Lineární algebra - přímé metody

Numerické metody

11. března 2018

FJFI ČVUT v Praze

Úvod

Úvod - Lineární algebraické rovnice

Základní úlohou v lineární algebře je vyřešit

$$\mathbf{A}\vec{x} = \vec{b},$$

kde $\mathbf{A} \in \mathbb{C}^{n,n}$ je regulární matice, $\vec{b} \in \mathbb{C}^n$ je vektor pravé strany a $\vec{x} \in \mathbb{C}^n$ je hledaný vektor řešení.

Úvod - Značení

Prvky matice $\mathbf{A}$ značíme (první index značí řádek a druhý sloupec)

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix},$$

sloupcového vektoru $\vec{b}$ a řešení $\vec{x}$

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \qquad \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}.$$

Úvod - Řešení - inverzní matice

Základní metodou, jak takovou úlohu vyřešit, je najít inverzní matici k matici $\mathbf{A}$, tedy $\mathbf{A}^{-1}$. Řešení je pak jednoduché násobení matice a vektoru

$$\vec{x} = \mathbf{A}^{-1}\vec{b}.$$

Úvod - Řešení - trojúhelníkový tvar

Druhou možnost je převést matici na $\triangle$ (trojúhelníkový) tvar.
Horní, resp. dolní $\triangle$ matice pak vypadá

$$\begin{pmatrix} \tilde{a}_{11} & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\ 0 & \tilde{a}_{22} & \cdots & \tilde{a}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{a}_{nn} \end{pmatrix}, \text{ resp. } \begin{pmatrix} \tilde{a}_{11} & 0 & \cdots & 0 \\ \tilde{a}_{21} & \tilde{a}_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{a}_{n1} & \tilde{a}_{n2} & \cdots & \tilde{a}_{nn} \end{pmatrix}.$$

Tímto procesem musí projít i vektor pravé strany $\vec{b}$, získáme tedy $\vec{\tilde{b}}$.

Úvod - Řešení - trojúhelníkový tvar

Výsledný vektor je už pak dán jednoduchými rovnice. Pro horní $\triangle$ tvar

$$
\begin{aligned}
x_n &= \frac{\tilde{b}_n}{\tilde{a}_{nn}}\,, \\
x_{n-1} &= \frac{\tilde{b}_{n-1} - \tilde{a}_{(n-1)n} x_n}{\tilde{a}_{(n-1)(n-1)}}\,, \\
&\vdots \\
x_1 &= \frac{\tilde{b}_1 - \sum_{i=2}^{n} \tilde{a}_{1i} x_i}{\tilde{a}_{11}}\,.
\end{aligned}
$$

Přímé metody

Přímé metody - Úvod

Právě převodem na inverzní matici nebo $\triangle$ tvar se zabývají přímé metody.

Základem těchto metod je Gaussova, resp. Gaussova-Jordanova eliminace, která převadí matici na $\triangle$ tvar, resp. inverní matici.

Tato metoda je založena na jednoduchých regulárních operací s maticí.

Přímé metody - Gaussova-Jordanova eliminace

Příklad na Gaussovu-Jordanovu eliminace. Vezmeme si matici

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 3 & 5 \\ 4 & 0 & 5 \end{pmatrix}$$

a vektor pravých stran

$$\vec{b} = \begin{pmatrix} 5 \\ 8 \\ 2 \end{pmatrix}.$$

Přímé metody - Gaussova-Jordanova eliminace

Nyní dáme matici $\mathbf{A}$, vektor $\vec{b}$ a jednotkovou matici do jedné matice

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 1 & 5 & 1 & 0 & 0 \\ 2 & 3 & 5 & 8 & 0 & 1 & 0 \\ 4 & 0 & 5 & 2 & 0 & 0 & 1 \end{array}\right)$$

a s ní budeme provádět základní operace.

Přímé metody - Gaussova-Jordanova eliminace

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 1 & 5 & 1 & 0 & 0 \\ 2 & 3 & 5 & 8 & 0 & 1 & 0 \\ 4 & 0 & 5 & 2 & 0 & 0 & 1 \end{array}\right)$$

$$\underbrace{\qquad\qquad}_{\mathbf{A}} \quad \underbrace{\qquad}_{\vec{b}} \quad \underbrace{\qquad\qquad}_{\mathbb{I}}$$

$$\xrightarrow{R_2 - 2R_1}$$

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 1 & 5 & 1 & 0 & 0 \\ 0 & 1 & 3 & -2 & -2 & 1 & 0 \\ 4 & 0 & 5 & 2 & 0 & 0 & 1 \end{array}\right)$$

Přímé metody - Gaussova-Jordanova eliminace

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 1 & 5 & 1 & 0 & 0 \\ 0 & 1 & 3 & -2 & -2 & 1 & 0 \\ 4 & 0 & 5 & 2 & 0 & 0 & 1 \end{array}\right)$$

$$\overset{R_3 - 4R_1}{\longrightarrow}$$

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 1 & 5 & 1 & 0 & 0 \\ 0 & 1 & 3 & -2 & -2 & 1 & 0 \\ 0 & -4 & 1 & -18 & -4 & 0 & 1 \end{array}\right)$$

Přímé metody - Gaussova-Jordanova eliminace

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 1 & 5 & 1 & 0 & 0 \\ 0 & 1 & 3 & -2 & -2 & 1 & 0 \\ 0 & -4 & 1 & -18 & -4 & 0 & 1 \end{array}\right)$$

$$\xrightarrow{R_3+4R_2}$$

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 1 & 5 & 1 & 0 & 0 \\ 0 & 1 & 3 & -2 & -2 & 1 & 0 \\ 0 & 0 & 13 & -26 & -12 & 4 & 1 \end{array}\right)$$

V této chvíli by Gaussova eleminace skončila. Gaussova-Jordanova elimanace se ale snaží převést původní matici na diagonální, resp. jednotkovou.

Přímé metody - Gaussova-Jordanova eliminace

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 1 & 5 & 1 & 0 & 0 \\ 0 & 1 & 3 & -2 & -2 & 1 & 0 \\ 0 & 0 & 13 & -26 & -12 & 4 & 1 \end{array}\right)$$

$$\overset{\frac{1}{13}R_3}{\longrightarrow}$$

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 1 & 5 & 1 & 0 & 0 \\ 0 & 1 & 3 & -2 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & -\frac{12}{13} & \frac{4}{13} & \frac{1}{13} \end{array}\right)$$

Přímé metody - Gaussova-Jordanova eliminace

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 1 & 5 & 1 & 0 & 0 \\ 0 & 1 & 3 & -2 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & -\frac{12}{13} & \frac{4}{13} & \frac{1}{13} \end{array}\right)$$

$$\xrightarrow{R_2 - 3R_3}$$

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 1 & 5 & 1 & 0 & 0 \\ 0 & 1 & 0 & 4 & \frac{10}{13} & \frac{1}{13} & \frac{-3}{13} \\ 0 & 0 & 1 & -2 & \frac{-12}{13} & \frac{4}{13} & \frac{1}{13} \end{array}\right)$$

Přímé metody - Gaussova-Jordanova eliminace

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 1 & 5 & 1 & 0 & 0 \\ 0 & 1 & 0 & 4 & \frac{10}{13} & \frac{1}{13} & \frac{-3}{13} \\ 0 & 0 & 1 & -2 & \frac{-12}{13} & \frac{4}{13} & \frac{1}{13} \end{array}\right)$$

$$\xrightarrow{R_1 - R_3}$$

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 0 & 7 & \frac{25}{13} & \frac{-4}{13} & \frac{-1}{13} \\ 0 & 1 & 0 & 4 & \frac{10}{13} & \frac{1}{13} & \frac{-3}{13} \\ 0 & 0 & 1 & -2 & \frac{-12}{13} & \frac{4}{13} & \frac{1}{13} \end{array}\right)$$

Přímé metody - Gaussova-Jordanova eliminace

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 0 & 7 & \frac{25}{13} & \frac{-4}{13} & \frac{-1}{13} \\ 0 & 1 & 0 & 4 & \frac{10}{13} & \frac{1}{13} & \frac{-3}{13} \\ 0 & 0 & 1 & -2 & \frac{-12}{13} & \frac{4}{13} & \frac{1}{13} \end{array}\right)$$

$$\overset{R_1 - R_2}{\longrightarrow}$$

$$\left(\begin{array}{ccc|c|ccc} 1 & 0 & 0 & 3 & \frac{15}{13} & \frac{-5}{13} & \frac{2}{13} \\ 0 & 1 & 0 & 4 & \frac{10}{13} & \frac{1}{13} & \frac{-3}{13} \\ 0 & 0 & 1 & -2 & \frac{-12}{13} & \frac{4}{13} & \frac{1}{13} \end{array}\right)$$

Přímé metody - Gaussova-Jordanova eliminace

Výsledkem je tedy matice

$$\left(\begin{array}{ccc|c|ccc} \multicolumn{3}{c|}{\underbrace{\begin{matrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{matrix}}_{\mathbb{I}}} & \underbrace{\begin{matrix}3\\ 4\\ -2\end{matrix}}_{\vec{x}} & \multicolumn{3}{c}{\underbrace{\begin{matrix}\frac{15}{13} & \frac{-5}{13} & \frac{2}{13}\\ \frac{10}{13} & \frac{1}{13} & \frac{-3}{13}\\ \frac{-12}{13} & \frac{4}{13} & \frac{1}{13}\end{matrix}}_{\mathbf{A}^{-1}}} \end{array}\right) .$$

Přímé metody - Přímá Gaussova-Jordanova eliminace

Toto je klasická Gaussova-Jordanova eliminace, která se dělá v **LA**.

V případě programování je to ale ekvivalentní jako Gaussova eliminace + zpětný běh.

Proto se postupuje jinak a eliminují se v každém kroku všechny řádky mimo diagonálu

Přímé metody - Přímá Gaussova-Jordanova eliminace - příklad

První krok je stejný

$$\left(\underbrace{\begin{matrix} 1 & 1 & 1 \\ 2 & 3 & 5 \\ 4 & 0 & 5 \end{matrix}}_{\mathbf{A}} \middle| \underbrace{\begin{matrix} 5 \\ 8 \\ 2 \end{matrix}}_{\vec{b}} \middle| \underbrace{\begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}}_{\mathbb{I}}\right)$$

$$\overset{R_2 - 2R_1}{\longrightarrow}$$

$$\left(\begin{matrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 4 & 0 & 5 \end{matrix} \middle| \begin{matrix} 5 \\ -2 \\ 2 \end{matrix} \middle| \begin{matrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{matrix}\right)$$

- druhý

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 1 & 5 & 1 & 0 & 0 \\ 0 & 1 & 3 & -2 & -2 & 1 & 0 \\ 4 & 0 & 5 & 2 & 0 & 0 & 1 \end{array}\right)$$

$$\stackrel{R_3-4R_1}{\longrightarrow}$$

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 1 & 5 & 1 & 0 & 0 \\ 0 & 1 & 3 & -2 & -2 & 1 & 0 \\ 0 & -4 & 1 & -18 & -4 & 0 & 1 \end{array}\right)$$

Přímé metody - Přímá Gaussova-Jordanova eliminace - příklad

U třetího už ale eliminujeme i první řádek

$$\left(\begin{array}{ccc|c|ccc} 1 & 1 & 1 & 5 & 1 & 0 & 0 \\ 0 & 1 & 3 & -2 & -2 & 1 & 0 \\ 0 & -4 & 1 & -18 & -4 & 0 & 1 \end{array}\right)$$

$$\xrightarrow{R_1 - R_2}$$

$$\left(\begin{array}{ccc|c|ccc} 1 & 0 & -2 & 7 & 3 & -1 & 0 \\ 0 & 1 & 3 & -2 & -2 & 1 & 0 \\ 0 & -4 & 1 & -18 & -4 & 0 & 1 \end{array}\right)$$

Přímé metody - Přímá Gaussova-Jordanova eliminace - příklad

$$\left(\begin{array}{ccc|c|ccc} 1 & 0 & -2 & 7 & 3 & -1 & 0 \\ 0 & 1 & 3 & -2 & -2 & 1 & 0 \\ 0 & -4 & 1 & -18 & -4 & 0 & 1 \end{array}\right)$$

$$\xrightarrow{R_3+4R_2}$$

$$\left(\begin{array}{ccc|c|ccc} 1 & 0 & -2 & 7 & 3 & -1 & 0 \\ 0 & 1 & 3 & -2 & -2 & 1 & 0 \\ 0 & 0 & 13 & -26 & -12 & 4 & 1 \end{array}\right)$$

Přímé metody - Přímá Gaussova-Jordanova eliminace - příklad

$$\left(\begin{array}{ccc|c|ccc} 1 & 0 & -2 & 7 & 3 & -1 & 0 \\ 0 & 1 & 3 & -2 & -2 & 1 & 0 \\ 0 & 0 & 13 & -26 & -12 & 4 & 1 \end{array}\right)$$

$$\overset{R_2 - \frac{3}{13}R_3}{\longrightarrow}$$

$$\left(\begin{array}{ccc|c|ccc} 1 & 0 & -2 & 7 & 3 & -1 & 0 \\ 0 & 1 & 0 & 4 & \frac{10}{13} & \frac{1}{13} & -\frac{3}{13} \\ 0 & 0 & 13 & -26 & -12 & 4 & 1 \end{array}\right)$$

Přímé metody - Přímá Gaussova-Jordanova eliminace - příklad

$$\left(\begin{array}{ccc|c|ccc} 1 & 0 & -2 & 7 & 3 & -1 & 0 \\ 0 & 1 & 0 & 4 & \frac{10}{13} & \frac{1}{13} & -\frac{3}{13} \\ 0 & 0 & 13 & -26 & -12 & 4 & 1 \end{array}\right)$$

$$\xrightarrow{R_2 + \frac{2}{13} R_3}$$

$$\left(\begin{array}{ccc|c|ccc} 1 & 0 & 0 & 3 & \frac{15}{13} & -\frac{5}{13} & \frac{2}{13} \\ 0 & 1 & 0 & 4 & \frac{10}{13} & \frac{1}{13} & -\frac{3}{13} \\ 0 & 0 & 13 & -26 & -12 & 4 & 1 \end{array}\right)$$

Přímé metody - Přímá Gaussova-Jordanova eliminace - příklad

Matici máme v diagonálním tvaru a můžeme tedy lehce vypočíst řešení.

V obou postupech je stejný počet kroků, jelikož potřebujeme vynulovat stejný počet prvků, ale v případě operací se už oba postupy liší.

Přímé metody - Gaussova eliminace - náročnost

Mějme matici o velikosti $n \times n$. Potřebujeme vynulovat $\frac{1}{2}n(n-1)$ (tedy počet prvků pod diagonálou), každý tento prvek nulujeme pomocí maximálně n operací (protože procházíme všech n sloupců).

Složitost algoritmu je tedy $\sim n^3$ (přesněji $\approx \frac{1}{3}n^3$).

Na zpětný běh (tedy na počítání s $\triangle$ tvarem matice) potřebujeme $\sim n^2$ operací (přesněji $\approx \frac{1}{2}n^2$). Protože pro každý prvek vysčítáváme (a dělíme) maximálně přes n prvků.

Přímé metody - Gaussova-Jordanova eliminace - náročnost

Mějme matici o velikosti $n \times n$. Náročnost této metody je pořád $\sim n^3$, ale je tam větší konstanta.

Máme už diagonální matici, pro výpočet řešení už tedy stačí pouze n operací.

Přímé metody - Vývoj algoritmu

Pro lepší pochopení algoritmu Gaussovy(-Jordanovy) eliminace si ukážeme jeho vývoj

1. ukázka eliminace prvku v druhém řádku a prvním sloupci zde
2. rožšíření na libovolný řádek zde
3. eliminace prvního sloupce zde
4. eliminace k-tého sloupce (pokud $k = 1$), tedy poddiagonálních prvků zde
5. eliminace k-tého sloupce (ne nutně $k = 1$) zde
6. eliminace všech poddiagonálních prvků, tedy Gaussova eliminace zde
7. eliminace všech mimodiagonálních prvků, tedy Gaussova-Jordanova eliminace zde

Přímé metody - Zpětné běhy

Pokud máme △ matici stačí provést pouze zpětný běh

- pro horní △ matici je program zde,
- pro dolní △ matici je program zde.

Přímé metody - Pivoting

Minulou hodinu jsem mluvili o chybách a o tom, že kritická operace je odečítání. Toho se u Gaussovy-Jordanovy eliminaci dopouštíme pořád.

Je nutné přehazovat řádky a sloupce, tak abychom tuto chybu co nejvíce potlačili. Tomuto procesu se říká *pivoting*.

Bez něho jsou prakticky přímé metody nepoužitelné.

Přímé metody - Výběr pivota

Snažíme se volit největší prvek, aby byl na diagonále. Díky tomu vím, že nebudu odečítat podobné čísla.

Volí se různé strategie výběru maximálního prvku

- maximální prvek z celé neupravené submatice - náročné,
- sloupcový - v k-tém kroku hledám největší prvek v k-tém sloupci,
- řádkový - v k-tém kroku hledám největší prvek v k-tém řádku (méně časté než sloupcový).

Přímé metody - Pivoting - příklad

Mějme následující matici a vektor pravých stran

$$\mathbf{A} = \begin{pmatrix} 10^{-4} & 1.0 & 0 \\ 1.0 & 0.25 & 1.0 \\ 0 & 1.0 & 10^{-4} \end{pmatrix}, \qquad \vec{b} = \begin{pmatrix} 0.1 \\ 1.0 \\ 0.1 \end{pmatrix}.$$

Nyní budeme tuto soustavu řešit (s přesností na tři platné cifry) pomocí Gaussovy eleminace a následného zpětného běhu.

Přesné řešení $\mathbf{A}\vec{x} = \vec{b}$ je rovno (s přesností na tři platné cifry)

$$\vec{x} = \begin{pmatrix} 0.488 \\ 0.1 \\ 0.488 \end{pmatrix}$$

Přímé metody - Pivoting - příklad

$$\left(\begin{array}{ccc|c} 10^{-4} & 1.0 & 0 & 0.1 \\ 1.0 & 0.25 & 1.0 & 1.0 \\ 0 & 1.0 & 10^{-4} & 0.1 \end{array}\right)$$

$$\xrightarrow{R_2 - 10^4 R_1}$$

$$\left(\begin{array}{ccc|c} 10^{-4} & 1.0 & 0 & 0.1 \\ 0 & 9\,999.75 & 1.0 & -999 \\ 0 & 1.0 & 10^{-4} & 0.1 \end{array}\right)$$

Přímé metody - Pivoting - příklad

$$\left(\begin{array}{ccc|c} 10^{-4} & 1.0 & 0 & 0.1 \\ 0 & 9\,999.75 & 1.0 & -999 \\ 0 & 1.0 & 10^{-4} & 0.1 \end{array}\right) \xrightarrow{\text{zaokrouhelní na tři platné cifry}} \left(\begin{array}{ccc|c} 10^{-4} & 1.0 & 0 & 0.1 \\ 0 & 10^{4} & 1.0 & -999 \\ 0 & 1.0 & 10^{-4} & 0.1 \end{array}\right)$$

Přímé metody - Pivoting - příklad

$$\left(\begin{array}{ccc|c} 10^{-4} & 1.0 & 0 & 0.1 \\ 0 & -10^4 & 1.0 & -999 \\ 0 & 1.0 & 10^{-4} & 0.1 \end{array}\right)$$

$$\overset{R_3 - 10^{-4} R_2}{\longrightarrow}$$

$$\left(\begin{array}{ccc|c} 10^{-4} & 1.0 & 0 & 0.1 \\ 0 & -10^4 & 1.0 & -999 \\ 0 & 0 & 2 \cdot 10^{-4} & 10^{-4} \end{array}\right)$$

Přímé metody - Pivoting - výsledek bez pivotingu

Není těžké spočítat, že řešení $\mathbf{A}\vec{x} = \vec{b}$ pomocí zpětného běhu je rovno

$$\vec{x} = \begin{pmatrix} 0 \\ 0.1 \\ 0.5 \end{pmatrix}.$$

Vidíme, že výsledek je v první složce nulový, což absolutně nesouhlasí s naším řešením.

Přímé metody - Pivoting - příklad

Nyní prohodíme pouze první a druhý řádek matice, tak abychom měli na pozici $1, 1$ větší prvek. Máme tedy

$$\tilde{\mathbf{A}} = \begin{pmatrix} 1.0 & 0.25 & 1.0 \\ 10^{-4} & 1.0 & 0 \\ 0 & 1.0 & 10^{-4} \end{pmatrix}, \qquad \tilde{\vec{b}} = \begin{pmatrix} 1.0 \\ 0.1 \\ 0.1 \end{pmatrix}.$$

Nyní budeme tuto soustavu opět řešit (s přesností na tři platné cifry) pomocí Gaussovy eleminace a následného zpětného běhu.

Přímé metody - Pivoting - příklad

$$\left(\begin{array}{ccc|c} 1.0 & 0.25 & 1.0 & 1.0 \\ 10^{-4} & 1.0 & 0.0 & 0.1 \\ 0 & 1.0 & 10^{-4} & 0.1 \end{array}\right)$$

$$\overset{R_2 - 10^{-4} R_1}{\longrightarrow}$$

$$\left(\begin{array}{ccc|c} 1.0 & 0.25 & 1.0 & 1.0 \\ 0 & 0.999\,975 & -10^{-4} & 0.099\,9 \\ 0 & 1.0 & 10^{-4} & 0.1 \end{array}\right)$$

Přímé metody - Pivoting - příklad

$$\left(\begin{array}{ccc|c} 1.0 & 0.25 & 1.0 & 1.0 \\ 0 & 0.999\,975 & -10^{-4} & 0.099\,9 \\ 0 & 1.0 & 10^{-4} & 0.1 \end{array}\right) \xrightarrow{\text{zaokrouhelní na tři platné cifry}} \left(\begin{array}{ccc|c} 1.0 & 0.25 & 1.0 & 1.0 \\ 0 & 1.0 & -10^{-4} & 0.099\,9 \\ 0 & 1.0 & 10^{-4} & 0.1 \end{array}\right)$$

Přímé metody - Pivoting - příklad

$$\left(\begin{array}{ccc|c} 1.0 & 0.25 & 1.0 & 1.0 \\ 0.0 & 1.0 & -10^{-4} & 0.0999 \\ 0.0 & 1.0 & 10^{-4} & 0.1 \end{array}\right)$$

$$\overset{R_3 - R_2}{\longrightarrow}$$

$$\left(\begin{array}{ccc|c} 1.0 & 0.25 & 1.0 & 1.0 \\ 0.0 & 1.0 & -10^{-4} & 0.0999 \\ 0.0 & 0.0 & 2 \cdot 10^{-4} & 10^{-4} \end{array}\right)$$

Přímé metody - Pivoting - výsledek s pivotingem

Pokud opět použijeme zpětný během na spočítání řešení $\mathbf{A}\vec{x} = \vec{b}$, dostaneme

$$\vec{x} = \begin{pmatrix} 0.475 \\ 0.1 \\ 0.5 \end{pmatrix}.$$

Vidíme, že výsledek není zcela přesný, je porušena i symetrie mezi první a třetí složkou, ale i tak se jedná o velké zlepšení (při počítání na tři platné cifry).

Přímé metody - Pivoting - příklad v Matlabu

Implementaci příkladu (pro zvídavé) můžete najít zde.

Oproti řešení ukázanému tady v prezentaci se výsledek liší, to je tím, že matlab si čísla platná na tři cifry reprezentuje jinak (dle standardu **IEEE**).

LU dekompozice

LU dekompozice - Definice

Každou (silně) regulární matici $\mathbf{A}$ lze rozložit na dvě $\triangle$ matice $\mathbf{L}$ (dolní $\triangle$) a $\mathbf{U}$ (horní $\triangle$) tak, že platí

$$\mathbf{A} = \mathbf{L}\mathbf{U}.$$

Tento rozklad je nejednoznačný kvůli diagonále, neboť máme tři varianty, jak diagonálu rozdělit

1. přiřadit ji jenom k $\mathbf{L}$ a do $\mathbf{U}$ dát 1 na diagonálu
2. přiřadit ji jenom k $\mathbf{U}$ a do $\mathbf{L}$ dát 1 na diagonálu
3. rozdělit diagonálu do obou matici (odmocnina z čísla).

Často se také dělá tzv. **LDU** rozklad, kdy máme navíc kromě $\mathbf{L}$ a $\mathbf{U}$ diagonální matici $\mathbf{D}$ a platí

$$\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{U}.$$

Na diagonálách $\mathbf{L}$ a $\mathbf{U}$ jsou tedy jedničky.

LU dekompozice - Odvození

Není těžké odvodit algoritmus pro **LU rozklad** (dekompozici).

Uvažujem případ, kdy máme

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

a chceme zjistit její rozklad na matice

$$\mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{pmatrix}, \quad \mathbf{U} = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{pmatrix}.$$

LU dekompozice - Odvození

Pokud tedy vynásobíme matici $\mathbf{L}$ s $\mathbf{U}$ dostaneme rovnice

$$\begin{aligned} a_{11} &= u_{11} & a_{12} &= u_{12} \\ a_{21} &= u_{11}l_{21} & a_{22} &= u_{22} + l_{21}u_{12} \\ &\vdots & &\vdots \\ a_{k1} &= u_{11}l_{k1} & a_{k2} &= l_{k2}u_{22} + u_{12}l_{k1} & \quad \dots, \\ &\vdots & &\vdots \\ a_{n1} &= u_{11}l_{n1} & a_{n2} &= l_{n2}u_{22} + u_{12}l_{n1} \end{aligned}$$

resp. souhrně jako

$$a_{ij} = \begin{cases} u_{ij} + \sum_{k=1}^{i-1} l_{ik}u_{kj} & \text{pro } i \leq j \\ l_{ij}u_{jj} + \sum_{k=1}^{j-1} l_{ik}u_{kj} & \text{pro } i > j \end{cases}.$$

LU dekompozice - Odvození

Z těchto rovnic lze již snad vyjádřit předpisi pro prvky u_{ij} a l_{ij} jako

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \,,$$
$$l_{ij} = \frac{1}{u_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \right) \,.$$

LU dekompozice - Řešení

Pokud máte tedy **LU rozklad** matice $\mathbf{A}$, můžeme původní úlohu přepsat jako

$$\mathbf{L}\mathbf{U}\vec{x} = \vec{b},$$

to můžeme přepsat na dvě úlohy

$$\mathbf{U}\vec{x} = \vec{y}\,, \quad \mathbf{L}\vec{y} = \vec{b}\,.$$

Tedy jedná se pouze o dva zpětné běhy.

Je dobré opět použít pivoting.

Náročnost rozkladu jako Gaussova-Jordanova eliminace, pak dva zpětné běhy, ale pro různé pravé strany nemusíme přepočítávat.

LU dekompozice - Příklad

Mějme matici

$$\mathbf{A} = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}$$

Provedeme její **LU rozklad** postupně po postupně po sloupcích.

LU dekompozice - Příklad

$$\mathbf{A} = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}$$

$j = 1$

$$u_{i1} = a_{i1}\,, \quad i \leq 1\,,$$
$$l_{i1} = \frac{a_{i1}}{u_{11}}\,, \quad i \geq 1\,.$$

$$\mathbf{U} = \begin{pmatrix} 4 & ? & ? & ? \\ 0 & ? & ? & ? \\ 0 & ? & ? & ? \\ 0 & ? & ? & ? \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} 1 & ? & ? & ? \\ -\frac{1}{4} & ? & ? & ? \\ -\frac{1}{4} & ? & ? & ? \\ 0 & ? & ? & ? \end{pmatrix}$$

LU dekompozice - Příklad

$$\mathbf{A} = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}$$

$j = 2$

$$u_{i2} = a_{i2} - \sum_{k=1}^{i-1} l_{ik} u_{k2}\,, \quad i \leq 2\,,$$

$$l_{i2} = \frac{1}{u_{22}} \left(a_{i2} - \sum_{k=1}^{1} l_{ik} u_{k2} \right)\,, \quad i \geq 2\,.$$

$$\mathbf{U} = \begin{pmatrix} 4 & -1 & ? & ? \\ 0 & \frac{15}{4} & ? & ? \\ 0 & 0 & ? & ? \\ 0 & 0 & ? & ? \end{pmatrix}\,, \quad \mathbf{L} = \begin{pmatrix} 1 & 0 & ? & ? \\ -\frac{1}{4} & 1 & ? & ? \\ -\frac{1}{4} & -\frac{1}{15} & ? & ? \\ 0 & -\frac{4}{15} & ? & ? \end{pmatrix}$$

LU dekompozice - Příklad

$$\mathbf{A} = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}$$

$j = 3$

$$u_{i3} = a_{i3} - \sum_{k=1}^{i-1} l_{ik} u_{k3}\,, \quad i \leq 3\,,$$

$$l_{i3} = \frac{1}{u_{33}} \left(a_{i3} - \sum_{k=1}^{2} l_{ik} u_{k3} \right)\,, \quad i \geq 3\,.$$

$$\mathbf{U} = \begin{pmatrix} 4 & -1 & -1 & ? \\ 0 & \frac{15}{4} & -\frac{1}{4} & ? \\ 0 & 0 & \frac{56}{15} & ? \\ 0 & 0 & 0 & ? \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & ? \\ -\frac{1}{4} & 1 & 0 & ? \\ -\frac{1}{4} & -\frac{1}{15} & 1 & ? \\ 0 & -\frac{4}{15} & -\frac{2}{7} & ? \end{pmatrix}$$

LU dekompozice - Příklad

$$\mathbf{A} = \begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 4 & 0 & -1 \\ -1 & 0 & 4 & -1 \\ 0 & -1 & -1 & 4 \end{pmatrix}$$

$j = 4$

$$u_{i4} = a_{i4} - \sum_{k=1}^{i-1} l_{ik} u_{k4}\,, \quad i \leq 4\,,$$

$$l_{i4} = \frac{1}{u_{44}} \left(a_{i4} - \sum_{k=1}^{3} l_{ik} u_{k4} \right)\,, \quad i \geq 4\,.$$

$$\mathbf{U} = \begin{pmatrix} 4 & -1 & -1 & 0 \\ 0 & \frac{15}{4} & -\frac{1}{4} & -1 \\ 0 & 0 & \frac{56}{15} & -\frac{16}{15} \\ 0 & 0 & 0 & \frac{24}{7} \end{pmatrix}\,, \quad \mathbf{L} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{4} & 1 & 0 & 0 \\ -\frac{1}{4} & -\frac{1}{15} & 1 & 0 \\ 0 & -\frac{4}{15} & -\frac{2}{7} & 1 \end{pmatrix}$$

LU dekompozice - Algoritmus v Matlabu

Příslušnou implemetanci tohoto algoritmu najdete zde

Provadí stejný rozklad, tedy matice $\mathbf{L}$ má na diagonále jedničky a $\mathbf{U}$ libovolné čísla. Provede také kontrolu násobením matice $\mathbf{L}$ a $\mathbf{U}$, jestli se rovná původní matici $\mathbf{A}$.

LU dekompozice - Symetrická matice

Pokud je matice $\mathbf{A}$ symetrická, tedy platí

$$\mathbf{A} = \mathbf{A}^\intercal,$$

pak existuje tzv. **Choleskyho rozklad**

$$\mathbf{A} = \mathbf{L}\mathbf{L}^\intercal\,, \quad \text{resp.} \quad \mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^\intercal,$$

kde $\mathbf{L}$ je dolní $\triangle$ matice, resp. $\mathbf{L}$ je dolní $\triangle$ s jedničkami na diagonále a $\mathbf{D}$ je diagonální matice.

Navíc

Navíc - Programy

Mírně pokročilé programy

- Gaussova eliminace na horní $\triangle$ matici zde
- Zpětný běh pro horní $\triangle$ matici zde
- Gaussova eliminace na dolní $\triangle$ matici zde
- Zpětný běh pro dolní $\triangle$ matici zde
- Gaussova-Jordanova eliminace zde
- Zpětný běh pro diagonální matici zde
- LU dekompozice zde

s ukázkovým programem na spuštění zde.