$\backslash(\begin{aligned} a_{21} &= l_{21} u_{11} + 1 \cdot u_{22} + \dots + 0 \cdot u_{2n} \\ a_{31} &= l_{31} u_{11} + l_{32} u_{21} + 1 \cdot u_{32} + \dots + 0 \cdot u_{3n} \\ \vdots &= \vdots l_{n1} u_{11} + l_{n2} u_{21} + \dots + l_{nn} u_{nn} \end{aligned})$$

stąd łatwo wyprowadzić wyrażenia na wartości $\backslash(l_{i1})$ zakładając, że wartości $\backslash(u_{11})$ została już obliczona w poprzednim kroku. W trzecim kroku algorytmu wróćmy do macierzy $\backslash(U)$, tym razem do drugiego wiersza, analogicznie wyprowadzając wzory na współczynniki z operacji mnożenia wiersza razy kolumna:

$$\backslash(\begin{aligned} a_{22} &= l_{21} u_{12} + 1 u_{22} + \dots + 0 \cdot u_{2n} \\ a_{32} &= l_{31} u_{12} + l_{32} u_{21} + 1 u_{32} + \dots + 0 \cdot u_{3n} \\ \vdots &= \vdots l_{n1} u_{12} + l_{n2} u_{21} + \dots + l_{nn} u_{nn} \end{aligned})$$

i tak dalej.



Rys. 2.6. Schemat z kolejnością obliczeń wierszy i kolumn macierzy $\backslash(L)$ i $\backslash(U)$ w faktoryzacji $\backslash(LU)$.

Wykorzystując ten schemat możemy określić algorytm faktoryzacji LU metodą Doolittle'a w następujący sposób:

- pierwszy wiersz U kopujemy z pierwszego wiersza A
 $\backslash(u_{1i}=a_{1i})$ dla $\backslash(i=1,2,\dots,n)$
- pierwsza kolumna L obliczana jest za pomocą:
 $\backslash(l_{i1}=a_{i1}/u_{11})$ dla $\backslash(i=1,2,\dots,n)$ dla $\backslash(i=2,3,\dots,n)$
- następnie dla każdej pary $\backslash(i^t y)$ wiersz $\backslash(U)$ oraz $\backslash(i^t a)$ kolumna $\backslash(L)$:

$$\backslash(\begin{aligned} \text{dla } i=2,3, \dots, n, \text{ dla } j=1, \dots, i-1, \text{ dla } k=i+1, \dots, n \\ l_{ik} = a_{ik} - \sum_{j=1}^{i-1} l_{ij} u_{jk} \end{aligned})$$

Implementacja tego kodu w postaci funkcji MATLABa jest następująca.

```
function [L, U] = doolittle(A)
n = size(A,1);
L = eye( n ); % inicjujemy macierz jednsotkowa, poniewaz zawsze na diagonali sa jedynki
U = zeros( n ); % pusta (na razie) macierz gornotrojkatna
U(1,: ) = A(1,: ) % kopujemy pierwszy wiersz
L(2:n,1) = A(2: n ,1) / U(1,1); % obliczamy pierwsza kolumnę

% wykonujemy parami obliczenia kolejno wierszy U i kolumn L
for i = 2:n
    for k = i:n
        s = 0;
        for j=1:i-1
            s = s + L(i,j)*U(j,k);
        end
        U(i,k) = A(i,k) - s;
    end
    for k = i+1:n
        s = 0;
        for j=1:i-1
            s = s + L(k,j)*U(j,i);
        end
        L(k,i) = (A(k,i) - s) / U(i,i);
    end
end
end
```

Przykład 2.7

Przykład

Wykorzystując przykładową, powyższą funkcję MATLABa oraz wcześniej zdefiniowane funkcje do wstecznego podstawienia znajdź rozwiązańe układu równań:

$$\begin{pmatrix} 1 & -1 & 1 & 1 \\ 4 & 3 & -1 & 2 \\ 3 & 2 & 2 & 5 \\ 8 & 9 & 5 & 8 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 \\ 6 \\ 15 \\ 1 \end{pmatrix}$$

Rozwiązanie:

```
function L04_lu
A = [1 -1 1 1
      4 3 -1 2
      3 2 2 5
      8 9 5 8];
b = [4 6 15 1]';
[L, U] = doolittle(A)
A = L*U % powinna być macierz zerowa
y = wsteczne_dolnotrojkatne(L,b)
x = wsteczne_gornotrojkatne(U, y)

% sprawdzam rozwiazanie - norma powinna być zero
norm(A*x-b)
end

L =
1.0000      0      0      0
4.0000    1.0000      0      0
3.0000    0.7143    1.0000      0
8.0000    2.4286    3.5556    1.0000

U =
1.0000   -1.0000    1.0000    1.0000
0     7.0000   -5.0000   -2.0000
0       0     2.5714    3.4286
0       0       0    -7.3333

ans =
0   0   0   0
0   0   0   0
0   0   0   0
0   0   0   0

y =
4.0000
-10.0000
10.1429
-42.7778

x =
-0.5000
-2.5000
-3.8333
5.8333

ans =
1.9860e-15
```

8.2. Metoda eliminacji Gaussa

Drugim, najbardziej użytecznym z praktycznego punktu widzenia sposobem jest wykorzystanie eliminacji Gaussa. Znaczenie tego podejścia jest bardzo istotne, gdyż pozwala na selekcję elementu głównego w kolejnych krokach metody. Metoda Doolittle'a nie pozwala na to. Dzięki selekcji jesteśmy zabezpieczeni przed dzieleniem przez zero na diagonali oraz redukujemy błędy zaokrągleń.

Algorytm faktoryzacji LU z wykorzystaniem eliminacji Gaussa jest bardzo prosty do implementacji. Pomijając szczegóły związane z wyprowadzeniem (wyrazilibyśmy operacje wierszowe za pomocą operatorów macierzowych oraz metodą Gaussa-Jordana wyprowadziliśmy macierze odwrotne tych operatorów) przedstawmy algorytm.

Faktoryzacja (LU) metodą eliminacji Gaussa przebiega zgodnie z procesem eliminacji, ale w trakcie procesu współczynniki (l_{ij}) , które używaliśmy do wymnożenia macierzy diagonalnych podczas zerowania elementów zapamiętujemy w odpowiednich miejscach (ij) macierzy (L) . Macierz wynikowa eliminacji Gaussa staje się wynikową macierzą (U) .

Zatem, współczynnik (l_{21}) z rysunku 2.7 wstawiamy w miejsce $(2,1)$ macierzy docelowej (L) (patrz rysunek 2.8).

$$-l_{21} = -\left(\frac{a_{21}}{a_{11}}\right) \times \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ \cancel{a_{21}} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Rys. 2.7. Ilustracja operacji odejmowania pierwszego wiersza macierzy w trakcie eliminacji Gaussa od wiersza drugiego, z zaznaczonym współczynnikiem (l_{21}) , który wykorzystywany jest do wstawienia do macierzy (L) .

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix}$$

Rys. 2.8. Ilustracja z zaznaczonym elementem (l_{21}) macierzy (L) .

Implementacja faktoryzacji (LU) z wykorzystaniem eliminacji Gaussa w środowisku MATLAB (bez selekcji elementu głównego) została przedstawiona poniżej.

```
function [L, U] = lu_gaussian( A )

n = size( A, 1 );
L = eye( n );
for j = 1:n-1
    for i = j+1:n
        f = A( i,j ) / A( j,j );
        A( i, : ) = A( i, : ) - f*A( j,: );
        L( i, j ) = f; % tutaj zapamietujemy współczynnik
    end
end
U = A;
end
```

Przykład 2.8

Przykład

Wykorzystaj implementację faktoryzacji LU na przykładowej macierzy losowej o rozmiarach (4×4) . Sprawdź wynik faktoryzacji

obliczając normę $\|LU-A\|$

Rozwiążanie

```
function l05_lu_gaussian
    A = rand( 4 )
    [L, U] = lu_gaussian( A )
    norm( L * U - A, 2 )
end

>> l05_lu_gaussian
A =
    0.8147    0.6324    0.9575    0.9572
    0.9058    0.0975    0.9649    0.4854
    0.1270    0.2785    0.1576    0.8003
    0.9134    0.5469    0.9706    0.1419

L =
    1.0000      0      0      0
    1.1118    1.0000      0      0
    0.1559   -0.2972    1.0000      0
    1.1211    0.2676    3.5869    1.0000

U =
    0.8147    0.6324    0.9575    0.9572
    0   -0.6055   -0.0996   -0.5788
    0      0   -0.0212    0.4791
    0      0      0   -2.4948

ans =
    2.2204e-16
```

normy