UNIVERZITA PALACKÉHO V OLOMOUCI
PŘÍRODOVĚDECKÁ FAKULTA
KATEDRA MATEMATICKÉ ANALÝZY A APLIKACÍ MATEMATIKY

BAKALÁŘSKÁ PRÁCE

Rozklady matic a jejich použití

Vedoucí bakalářské práce:
RNDr. Jitka Machalová, Ph.D.
Rok odevzdání: 2009

Vypracovala:
Pavlína Jiříčková
MAP, III. ročník

Prohlášení

Prohlašuji, že jsem vytvořila tuto bakalářskou práci samostatně za vedení RNDr. Jitky Machalové, Ph.D. a že jsem v seznamu použité literatury uvedla všechny zdroje použité při zpracování práce.

V Olomouci dne 30.listopadu 2009

Poděkování

Ráda bych na tomto místě poděkovala vedoucí mé bakalářské práce RNDr. Jitce Machalové, Ph.D. za obětavou spolupráci a čas, který mi věnovala při konzultacích. Dále si zaslouží poděkování můj partner a kamarádi, kteří mě po celou dobu studia podporovali.

Obsah

Použité symboly

$\mathbb{C}$	množina komplexních čísel
$\mathbb{R}$	množina reálných čísel
$\mathbb{N}$	množina přirozených čísel
$\mathbb{R}^n$	n-rozměrný vektorový prostor nad reálnými čísly
$\mathbb{C}^n$	n-rozměrný vektorový prostor nad komplexními čísly
M_n	množina matic řádu n nad reálnými čísly
C_n	množina matic řádu n nad komplexními čísly
$M_{m,n}$	množina matic řádu $m \times n$ nad reálnými čísly
$C_{m,n}$	množina matic řádu $m \times n$ nad komplexními čísly
$\|a\|$	absolutní hodnota čísla a
A^T	transponovaná matice k matici A
A^{-1}	inverzní matice k matici A
$\overline{A}$	matice s komplexně sdruženými prvky
$A^* = (\overline{A})^T$	hermitovská matice k matici A
x^T	transponovaný vektor k vektoru x
x^*	hermitovský vektor k vektoru x
I_n	jednotková matice řádu n
$h(A)$	hodnost matice A
$det(A)$	determinant matice A
$A = (a_{ij})$	matice s prvky a_{ij}
δ_{jk}	Kroneckerovo delta

Úvod

Hlavním cílem této bakalářské práce je nastudovat základní typy rozkladů různých typů matic, rozklady předvést na příkladech a v matematickém programu Matlab vytvořit programy, pomocí kterých uvedené rozklady můžeme nalézt. Také si ukážeme, jak můžeme tyto rozklady matic využít např. v přímých metodách řešení systému lineárních rovnic nebo při určování pseudoinverze dané matice.

První kapitola je pojata jako přípravná a zavedeme v ní pojmy z lineární algebry a numerické matematiky, které sice nejsou tématem této bakalářské práce, ale blízce s ním souvisí.

Druhá kapitola pojednává o LU rozkladu, který lze považovat za nejznámější a vychází z tzv. Gaussovy eliminační metody. Tento rozklad je typickým příkladem součinového rozkladu, proto se jím budeme zabývat důkladněji. Ukážeme si jeho odvození, využití při řešení systému lineárních rovnic a aplikaci na hermitovské a třídiagonální matice.

Ve třetí kapitole se budeme věnovat skeletovému rozkladu. Pokusíme se přiblížit a vysvětlit postup jeho hledání na příkladech. Jeho využití můžeme nalézt jak při řešení systému lineárních rovnic tak i při určování pseudoinverzní matice.

Poslední čtvrtá kapitola se zabývá singulárním rozkladem, který se může zdát nejsložitější. Jeho využítí je opět při řešení systému lineárních rovnic a při určování pseudoinverze.

V celé práci budeme příklady a programy sestavené v matematickém softwaru Matlab, tzv. m-fily, ukončovat symbolem $\spadesuit$ a důkazy symbolem $\square$. V m-filech budeme používat určitou přesnost "eps", což bude nějaké velmi malé kladné číslo. V příkladech budeme zpravidla volit $eps = 10^{-16}$.

K této bakalářské práci patří CD, na kterém se nacházejí m-fily sestavené podle jednotlivých algoritmů nebo postupů hledání rozkladů matic, o kterých tato bakalářská práce pojednává.

1 Přípravná kapitola

V této kapitole připomeneme některé základní pojmy z lineární algebry a numerické matematiky, jejichž znalost je pro následující text nezbytná.

Definice 1.1. Matici $A \in C_n$ nazýváme **regulární**, jestliže její determinant je různý od nuly, tzn. $det(A) \neq 0$. Není-li matice A regulární, nazýváme ji **singulární**.

Definice 1.2. Nechť $A \in C_{m,n}$. **Hodností** matice A nazveme takové číslo $h(A)$, které je rovno maximálnímu počtu lineárně nezávislých řádků matice A.

Poznámka 1.1. Ekvivalentně lze vyslovit definici 1.1 také tak, že matice $A \in C_n$ je regulární, jestliže:

(i) její řádky jsou lineárně nezávislé

(ii) její sloupce jsou lineárně nezávislé

(iii) hodnost matice A je právě n

(iv) k ní existuje matice inverzní.

Uvažujme nyní systém lineárních rovnic

$$Ax = b \tag{1}$$

kde $A \in C_n$, $A = (a_{ij})$, $x \in \mathbb{C}^n$, $x = (x_1, \ldots, x_n)^T$, $b \in \mathbb{C}^n$, $b = (b_1, \ldots, b_n)^T$. Je-li matice A regulární, pak existuje právě jedno řešení systému (1), které je tvaru $x^* = A^{-1}b$. Metody, jak nalézt řešení soustav lineárních rovnic, se většinou rozdělují na dvě skupiny: metody přímé a metody iterační. V této bakalářské práci se budeme zabývat pouze metodami přímými. Jsou to metody, při jejichž aplikaci získáme po konečném počtu kroků přesné řešení x^*. To vše za předpokladu, že všechny aritmetické operace provádíme přesně a vstupní data jsou také dána přesně.

Uveďme nyní základní poznatky z lineární algebry, které souvisí s uvedeným systémem lineárních rovnic (1).

Definice 1.3. Systém lineárních rovnic (1) se nazývá **řešitelný** (resp. **neřešitelný**), jestliže existuje alespoň jedno (resp. neexistuje žádné) řešení systému lineárních rovnic (1).

Věta 1.1 (Frobenius)**.** *Systém lineárních rovnic (1) je řešitelný právě tehdy, když hodnost matice A je rovna hodnosti rozšířené matice $(A|b)$ systému lineárních rovnic (1).*

Důkaz: viz. [5], strana 38.

Definice 1.4. Matice $R \in C_n$, $R = (r_{ij})$ se nazývá **horní trojúhelníková** matice, jestliže $r_{ij} = 0$ pro $i > j$, $i, j = 1, \ldots, n$. Matice $S \in C_n$, $S = (s_{ij})$ se nazývá **dolní trojúhelníková** matice, jestliže $s_{ij} = 0$ pro $i < j$, $i, j = 1, \ldots, n$.

Příklad 1.1. Uveďme si příklad horní a dolní trojúhelníkové matice

$$R = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \qquad S = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 0 & 0 & 0 \\ 1 & 2 & 3 & 4 & 0 & 0 \\ 1 & 2 & 3 & 4 & 5 & 0 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix}$$

Definice 1.5. Řekneme, že matice $R \in C_n$, $R = (r_{ij})$ je **(p,q)-pásová matice**, jestliže existují $p, q \in \mathbb{N}$ taková, že $1 < p$, $q < n$ a $r_{ij} = 0$ pro $i + p \leq j$ nebo $j + q \leq i$. **Šířkou pásu** budeme rozumět číslo $w = p + q - 1$.

Příklad 1.2. Uveďme si příklad (2,4)-pásové matice

$$R = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 0 & 0 \\ 6 & 2 & 5 & 8 & 4 & 6 & 0 \\ 5 & 10 & 1 & 3 & 3 & 8 & 3 \\ 0 & 6 & 23 & 1 & 2 & 3 & 7 \\ 0 & 0 & 12 & 11 & 1 & 2 & 14 \\ 0 & 0 & 0 & 4 & 7 & 1 & 9 \end{pmatrix}$$

Poznámka 1.2. Ve speciálním případě, kdy $p = q = 2$ se pásová matice $R = (r_{ij})$ nazývá **třídiagonální** a je ve tvaru

$$R = \begin{pmatrix} r_{11} & r_{12} & 0 & \dots & \dots & 0 \\ r_{21} & r_{22} & r_{23} & \dots & \dots & 0 \\ 0 & r_{32} & r_{33} & & & \vdots \\ \vdots & & & \ddots & & 0 \\ \vdots & & & & r_{n-1,n-1} & r_{n-1,n} \\ 0 & \dots & \dots & 0 & r_{n,n-1} & r_{nn} \end{pmatrix}$$

Definice 1.6. Matice $A \in C_n$, $A = (a_{ij})$ se nazývá

(i) **řádkově diagonálně dominantní**, jestliže

$$|a_{ii}| \geq \sum_{\substack{i \neq j \\ j=1}}^{n} |a_{ij}| \qquad \forall i = 1, \dots, n$$

(ii) **ryze řádkově diagonálně dominantní**, jestliže

$$|a_{ii}| > \sum_{\substack{i \neq j \\ j=1}}^{n} |a_{ij}| \qquad \forall i = 1, \dots, n$$

Věta 1.2. *Je-li matice $A \in C_n$ ryze řádkově diagonálně dominantní, pak je regulární.*

Důkaz: viz. [4], strana 94.

Definice 1.7. Nechť $A \in C_n, A = (a_{ij})$. Pod pojmem **hermitovská matice** rozumíme takovou komplexní matici, pro kterou platí

$$A = A^*$$

Poznámka 1.3. Uvědomme si, že každá hermitovská matice A má všechna vlastní čísla reálná, i když její prvky jsou komplexní, o čemž se můžeme dočíst např. v literatuře [1].

Poznámka 1.4. Je-li matice $A \in M_n$, tak se místo pojmu hermitovská matice užívá pojem symetrická matice.

Definice 1.8. Hermitovská matice $A \in C_n$ se nazývá **pozitivně definitní**, jestliže $x^*Ax > 0$ pro každý nenulový vektor $x \in \mathbb{C}^n$. Hermitovská matice $A \in C_n$ se nazývá **pozitivně semidefinitní**, jestliže $x^*Ax \geq 0$ pro každý nenulový vektor $x \in \mathbb{C}^n$.

Věta 1.3. *Matice $A \in C_n, A = (a_{ij})$ je pozitivně definitní právě tehdy, když má všechny hlavní minory kladné, tedy když platí*

$$a_{11} > 0, \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0, \ldots, det(A) > 0$$

Důkaz: viz. [2], strana 56.

Věta 1.4. *Je-li matice $A \in C_n$ pozitivně definitní, pak je regulární.*

Důkaz: viz. [4], strana 95.

Definice 1.9. Čtvercovou matici $A \in C_n$ nazýváme **silně regulární** právě tehdy, když má všechny hlavní minory nenulové, tedy když platí

$$a_{11} \neq 0, \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \neq 0, \ldots, det(A) \neq 0$$

Definice 1.10. Elementární řádkovou úpravou matic rozumíme

(i) vzájemnou výměnu libovolných řádků

(ii) vynásobení libovolného řádku nenulovou konstantou

(iii) přičtení lineární kombinace ostatních řádků (speciálně přičtení násobku jiného řádku) k některému řádku

(iv) vynechání řádku, který je lineární kombinací ostatních řádků (speciálně vynechání nulového řádku).

Poznámka 1.5. Všechny čtyři úpravy uvedené v definici 1.10 lze provádět i na sloupcích matice a platí, že ekvivalentní úpravy prováděné na matici zachovávají hodnost dané matice.

Definice 1.11. Nechť je dána matice $A \in C_{m,n}$. Platí-li pro skalár $\lambda \in \mathbb{C}$ a nenulový vektor $x \in \mathbb{C}^n$

$$Ax = \lambda x$$

řekneme, že λ je **vlastní hodnota matice** A a x **vlastní vektor matice** A **příslušný vlastní hodnotě** λ .

Věta 1.5. *Symetrická matice A je pozitivně semidefinitní právě tehdy, když všechny její hlavní minory jsou nezáporné.*

Důkaz: viz [5], strana 95.

Definice 1.12. Nechť $A \in C_{m,n}, A = (a_{ij})$. Řekneme, že prvek a_{ij} je **vedoucí prvek** $i - t\acute{e}ho$ řádku matice A, pokud $a_{ij} \neq 0$ a zároveň platí, že $a_{ik} = 0$ pro všechny $1 \leq k < j$.

Poznámka 1.6. Definici 1.12 lze rozumět také tak, že vedoucím prvkem $i - t\acute{e}ho$ řádku je jeho první nenulový prvek.

Definice 1.13. Řekneme, že matice $A \in C_n, A = (a_{ij})$ je v **redukovaném stupňovitém tvaru**, pokud splňuje následující čtyři podmínky

(i) každý nenulový řádek matice A leží nad každým jejím nulovým řádkem

(ii) vedoucí prvek daného řádku leží více vlevo než vedoucí prvek řádku pod ním

(iii) vedoucí prvek každého nenulového řádku je 1

(iv) ve sloupci, ve kterém se nachází vedoucí prvek některého řádku, jsou všechny ostatní prvky rovny nule.

Poznámka 1.7. Jestliže matice A splňuje pouze první dvě podmínky z předchozí definice 1.13 řekneme, že je ve **stupňovitém tvaru**. Ukažme si nyní příklad matic, které

(i) jsou ve stupňovitém tvaru, ale nejsou v redukovaném stupňovitém tvaru

$$\begin{pmatrix} 2 & 3 & 0 & 1 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

(ii) jsou v redukovaném stupňovitém tvaru

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Definice 1.14. Vektorová norma $\|.\|$ je funkce $\|.\| : \mathbb{C}^n \longrightarrow \mathbb{R}$ taková, že platí:

(i) $\|x\| \geq 0 \qquad \forall x \in \mathbb{C}^n$

(ii) $\|x\| = 0 <=> x = o$, kde o je nulový vektor

(iii) $\|cx\| = |c|\|x\| \qquad \forall x \in \mathbb{C}^n, \forall c \in \mathbb{C}$

(iv) $\|x+y\| \leq \|x\| + \|y\| \qquad \forall x, y \in \mathbb{C}^n$

Definice 1.15. **Ortogonální maticí** budeme rozumět takovou matici $A \in M_n$, pro kterou platí $A^{-1} = A^T$. **Unitární maticí** budeme rozumět takovou matici $B \in C_n$, pro kterou platí $B^{-1} = B^*$.

Definice 1.16. Nechť $A \in C_{m,n}$, $X \in C_{n,m}$. Matice X se nazývá **Moore - Penroseova inverze** matice A, jestliže platí

(i) $AXA = A$ (iii) $(AX)^* = AX$

(ii) $XAX = X$ (iv) $(XA)^* = XA$

Poznámka 1.8.

(i) Podmínky (i) - (iv) z definice 1.16 se nazývají Moore - Penroseovy podmínky. Matice X se stručně nazývá MP inverze. Často mluvíme také jen o pseudoinverzi nebo též o pseudoinverzní matici.

(ii) Pseudoinverzní matici k matici $A \in C_{m,n}$ zpravidla označujeme symbolem A^+. Jestliže $h(A) = m$, pak pro výpočet pseudoinverze matice A platí, že $A^+ = A^*(AA^*)^{-1}$ a jestliže $h(A) = n$, pak pro výpočet pseudoinverze matice A platí, že $A^+ = (A^*A)^{-1}A^*$, což můžeme najít v literatuře [1], str. 211.

2 LU rozklad

V této kapitole se budeme zabývat rozkladem matice, který se nazývá *LU* rozklad. Ukážeme si jeho využití a aplikujeme jej na matice v určitém tvaru.

Nejdříve si odvodíme Gaussovu eliminační metodu. Jde o metodu exaktního řešení soustav lineárních algebraických rovnic, ze které *LU* rozklad vychází. Tento rozklad se využívá hlavně při řešení více soustav lineárních rovnic, které mají různé pravé strany, ale stejnou matici. Najdeme-li totiž *LU* rozklad dané matice, počítání se zjednoduší, jelikož dále je možné pracovat pouze s trojúhelníkovými maticemi. Narazíme také na problémy, které mohou být odstraněny pomocí řádkových elementárních úprav. V této sekci si také ukážeme využití *LU* rozkladu při výpočtu inverzní matice a determinantu.

Dovolila bych si v této bakalářské práci uvést odvození Gaussovy eliminační metody v podobě, která je uvedena ve skriptu [4], jelikož je zde podle mého názoru uvedeno srozumitelně.

LU rozklad je příkladem součinového rozkladu a vychází z jedné z nejznámějších a nejzákladnějších přímých metod pro řešení systému lineárních rovnic. Jde o již zmíněnou tzv. **Gaussovu eliminační metodu** označovanou zkratkou "GEM". Princip GEM je postaven na převedení systému lineárních rovnic (1) na tzv. redukovaný systém $Ux = c$, kde $U = (u_{ij})$ je horní trojúhelníková matice a $c = (c_1, \dots, c_n)^T$ je vektor pravé strany. Tomuto převodu, při němž používáme vhodné elementární úpravy, říkáme **přímý chod**. Řešení redukovaného systému je totožné s řešením původního systému a lze jej jednoduše získat pomocí tzv. **zpětného chodu** (za předpokladu $u_{ii} \neq 0, i = 1, 2, \dots, n$):

$$x_i = \frac{c_i - \sum_{k=i+1}^{n} u_{ik} x_k}{u_{ii}}, \qquad i = n, n-1, \dots, 1$$

Nyní si názorně vysvětlíme princip GEM. Nechť tedy máme k dispozici matici

$$(A|b) = \left(\begin{array}{ccc|c} a_{11} & \dots & a_{1n} & b_1 \\ a_{21} & \dots & a_{2n} & b_2 \\ \vdots & & \vdots & \vdots \\ a_{n1} & \dots & a_{nn} & b_n \end{array}\right)$$

Aplikací Gaussovy eliminační metody na matici $(A|b)$ dostáváme následující posloupnost matic

$$(A|b) = \left(\begin{array}{ccc|c} a_{11}^{(0)} & \dots & a_{1n}^{(0)} & b_1^{(0)} \\ a_{21}^{(0)} & \dots & a_{2n}^{(0)} & b_2^{(0)} \\ \vdots & & \vdots & \vdots \\ a_{n1}^{(0)} & \dots & a_{nn}^{(0)} & b_n^{(0)} \end{array}\right) \sim \left(\begin{array}{cccc|c} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1n}^{(1)} & b_1^{(1)} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} & b_2^{(1)} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} & b_n^{(2)} \end{array}\right) \sim \dots$$

$$\dots \sim \left(\begin{array}{ccccc|c} a_{11}^{(n-1)} & a_{12}^{(n-1)} & \dots & \dots & a_{1n}^{(n-1)} & b_1^{(n-1)} \\ 0 & a_{22}^{(n-1)} & \dots & \dots & a_{2n}^{(n-1)} & b_2^{(n-1)} \\ 0 & 0 & a_{33}^{(n-1)} & \dots & a_{3n}^{(n-1)} & b_3^{(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & 0 & \dots & a_{nn}^{(n-1)} & b_n^{(n-1)} \end{array}\right) = \left(\begin{array}{cccc|c} u_{11} & \dots & \dots & u_{1n} & c_1 \\ 0 & u_{22} & \dots & u_{2n} & c_2 \\ \vdots & & \ddots & \vdots & \\ 0 & \dots & 0 & u_{nn} & c_n \end{array}\right) = (U|c)$$

kde $A^{(0)} = (a_{ij}^{(0)})$, $A^{(1)} = (a_{ij}^{(1)})$, $\dots$, $A^{(n-1)} = (a_{ij}^{(n-1)}) = (u_{ij}) = U$.

Použijeme-li stručnější zápis vypadá získaná posloupnost matic následovně

$$(A|b) = (A^{(0)}, b^{(0)}) \sim (A^{(1)}, b^{(1)}) \sim \dots \sim (A^{(n-1)}, b^{(n-1)}) = (U|c)$$

Pro lepší porozumění, si předvedeme $k - t\acute{y}$ krok při řešení systému lineárních rovnic pomocí Gaussovy eliminační metody. Předpokládejme tedy, že $k > 1$.
V $k - t\acute{e}m$ kroku již máme k dispozici matici

$$(A^{(k-1)}|b^{(k-1)}) = \left(\begin{array}{cccccc|c} a_{11}^{(k-1)} & a_{12}^{(k-1)} & \dots & & \dots & \dots\ a_{1n}^{(k-1)} & b_1^{(k-1)} \\ 0 & \ddots & & & & \vdots & \vdots \\ \vdots & & a_{k-1,k-1}^{(k-1)} & & & a_{k-1,n}^{(k-1)} & b_{k-1}^{(k-1)} \\ & & 0 & a_{kk}^{(k-1)} & & \vdots & \vdots \\ & & 0 & a_{k+1,k}^{(k-1)} & & & \\ \vdots & & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & & 0 & a_{n,k}^{(k-1)} & \dots & a_{nn}^{(k-1)} & b_n^{(k-1)} \end{array}\right)$$

Postupujeme tedy tak, že vhodný násobek $k-té$ rovnice postupně odečteme od zbývající $r-té$ rovnice, pro $r=k+1,\dots,n$ a to proto, že chceme z $r-té$ rovnice vyeliminovat proměnnou x_k, což znamená, že tato proměnná zůstane ve všech rovnicích od první až do $k-té$ a v ostatních jsou její koeficienty rovny nule. Tento postup je možný uskutečnit pouze je-li $a_{kk}^{(k-1)} \neq 0$. Pokud tento předpoklad není splněn, pak se provádí vhodná výměna rovnic, tj. nalezneme jeden prvek $a_{ik}^{(k-1)} \neq 0$, pro $i \in \{k+1,\dots,n\}$ a poté vyměníme $k-tý$ a $i-tý$ řádek. Matici s takto vyměněnými řádky označíme $(\bar{A}|\bar{b})$, kde $\bar{A}=(\bar{a}_{ij})$.

Pro $i=k+1\dots,n$ odečteme

$$l_{ik}=\frac{\bar{a}_{ik}}{\bar{a}_{kk}} \tag{2}$$

násobek $k-tého$ řádku od $i-tého$ řádku matice $(\bar{A}|\bar{b})$. Výsledkem je matice $(A^{(k)}|b^{(k)})$, která je tvaru

$$(A^{(k)}|b^{(k)})=\left(\begin{array}{ccccc|c}
a_{11}^{(k)} & a_{12}^{(k)} \dots & \dots & & \dots\ a_{1n}^{(k)} & b_1^{(k)} \\
0 & \ddots & & & \vdots & \vdots \\
\vdots & & a_{k,k}^{(k)} & & a_{k,n}^{(k)} & b_k^{(k)} \\
 & & 0 & a_{k+1,k+1}^{(k)} & \vdots & \vdots \\
 & & 0 & a_{k+2,k+1}^{(k)} & & \\
\vdots & & \vdots & \vdots & \ddots\ \vdots & \vdots \\
0 & & 0 & a_{n,k+1}^{(k)} & \dots\ a_{nn}^{(k)} & b_n^{(k)}
\end{array}\right)$$

Čísla l_{ik} se nazývají **multiplikátory**. Prvek $a_{ik}^{(k-1)}$ se nazývá **hlavní prvek**, někdy také **pivot**.

Celý tento postup můžeme zapsat pomocí maticového násobení

$$(A^{(k)}|b^{(k)})=G_kP_k(A^{(k-1)}|b^{(k-1)}) \tag{3}$$

Pro $k=1,2,\dots,n-1$ lze psát

$$(U,c)=G_{n-1}P_{n-1}\dots G_1P_1(A|b) \tag{4}$$

kde $P_k = (p_{ij})$ je **permutační matice** (permutační matice je matice, která je sestavená z jedniček a nul, kde v každém řádku a v každém sloupci se vyskytuje právě jedna jednička) taková, že platí $p_{ii} = p_{kk} = 0$, $p_{ik} = p_{ki} = 1$, $p_{jj} = 1$ pro $j \neq i, j \neq k$ a zbytek jsou nuly. Tedy matice P_k vznikla z jednotkové matice vyměnou $k - t\acute{e}ho$ a $i - t\acute{e}ho$ řádku. Matice G_k je dolní trojúhelníková matice tvaru

$$G_k = \begin{pmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & 1 & & \\ & & -l_{k+1,k} & & \\ & & \vdots & \ddots & \\ 0 & & -l_{nk} & & 1 \end{pmatrix}$$

Matice G_k se nazývá **Frobeniova matice**. Matice P_k a G_k jsou regulární, což plyne z tvaru obou matic a navíc pro ně platí vztahy

$$P_k^{-1} = P_k, \qquad G_k^{-1} = \begin{pmatrix} 1 & & & & 0 \\ & \ddots & & & \\ & & 1 & & \\ & & l_{k+1,k} & & \\ & & \vdots & \ddots & \\ 0 & & l_{nk} & & 1 \end{pmatrix}$$

Jak už jsme zmínili, tak Gaussova eliminační metoda je jednou z přímých metod řešení systému linárních rovnic. My ji v této bakalářské práci budeme využívat především k převedení dané matice na trojúhelníkový tvar. Tudíž s vektorem pravé strany nebudeme ve většině příkladů pracovat.

Následující věta je jedním z důležitých výsledků Gaussovy eliminační metody a je převzata z literatury [4]. Značení je v této větě přizpůsobeno vzhledem k výše uvedené teorii.

Věta 2.1 (O trojúhelníkovém rozkladu)**.** *Jestliže Gaussova eliminační metoda pro matici $A \in C_n$ lze provést bez výměny řádků, pak matici A lze rozložit na součin dolní a horní trojúhelníkové matice*

$$A = LU$$

kde matice $U = (u_{ij})$, $L = (l_{ij})$ jsou definovány takto:

$$u_{ij} = \begin{cases} a_{ij}^{(i-1)} & i = 1, \dots, j \\ 0 & i = j+1, j+2, \dots, n \end{cases}$$

a

$$l_{ij} = \begin{cases} 0 & i = 1, 2, \dots, j-1; j \geq 2 \\ 1 & i = j; j = 1, \dots, n \\ \frac{a_{ij}^{(j-1)}}{a_{jj}^{(j-1)}} & i = j+1, \dots, n \end{cases}$$

kde l_{ij} jsou příslušné multiplikátory z Gaussovy eliminační metody.

Důkaz: viz. [4], strana 99.

Poznámka 2.1. Jelikož ve větě 2.1 jsou permutační matice P_k rovny jednotkovým maticím, protože GEM šla provést bez výměny řádků, tak ze vztahu (4) plyne, že $L = G_1^{-1} G_2^{-1} \dots G_{n-1}^{-1}$, kde G_i jsou Frobeniovy matice.

Poznámka 2.2. Jestliže existují matice L a U definované v předchozí větě 2.1, pak rozklad matice $A = LU$ se nazývá **trojúhelníkový rozklad** matice A, nebo-li **LU rozklad**.

Věta 2.2.

(i) *Jestliže matice A je ryze řádkově diagonálně dominantní, pak lze Gaussovu eliminační metodu provést bez výměny řádků (z čehož plyne, že umíme najít LU rozklad matice A).*

(ii) Jestliže matice A je pozitivně definitní, pak lze Gaussovu eliminační metodu provést bez výměny řádků (z čehož plyne, že umíme najít LU rozklad matice A).

Důkaz: Vzhledem k větám 1.2 a 1.4 je tvrzení zřejmé.

Věta 2.3. *Nechť matice $A \in C_n, A = (a_{ij})$ je silně regulární, tzn. nechť všechny hlavní minory matice A jsou různé od nuly, tedy*

$$a_{11} \neq 0, \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \neq 0, \ldots, det(A) \ \neq 0$$

Pak matici A lze rozložit na součin dolní a horní trojúhelníkové matice.

Důkaz: viz. [4], strana 106.

Příklad 2.1. Najděte LU rozklad matice A pomocí Gaussovy eliminační metody

$$A = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 3 & -2 \\ 3 & -1 & 7 \end{pmatrix}$$

K nalezení LU rozkladu matice A pomocí Gaussovy eliminační metody budeme postupovat podle věty 2.1. V prvním kroku počítáme multiplikátory l_{21} a l_{31}

$$l_{21} = \tfrac{a_{21}}{a_{11}} = \tfrac{-1}{1} = -1$$
$$l_{31} = \tfrac{a_{31}}{a_{11}} = \tfrac{3}{1} = 3$$

a postupně odečteme l_{21} násobek resp. l_{31} násobek prvního řádku matice A od druhého resp. třetího řádku matice A. Tím dostáváme matici

$$A^{(1)} = \begin{pmatrix} 1 & -1 & 1 \\ 0 & 2 & -1 \\ 0 & 2 & 4 \end{pmatrix}$$

a Frobeniovu matici G_1

$$G_1 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix}$$

V druhém kroku počítáme multiplikátor l_{32} matice $A^{(1)} = (a_{ij}^{(1)})$

$$l_{32} = \tfrac{a_{32}^{(1)}}{a_{22}^{(1)}} = \tfrac{2}{2} = 1$$

a odečteme l_{32} násobek druhého řádku matice $A^{(1)}$ od třetího řádku matice $A^{(1)}$. Tedy dostáváme matici

$$A^{(2)} = \begin{pmatrix} 1 & -1 & 1 \\ 0 & 2 & -1 \\ 0 & 0 & 5 \end{pmatrix} = U$$

a Frobeniovu matici G_2

$$G_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

Vzhledem k tomu, že během Gaussovy eliminační metody nebylo zapotřebí vyměnit řádky, je permutační matice v tomto případě rovna matici jednotkové. Z tohoto důvodu ji zde nemusíme uvažovat a tedy z věty 2.1 a poznámky 2.1 plyne, že matici L dostaneme následujícím násobením $L = G_1^{-1} G_2^{-1}$. Tedy

$$G_1^{-1} G_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 3 & 1 & 1 \end{pmatrix} = L$$

Vynásobením matic L a U ověříme, zda jsme počítali správně

$$LU = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 3 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & 1 \\ 0 & 2 & -1 \\ 0 & 0 & 5 \end{pmatrix} = \begin{pmatrix} 1 & -1 & 1 \\ -1 & 3 & -2 \\ 3 & -1 & 7 \end{pmatrix} = A$$

♠

Poznámka 2.3. Rozklad matice A definovaný ve větě 2.1 je jednoznačný. Pokud bychom nepožadovali, aby matice L měla na diagonále samé jedničky, pak by již takovýto rozklad nebyl jednoznačný.

Ukážeme si nyní příklad LU rozkladu zadané matice, ve kterém nepožadujeme, aby na diagonále matice L byly jedničky.

Příklad 2.2. Nalezněte LU rozklad matice

$$A = \begin{pmatrix} 4 & 2 & 2 \\ 6 & 1 & 1 \\ 2 & -1 & 1 \end{pmatrix}$$

Matici A upravujeme pomocí řádkových elementárních úprav. V prvním kroku dělíme první řádek matice A vhodným dělitelem a zapíšeme ho na pozici 11 do matice L. Dále vypočítáme multiplikátory

$l_{21} = \frac{6}{2} = 3$
$l_{31} = \frac{2}{2} = 1$

a odečteme l_{21} násobek prvního řádku od druhého řádku a l_{31} násobek prvního řádku od třetího řádku. Tedy

$$A = \begin{pmatrix} 4 & 2 & 2 \\ 6 & 1 & 1 \\ 2 & -1 & 1 \end{pmatrix} \sim \begin{pmatrix} 2 & 1 & 1 \\ 6 & 1 & 1 \\ 2 & -1 & 1 \end{pmatrix} \sim \begin{pmatrix} 2 & 1 & 1 \\ 0 & -2 & -2 \\ 0 & -2 & 0 \end{pmatrix}$$

Nyní vydělíme druhý řádek vhodným dělitelem a zapíšeme jej do matice L na pozici 22. Touto úpravou dostáváme matici

$$\begin{pmatrix} 2 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & -2 & 0 \end{pmatrix}$$

Vypočteme multiplikátor

$l_{32} = \frac{-2}{1} = -2$

a odečteme l_{32} násobek druhého řádku od třetího řádku. Dostáváme matici

$$\begin{pmatrix} 2 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix}$$

Poslední řádek není nutné ničím dělit proto na pozici 33 matice L zapíšeme jedničku a matice L vypadá tedy takto

$$L = \begin{pmatrix} 2 & 0 & 0 \\ 3 & -2 & 0 \\ 1 & -2 & 1 \end{pmatrix}$$

Vidíme tedy, že na diagonále matice L nejsou prvky rovny jedné.

Snadno ověříme správnost počítání

$$LU = \begin{pmatrix} 2 & 0 & 0 \\ 3 & -2 & 0 \\ 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix} = A$$

♠

Nyní si na stejné matici z příkladu 2.2 ukážeme aplikaci věty 2.1, tedy požadujeme, aby na diagonále matice L byly jedničky.

Příklad 2.3. Pomocí GEM nalezněte LU rozklad matice

$$A = \begin{pmatrix} 4 & 2 & 2 \\ 6 & 1 & 1 \\ 2 & -1 & 1 \end{pmatrix}$$

K nalezení LU rozkladu matice A pomocí Gaussovy eliminační metody budeme postupovat podle věty 2.1. V prvním kroku počítáme multiplikátory l_{21} a l_{31}

$l_{21} = \frac{a_{21}}{a_{11}} = \frac{6}{4} = \frac{3}{2}$
$l_{31} = \frac{a_{31}}{a_{11}} = \frac{2}{4} = \frac{1}{2}$

a odečteme l_{21} násobek prvního řádku matice A od druhého řádku matice A a l_{31} násobek prvního řádku matice A od třetího řádku matice A. Tím dostáváme matici

$$A^{(1)} = \begin{pmatrix} 4 & 2 & 2 \\ 0 & -2 & -2 \\ 0 & -2 & 0 \end{pmatrix}$$

a Frobeniovu matici G_1

$$G_1 = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{3}{2} & 1 & 0 \\ -\frac{1}{2} & 0 & 1 \end{pmatrix}$$

V druhém kroku počítáme multiplikátor l_{32} matice $A^{(1)} = (a_{ij}^{(1)})$

$l_{32} = \frac{a_{32}^{(1)}}{a_{22}^{(1)}} = \frac{-2}{-2} = 1$

a odečteme l_{32} násobek druhého řádku matice $A^{(1)}$ od třetího řádku matice $A^{(1)}$. Tedy dostáváme matici

$$A^{(2)} = \begin{pmatrix} 4 & 2 & 2 \\ 0 & -2 & -2 \\ 0 & 0 & 2 \end{pmatrix} = U$$

a Frobeniovu matici G_2

$$G_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

Vzhledem k tomu, že během Gaussovy eliminční metody opět nebylo nutné vyměnit řádky, je permutační matice rovna matici jednotkové. Z věty 2.1 a poznámky 2.1 tedy plyne, že matici L dostaneme násobením $L = G_1^{-1}G_2^{-1}$. Tedy

$$G_1^{-1}G_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{3}{2} & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{3}{2} & 1 & 0 \\ \frac{1}{2} & 1 & 1 \end{pmatrix} = L$$

Vynásobením matic L a U ověříme, zda jsme počítali správně

$$LU = \begin{pmatrix} 1 & 0 & 0 \\ \frac{3}{2} & 1 & 0 \\ \frac{1}{2} & 1 & 1 \end{pmatrix} \begin{pmatrix} 4 & 2 & 2 \\ 0 & -2 & -2 \\ 0 & 0 & 2 \end{pmatrix} = \begin{pmatrix} 4 & 2 & 2 \\ 6 & 1 & 1 \\ 2 & -1 & 1 \end{pmatrix} = A$$

♠

Poznámka 2.4. Na příkladech 2.2 a 2.3 vidíme, že k jedné matici jsme nalezli dva různé LU rozklady. Pokud tedy nebudeme požadovat, aby na diagonále matice L byly jedničky, pak rozklad není jednoznačný.

M-file 2.1. K nalezení LU rozkladu dané matice pomocí Gaussovy eliminační metody jsem v matematickém softwaru Matlab sestavila m-file, který nejdříve ověří, zda lze GEM provést bez výměny řádků a poté nalezne LU rozklad dané matice pomocí Gaussovy eliminační metody.

```
function [L,U]=LUGauss(A,eps)
%VSTUP:  A... ctvercova matice
%        eps... požadovaná přesnost
%VÝSTUP: L...dolní trojúhelníková matice LU rozkladu
%        U...horní trojúhelníková matice LU rozkladu
[m,n]=size(A);
if (m ~= n)
    error('zadaná matice není čtvercová')
else
    L=eye(m);

    for i=2:m
        if (abs(A(i-1,i-1)) <= eps)
            error('Gaussova eliminační metoda nelze provést bez
                   výměny řádků')
        else
            for j=1:i-1
                L(i,j)=A(i,j)/A(j,j);
                A(i,:)=A(i,:) - L(i,j)*A(j,:);
            end
        end
    end
end
U=A;
```

♠

Příklad 2.4. Pomocí Gaussovy eliminační metody nalezněte LU rozklad matice

$$A = \begin{pmatrix} 4 & 1 & -1 & 0 \\ 2 & 12 & 7 & 1 \\ 1 & 3 & 6 & -1 \\ 1 & 2 & -1 & 5 \end{pmatrix}$$

Vyřešíme tento příklad pomocí m-filu 2.1 tak, že do Matlabu zadáme příkaz

```
>> A=[4 1 -1 0;2 12 7 1;1 3 6 -1;1 2 -1 5];
>> [L,U] = LUGauss(A,10^-16)
```

Matlab nám vrátí matice L a U, které tvoří LU rozklad matice A

```
L =

    1.0000         0         0         0
    0.5000    1.0000         0         0
    0.2500    0.2391    1.0000         0
    0.2500    0.1522   -0.4244    1.0000


U =

    4.0000    1.0000   -1.0000         0
         0   11.5000    7.5000    1.0000
         0         0    4.4565   -1.2391
         0         0         0    4.3220
```

♠

Poznámka 2.5. Jestliže Gaussova eliminační metoda nelze provést bez výměny řádků, pak tato metoda definuje rozklad matice PA

$$PA = LU$$

kde $P = P_{n-1}P_{n-2}\dots P_1$. Matice P_k je permutační matice zachycující výměnu řádků v $k - tém$ kroku GEM.

Při praktickém výpočtu však nevíme předem, které řádky bude zapotřebí vyměnit, takže není možné určit součin PA a pak provést rozklad. Rozklad lze ovšem vyřešit následujícím způsobem. Předpokládejme, že je třeba při výpočtu zaměnit k-tý a j-tý řádek, $k > j$. Tuto výměnu si můžeme během výpočtů zaznamenat a pak v matici $A^{(j)}$ ze systému $(A^{(j)}|b^{(j)})$ by byly oproti současnému stavu vyměněny celé řádky $k - tý$ s $j - tým$ a v matici ${G_j}^{-1}$ by byly vyměněné jen spočítané části obou řádků, tedy sloupce $1, \dots, k-1$.

Nyní na příkladě ilustrujeme postup v situaci, kdy je nutné během Gaussovy eliminační metody zaměnit řádky. Ukážeme jak pracovat s permutační maticí a jak poté najít správný tvar matice L.

Příklad 2.5. Najděte LU rozklad matice A pomocí GEM

$$A = \begin{pmatrix} 1 & -2 & 1 & 3 \\ 1 & -2 & -1 & 5 \\ 2 & -3 & 1 & 1 \\ -1 & -3 & 2 & 1 \end{pmatrix}$$

V prvním kroku počítáme multiplikátory l_{21},l_{31} a l_{41} matice A

$l_{21} = \frac{a_{21}}{a_{11}} = \frac{1}{1} = 1$
$l_{31} = \frac{a_{31}}{a_{11}} = \frac{2}{1} = 2$
$l_{41} = \frac{a_{41}}{a_{11}} = \frac{-1}{1} = -1$

a odečteme postupně l_{21},l_{31} a l_{41} násobek prvního řádku matice A od druhého,třetího a čtvrtého řádku matice A. Dostáváme tak matici $A^{(1)}$

$$A^{(1)} = \begin{pmatrix} 1 & -2 & 1 & 3 \\ 0 & 0 & -2 & 2 \\ 0 & 1 & -1 & -5 \\ 0 & -5 & 3 & 4 \end{pmatrix}$$

Frobeniovu matici G_1

$$G_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$

a matici P_1, která je ve tvaru jednotkové matice, jelikož jsme neprovedli výměnu řádků

$$P_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Nyní vidíme, že prvek matice $A^{(1)}$ na hlavní diagonále (pozice 22) je nulový. Z toho plyne, že pro další krok Gaussovy eliminace je zapotřebí vyměnit tento řádek s jiným řádkem, který na pozici $i2$ pro $i = 3, 4$ nemá nulu. Vyměníme proto například druhý řádek s třetím. Dostáváme matici

$$A^{(1)} = \begin{pmatrix} 1 & -2 & 1 & 3 \\ 0 & 0 & -2 & 2 \\ 0 & 1 & -1 & -5 \\ 0 & -5 & 3 & 4 \end{pmatrix} \sim \begin{pmatrix} 1 & -2 & 1 & 3 \\ 0 & 1 & -1 & -5 \\ 0 & 0 & -2 & 2 \\ 0 & -5 & 3 & 4 \end{pmatrix}$$

A tato výměna se projeví také na tvaru permutační matice P_2, která bude mít oproti matici P_1 vyměněný druhý a třetí řádek. Máme-li takto vyměněno, tak je multiplikátor $l_{32} = 0$ a spočítáme jenom chybějící multiplikátor l_{42}

$$l_{42} = \frac{a_{42}^{(1)}}{a_{22}^{(1)}} = \frac{-5}{1} = -5$$

a odečteme l_{42} násobek druhého řádku matice $A^{(1)}$ od čtvrtého řádku matice $A^{(1)}$. Tedy dostáváme matici $A^{(2)}$

$$A^{(2)} = \begin{pmatrix} 1 & -2 & 1 & 3 \\ 0 & 1 & -1 & -5 \\ 0 & 0 & -2 & 2 \\ 0 & 0 & -2 & -21 \end{pmatrix}$$

Frobeniovu matici G_2

$$G_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 5 & 0 & 1 \end{pmatrix}$$

a matici P_2

$$P_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

V dalším kroku počítáme multiplikátor l_{43}

$$l_{43} = \frac{a_{43}^{(2)}}{a_{33}^{(2)}} = \frac{-2}{-2} = 1$$

a odečteme l_{43} násobek třetího řádku matice $A^{(2)}$ od čtvrtého řádku matice $A^{(2)}$. Dostáváme tak již horní trojúhelníkovou matici U

$$U = \begin{pmatrix} 1 & -2 & 1 & 3 \\ 0 & 1 & -1 & -5 \\ 0 & 0 & -2 & 2 \\ 0 & 0 & 0 & -23 \end{pmatrix}$$

Frobeniovu matici G_3

$$G_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

a matici P_3

$$P_3 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Nyní ze vztahu (4) podobným způsobem jako v příkladě 2.1 získáme matici L takto

$$L = P_3 P_2 P_1 (G_3 P_3 G_2 P_2 G_1 P_1)^{-1}$$

a tedy

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ -2 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 \\ -8 & -1 & 5 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & -5 & 1 & 1 \end{pmatrix}$$

Nyní pro kontrolu provedeme násobení

$$LU = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & -5 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & -2 & 1 & 3 \\ 0 & 1 & -1 & -5 \\ 0 & 0 & -2 & 2 \\ 0 & 0 & 0 & -23 \end{pmatrix} = \begin{pmatrix} 1 & -2 & 1 & 3 \\ 2 & -3 & 1 & 1 \\ 1 & -2 & -1 & 5 \\ -1 & -3 & 2 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -2 & 1 & 3 \\ 1 & -2 & -1 & 5 \\ 2 & -3 & 1 & 1 \\ -1 & -3 & 2 & 1 \end{pmatrix} = PA$$

Vidíme, že jsme počítali správně.

♠

M-file 2.2. V softwaru Matlab jsem naprogramovala m-file, který nalezne LU rozklad dané matice přesto, že Gaussova eliminační metoda nelze provést bez výměny řádku.

```
function [L,U,P]=LUPGauss(A,eps)
%VSTUP:  A... ctvercova matice
%        eps... požadovaná přesnost
%VÝSTUP: L...dolní trojúhelníková matice LU rozkladu
%        U...horní trojúhelníková matice LU rozkladu
%        P...permutační matice
[m,n]=size(A);
if (m ~= n)
```

```
    error('zadaná matice není čtvercová')
else
    L=eye(m);
    P=eye(m);
    i = 1;
    while i<m
        if (abs(A(i,i)) <= eps)
            for j=1:m
               if ((abs(A(i,j)) > eps) & (abs(A(j,i)) > eps)) |
                     (j == m)
                   temp = A(j,:);
                   A(j,:)=A(i,:);
                   A(i,:)=temp;
                   tempP = P(j,:);
                   P(j,:)=P(i,:);
                   P(i,:)=tempP;
                   if i<j
                       tempL = L(i,1:i-1);
                       L(i,1:i-1) = L(j,1:i-1);
                       L(j,1:i-1) = tempL;
                   elseif i>j
                       tempL = L(i,1:j-1);
                       L(i,1:j-1) = L(j,1:j-1);
                       L(j,1:j-1) = tempL;
                   end
                   break;
               end
            end
```

```
        end
        if (abs(A(i,i)) <=eps)
            i = i + 1;
            continue;
        end
        for j=i+1:m
            L(j,i)=A(j,i)/A(i,i);
            A(j,:)=A(j,:) - L(j,i)*A(i,:);
        end
        i = i + 1;
    end
end
U=A;
```

♠

Příklad 2.6. Nalezněte LU rozklad matice A

$$\begin{pmatrix} 1 & 2 & 1 & 1 \\ 3 & -1 & 2 & 1 \\ 2 & 4 & 2 & 5 \\ 1 & -1 & -2 & 1 \end{pmatrix}$$

Vyřešíme tento příklad pomocí m-filu 2.2 tak, že do Matlabu zadáme příkaz

```
>>  A=[1 2 1 1;3 -1 2 1;2 4 2 5;1 -1 -2 1];
>> [L,U,P]=LUPGauss(A,10^-16)
```

Matlab nám vrátí následující výsledek

```
L =

    1.0000         0         0         0
    3.0000    1.0000         0         0
    1.0000    0.4286    1.0000         0
    2.0000         0         0    1.0000

U =

    1.0000    2.0000    1.0000    1.0000
         0   -7.0000   -1.0000   -2.0000
         0         0   -2.5714    0.8571
         0         0         0    3.0000

P =

     1     0     0     0
     0     1     0     0
     0     0     0     1
     0     0     1     0
```

♠

Trojúhelníkovým rozkladem se zabýváme především proto, že má důležitou roli při řešení systému lineárních rovnic. Známe-li $PA = LU$, pak systém $Ax = b$ lze ihned řešit pro libovolný vektor b. Platí totiž

$$PAx = LUx = Pb$$

K nalezení řešení x^* uvažovaného systému použijeme matice L a U. Vyřešením systému $Ly = Pb$, pomocí přímého chodu (jelikož matice L je dolní trojúhelníková a systém řešíme od první rovnice) a systému $Ux = y$ pomocí zpětného chodu (jelikož matice U je horní trojúhelníková a systém řešíme od poslední rovnice), dostáváme hledané řešení x^*.

Příklad 2.7. Najděte řešení systému lineárních rovnic s maticí

$$A = \begin{pmatrix} 1 & 1 & 3 \\ 1 & 2 & 1 \\ 2 & -1 & 1 \end{pmatrix}$$

a vektorem pravé strany

$$b = \begin{pmatrix} -1 \\ 2 \\ 1 \end{pmatrix}$$

Nejprve najdeme LU rozklad matice A pomocí Gaussovy eliminační metody. Budeme-li postupovat stejně jako v příkladu 2.1, nalezneme matice L a U ve tvaru

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & -3 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & 1 & 3 \\ 0 & 1 & -2 \\ 0 & 0 & -11 \end{pmatrix}$$

a jelikož během Gaussovy eliminační metody nebyla zapotřebí výměna řádku, je matice P ve tvaru jednotkové matice. Nyní budeme postupovat tak, že nejdříve vyřešíme systém $Ly = b$ tzv. přímým chodem, tj. řešíme

$$Ly = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 2 & -3 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} -1 \\ 2 \\ 1 \end{pmatrix}$$

Z této soustavy tedy můžeme vypočítat složky vektoru y

$y_1 = -1$

$y_2 = 2 - y_1 = 2 - (-1) = 3$

$y_3 = 1 - 2y_1 + 3y_2 = 1 - 2 \cdot (-1) + 3 \cdot 3 = 12$

Tedy

$$y = \begin{pmatrix} -1 \\ 3 \\ 12 \end{pmatrix}$$

Nyní řešíme systém $Ux = y$ tzv. zpětným chodem

$$Ux = \begin{pmatrix} 1 & 1 & 3 \\ 0 & 1 & -2 \\ 0 & 0 & -11 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} -1 \\ 3 \\ 12 \end{pmatrix}$$

Z této soustavy dostáváme složky vektoru x

$x_3 = -\frac{12}{11}$

$x_2 = 3 + 2x_3 = 3 + 2 \cdot (-\frac{12}{11}) = \frac{9}{11}$

$x_1 = -1 - x_2 - 3x_3 = -1 - \frac{9}{11} - 3 \cdot (-\frac{12}{11}) = \frac{16}{11}$

Řešením soustavy $Ax = b$ je vektor

$$x = \begin{pmatrix} \frac{16}{11} \\ \frac{9}{11} \\ -\frac{12}{11} \end{pmatrix}$$

♠

M-file 2.3. K hledání řešení systému lineárních rovnic za pomoci LU rozkladu jsem v Matlabu naprogramovala m-file 'linearEq', který využívá již zmíněnou funkci 'LUPGauss' k nalezení matic L, U a P. Dále jsem vytvořila pomocné funkce 'primyChod' a 'zpetnyChod, které řeší tento systém pomocí přímého a zpětného chodu.

(i)
```
function y=primyChod(A,b)
%VSTUP:  A... ctvercova matice
%        b...vektor prave strany
%VÝSTUP: y...reseni systemu Ay=b
[m,n]=size(A);
```

```
for i=1:n-1
    for j=i+1:n
        if (abs(A(i,j)) > eps)
           error('zadaná matice není dolní trojúhelníková')
        end
    end
end

if (m ~= n)
    error('zadaná matice není čtvercová')
else
    for i=1:m
        temp=0;
        for j=1:i-1
            temp=temp+A(i,j)*y(j);
        end
        y(i)=b(i)-temp;
    end
end
```

(ii)
```
function y=zpetnyChod(A,b)
%VSTUP:  A... ctvercova matice
%        b...vektor prave strany
%VÝSTUP: y...reseni systemu Ay=b
[m,n]=size(A);

for i=2:n
    for j=1:i-1
```

```
            if (abs(A(i,j)) > eps)
               error('zadaná matice není horní trojúhelníková')
            end
        end
    end

    if (m ~= n)
        error('zadaná matice není čtvercová')
    else
        for i=m:-1:1
            temp=0;
            for j=i+1:m
                temp=temp+A(i,j)*y(j);
            end
            y(i)=(b(i)-temp)/A(i,i);
        end
    end
```

(iii)
```
function x=linearEq(A,b,eps)
%VSTUP:  A... čtvercová matice
%        b... sloupcový vektor pravé strany
%        eps... požadovaná přesnost
%VÝSTUP: x...řešení soustavy lineárních rovnic pomocí LU rozkladu
[m,n]=size(A);
[k,l]=size(b);
if (m ~= n)
    error('zadaná matice není čtvercová ')
elseif (l>1)
    error('vektor b musí být sloupcový ')
```

```
elseif (k~= n )
    error('vektor b nemá stejný počet řádků jako matice A ')
else
    [L,U,P]=LUPGauss(A,eps);
    y=primyChod(L,b);
    x=zpetnyChod(U,y);
    end
end
```

♠

Příklad 2.8. Najděte řešení systému lineárních rovnic s maticí

$$A = \begin{pmatrix} 1 & 10 & 2 & 1 \\ 1 & 3 & 5 & 2 \\ 2 & 1 & 4 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

a vektorem pravé strany

$$b = \begin{pmatrix} 1 \\ 2 \\ 1 \\ -1 \end{pmatrix}$$

Vyřešíme tento příklad pomocí m-filu 2.3 tak, že do Matlabu zadáme příkaz

```
>> A=[1 10 2 1; 1 3 5 2; 2 1 4 1; 1 0 1 1];
>> b=[1; 2; 1; -1];
>> [x]=linearEq(A,b,10^-16)
```

Matlab nám vrátí řešení x systému $Ax = b$

```
x =

   -0.8929
    0.1071
    0.9286
   -1.0357
```

♠

Nyní bych chtěla zmínit problémy, které souvisí s výběrem hlavního prvku (pivota). Jak už bylo zmíněno, v případě, že je hlavní prvek roven nule, nelze Gaussova eliminační metoda provést. Řešením tohoto problému bude, již zmíněná, výměna řádků. Provedeme-li ovšem výměnu řádků, při které bude pivot blízký nule, tak se může stát, že získané výsledky budou nepřesné. Což je způsobeno tím, že díky hlavnímu prvku, který by byl velmi malý, bychom dostali velký multiplikátor a při jehož použití by byl rozdíl mezi přesným a vypočítaným řešením obrovský. Ilustruji tento případ na známém příkladě Forsytha a Molera (viz [4]).

Příklad 2.9. Najděte LU rozklad matice

$$A = \begin{pmatrix} 0.0001 & 1 \\ 1 & 1 \end{pmatrix}$$

Multiplikátor $l_{21} = \frac{1}{10^{-4}} = 10^4$.
Tedy

$$G_1 = \begin{pmatrix} 1 & 0 \\ -10^4 & 1 \end{pmatrix}$$

$$U = \begin{pmatrix} 0.0001 & 1 \\ 0 & -10^4 \end{pmatrix} \qquad \text{a} \qquad L = \begin{pmatrix} 1 & 0 \\ 10^4 & 1 \end{pmatrix}$$

neboť $1 - 10^4 \approx -10^4$. Po provedení součinu LU zjistíme, že

$$LU = \begin{pmatrix} 0.0001 & 1 \\ 1 & 0 \end{pmatrix}$$

což je ovšem různé od původní matice A.

Tomuto chybnému výsledku se můžeme vyhnout již zmíněnou záměnou řádků. Uvažujme matici

$$P_1 A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 0.0001 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0.0001 & 1 \end{pmatrix}.$$

Po prvním kroku Gaussovy eliminační metody je $l_{21} = 10^{-4}$ a

$$G_1 = \begin{pmatrix} 1 & 0 \\ -10^{-4} & 1 \end{pmatrix}$$

tedy dostáváme

$$U = G_1 P_1 A = \begin{pmatrix} 1 & 0 \\ -10^{-4} & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 10^{-4} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & 0 \\ 10^{-4} & 1 \end{pmatrix}$$

Přičemž $1 - 10^{-4} \approx 1$. Nyní součin $LU = \begin{pmatrix} 1 & 1 \\ 0.0001 & 1 \end{pmatrix} = P_1 A$.

Tedy dospěli jsme k správnému výsledku.

♠

Poznámka 2.6. Postupu, který jsme ukázali na předchozím příkladu se říká **Gaussova eliminace s částečným výběrem pivota**. Spočívá vlastně v tom, že v $k - tém$ kroku vybereme v příslušném sloupci dané matice prvek maximální v absolutní hodnotě, tj. určíme p tak, aby

$$|a_{pk}^{(k)}| = \max_{k \le i \le n} |a_{ik}^{(k)}|$$

a vyměníme $p - tou$ a $k - tou$ rovnici.

Dalším možným postupem je **Gaussova eliminace s úplným výběrem pivota**.Tady v $k-tém$ kroku vybíráme prvek maximální v absolutní hodnotě tak, že

$$|a_{rs}^{(k)}| = \max_{i,j=1,\ldots,n} |a_{ij}^{(k)}|$$

Gaussova eliminace s úplným i částečným výběrem pivota je proveditelná pro libovolnou regulární matici. Podobné problémy jako jsme ukázali v příkladě 2.9 jsou minimalizovány úplným výběrem pivota, avšak podstatně méně pracný částečný výběr pivota bývá většinou také dostatečně uspokojivý.

Z předchozích úvah a odvození je zřejmý vztah mezi Gaussovou eliminační metodou a LU rozkladem, tedy že pomocí GEM získáme LU rozklad. Nyní si uvedeme algoritmus, odvozený z Gaussovy eliminační metody, pomocí nějž nalezneme LU rozklad dané matice bez přímého počítání Gaussovy eliminační metody. Je tedy zřejmé, že LU rozklad pomocí GEM bude stejný jako LU rozkad získaný pomocí algoritmu 2.1. Tento algoritmus je převzat z literatury [4], str.94.

Algoritmus 2.1. Nechť je dána matice $A \in C_n, A = (a_{ij})$. Položíme

$$u_{11} = a_{11}$$

Pro $i = 1, \ldots, n$:

$$l_{ii} = 1$$

Pro $i = 2, \ldots, n$:

$$l_{i1} = \frac{a_{i1}}{u_{11}} \quad , \text{kde}\, u_{11} \neq 0$$

$$l_{1i} = 0$$

Pro $r = 2, \ldots, n$:

$$u_{1r} = a_{1r}$$

$$u_{r1} = 0$$

Pro $i = 2, \ldots, r$:

$$u_{ir} = a_{ir} - \sum_{j=1}^{i-1} l_{ij} u_{jr}$$

$$u_{ri} = 0$$

Pro $i = r+1, \ldots, n$:

$$l_{ir} = \frac{1}{u_{rr}}(a_{ir} - \sum_{j=1}^{r-1} l_{ij} u_{jr}) \quad , \text{kde}\, u_{rr} \neq 0$$

$$l_{ri} = 0$$

Příklad 2.10. Pomocí algoritmu 2.1 najděte LU rozklad matice

$$A = \begin{pmatrix} 3 & 1 & -1 & 1 \\ 1 & 5 & 1 & 3 \\ -1 & 3 & 8 & -2 \\ 2 & 4 & 1 & 9 \end{pmatrix}$$

Budeme postupovat přesně podle algoritmu 2.1

$u_{11} = a_{11} = 3$
$l_{21} = \frac{a_{21}}{u_{11}} = \frac{1}{3}$
$l_{31} = \frac{a_{31}}{u_{11}} = \frac{-1}{3} = -\frac{1}{3}$
$l_{41} = \frac{a_{41}}{u_{11}} = \frac{2}{3}$
$u_{12} = a_{12} = 1$
$u_{13} = a_{13} = -1$
$u_{14} = a_{14} = 1$
$u_{22} = a_{22} - l_{21}u_{12} = 5 - \frac{1}{3} \cdot 1 = \frac{14}{3}$
$u_{23} = a_{23} - l_{21}u_{13} = 1 - (\frac{1}{3} \cdot (-1)) = \frac{4}{3}$
$u_{24} = a_{24} - l_{21}u_{14} = 3 - \frac{1}{3} \cdot 1 = \frac{8}{3}$
$l_{32} = \frac{1}{u_{22}}(a_{32} - l_{31}u_{12}) = \frac{3}{14}(3 - (-\frac{1}{3} \cdot 1)) = \frac{5}{7}$
$l_{42} = \frac{1}{u_{22}}(a_{42} - l_{41}u_{12}) = \frac{3}{14}(4 - (\frac{2}{3} \cdot 1) = \frac{5}{7}$
$u_{33} = a_{33} - (l_{31}u_{13} + l_{32}u_{23}) = 8 - (-\frac{1}{3} \cdot (-1) + \frac{5}{7} \cdot \frac{2}{3}) = \frac{141}{21}$
$u_{34} = a_{34} - (l_{31}u_{14} + l_{32}u_{24}) = -2 - (-\frac{1}{3} \cdot 1 + \frac{5}{7} \cdot \frac{8}{3}) = -\frac{25}{7}$
$l_{43} = \frac{1}{u_{33}}(a_{43} - (l_{41}u_{13} + l_{42}u_{23})) = \frac{21}{141}(1 - (\frac{2}{3} \cdot (-1) + \frac{5}{7} \cdot \frac{4}{3})) = \frac{15}{141}$
$u_{44} = a_{44} - (l_{41}u_{14} + l_{42}u_{24} + l_{43}u_{34}) = 9 - (\frac{2}{3} \cdot 1 + \frac{5}{7} \cdot \frac{8}{3} + \frac{15}{141} \cdot (-\frac{25}{7})) = \frac{2240}{329}$

Po těchto výpočtech dostáváme matice L a U

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{3} & 1 & 0 & 0 \\ -\frac{1}{3} & \frac{5}{7} & 1 & 0 \\ \frac{2}{3} & \frac{5}{7} & \frac{15}{141} & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 3 & 1 & -1 & 1 \\ 0 & \frac{14}{3} & \frac{4}{3} & \frac{8}{3} \\ 0 & 0 & \frac{141}{21} & -\frac{25}{7} \\ 0 & 0 & 0 & \frac{2240}{329} \end{pmatrix}$$

Nyní násobením ověříme, zda jsme počítali správně

$$LU = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{3} & 1 & 0 & 0 \\ -\frac{1}{3} & \frac{5}{7} & 1 & 0 \\ \frac{2}{3} & \frac{5}{7} & \frac{15}{141} & 1 \end{pmatrix} \begin{pmatrix} 3 & 1 & -1 & 1 \\ 0 & \frac{14}{3} & \frac{4}{3} & \frac{8}{3} \\ 0 & 0 & \frac{141}{21} & -\frac{25}{7} \\ 0 & 0 & 0 & \frac{2240}{329} \end{pmatrix} = \begin{pmatrix} 3 & 1 & -1 & 1 \\ 1 & 5 & 1 & 3 \\ -1 & 3 & 8 & -2 \\ 2 & 4 & 1 & 9 \end{pmatrix} = A$$

♠

M-file 2.4. Pro nalezení LU rozkladu dané matice jsem naprogramovala v softwaru Matlab program, který nalezne matice L a U pomocí algoritmu 2.1.

Funkce nejdříve ověří, zda je zadaná matice čtvercová a poté spočítá prvky matic L a U pomocí algoritmu 2.1. Funkce se zastaví a ohlásí chybu pokud LU rozklad nelze najít pomocí tohoto algoritmu, tzn. pokud již spočítané prvky na diagonále matice U jsou nulové.

```
function [L,U]=rozklad(A,eps)
%VSTUP:  A... čtvercová matice
%        eps... požadovaná přesnost
%VÝSTUP: L...dolní trojúhelníková matice
%        U...horní trojúhelníková matice
[k,l]=size(A);
if k~=l
    error('ERROR - matice A není čtvercová')
else
    n=length(A);
```

```
    for r=1:n
        for j=r:n
            temp = 0;
            for s˜= 1:r-1
                temp = temp + L(r,s) * U(s,j);
            end
            U(r,j)=A(r,j)- temp;
        end
        for i=r+1:n
            f = 0;
            for s˜= 1:r-1
                f = f + L(i,s) * U(s,r);
            end
            if (abs(U(r,r)) >= eps)
               L(i,r) = ((A(i,r) - f)/ U(r,r));
            else
                error('LU rozklad nelze najít pomocí tohoto
                       algoritmu')
            end
        end
    end
    for i=1:n
        L(i,i)=1;
    end
end
```

♠

Příklad 2.11. Nalezněte LU rozklad matice A pomocí algoritmu 2.1.

$$A = \begin{pmatrix} 1 & -3 & 1 & 4 \\ -1 & 5 & 2 & -3 \\ 2 & -2 & 6 & 3 \\ 0 & 2 & -1 & 2 \end{pmatrix}$$

Chceme-li hledat LU rozklad matice A pomocí m-filu 2.4, zadáme do Matlabu:

```
>> A=[1 -3 1 4;-1 5 2 -3;2 -2 6 3;0 2 -1 2];

>> [L,U]=rozklad(A,10^-16)
```

Matlab nám vrátí rozklad matice A

```
L =

     1     0     0     0
    -1     1     0     0
     2     2     1     0
     0     1     2     1


U =

     1    -3     1     4
     0     2     3     1
     0     0    -2    -7
     0     0     0    15
```

♠

Poznámka 2.7. Software Matlab má svůj příkaz k nalezení LU rozkladu. Tento příkaz je možné použít dvěma způsoby. Zaleží na tom jaká požadujeme výstupní data.

(i) v prvním způsobu zadání jsou výstupní data matice L a U. Tedy tento příkaz zadáváme v následujícím tvaru

$$[L, U] = lu(A)$$

Ovšem pokud je nutné během výpočtu vyměnit řádky matice A, pak Matlab vrátí matici L, která není dolní trojúhelníková, ale je zleva vynásobená permutační maticí, která zachycuje zmiňovanou výměnu řádků. Např.

```
>> A=[1 2 1 1;3 -1 2 1;2 4 2 5;1 -1 -2 1];
>> [L,U] = lu(A)

L =

    0.3333    0.5000         0    1.0000
    1.0000         0         0         0
    0.6667    1.0000         0         0
    0.3333   -0.1429    1.0000         0


U =

    3.0000   -1.0000    2.0000    1.0000
         0    4.6667    0.6667    4.3333
         0         0   -2.5714    1.2857
         0         0         0   -1.5000
```

(ii) v druhém způsobu zadání se mezi výstupními daty objevuje i permutační matice P

$$[L, U, P] = lu(A)$$

Pomocí tohoto příkazu můžeme nalézt LU rozklad matice PA. Aplikujeme tento tvar příkazu na matici A z (i)

```
>> A=[1 2 1 1;3 -1 2 1;2 4 2 5;1 -1 -2 1];
>> [L,U,P] = lu(A)
L =

    1.0000         0         0         0
    0.6667    1.0000         0         0
    0.3333   -0.1429    1.0000         0
    0.3333    0.5000         0    1.0000

U =

    3.0000   -1.0000    2.0000    1.0000
         0    4.6667    0.6667    4.3333
         0         0   -2.5714    1.2857
         0         0         0   -1.5000

P =

     0     1     0     0
     0     0     1     0
     0     0     0     1
     1     0     0     0
```

Vidíme, že matice L z (i) je rovna matici PL z (ii). Z toho plyne, že pro námi zadanou matici A, je vhodnější použít tvar příkazu z (ii).

Pokud bych měla zmínit nějaké další využití LU rozkladu, jednalo by se o výpočet determinantu a inverzní matice.

Pro inverzní matici X matice $A \in C_n$ platí $AX = I$, kde I je jednotková matice se sloupci e_i, pro $i = 1, \dots, n$. Sloupce e_i mají na $i - tém$ řádku jedničku a zbytek jsou nuly. Jsou-li x_{ij} prvky matice X, řešíme při hledání matice X n systémů lineárních rovnic se stejnou maticí A, ale pokaždé s jinou pravou stranou. Vektory pravé strany jsou postupně sloupce e_i matice I.Tedy řešíme tyto systémy

$$A\begin{pmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad A\begin{pmatrix} x_{12} \\ x_{22} \\ \vdots \\ x_{n2} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} \quad , \dots , \quad A\begin{pmatrix} x_{1n} \\ x_{2n} \\ \vdots \\ x_{nn} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

Máme-li již k dispozici LU rozklad matice A, můžeme jej využít tak, že počítáme soustavu $LUx_i = e_i$, kde x_i jsou sloupce matice X a e_i jsou sloupce jednotkové matice I. Řešení se skládá ze dvou kroků. V prvním hledáme vektor y_i přímým chodem, tedy řešíme soustavu $Ly_i = e_i$. V druhém hledáme zpětným chodem vektor x_i, který je řešením soustavy $Ux_i = y_i$.

Výpočet inverzní matice si ukážeme názorně na následujícím příkladě

Příklad 2.12. Najděte inverzní matici k matici A,

$$A = \begin{pmatrix} 1 & -3 & 1 & 4 \\ -1 & 5 & 2 & -3 \\ 2 & -2 & 6 & 3 \\ 0 & 2 & -1 & 2 \end{pmatrix}$$

jejíž LU rozklad jsme počítali v m-filu 2.4. Tedy máme již nalezené matice L a U

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 2 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{pmatrix}, U = \begin{pmatrix} 1 & -3 & 1 & 4 \\ 0 & 2 & 3 & 1 \\ 0 & 0 & -2 & -7 \\ 0 & 0 & 0 & 15 \end{pmatrix}$$

Nyní postupujeme podle uvedeného návodu. Řešíme systém $Ly_1 = e_1$

$$Ly_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 2 & 2 & 1 & 0 \\ 0 & 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} y_{11} \\ y_{21} \\ y_{31} \\ y_{41} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Z této soustavy dostáváme

$y_{11} = 1, \quad y_{21} = 1, \quad y_{31} = -4, \quad y_{41} = 7.$

Nyní řešíme soustavu $Ux_1 = y_1$

$$Ux_1 = \begin{pmatrix} 1 & -3 & 1 & 4 \\ 0 & 2 & 3 & 1 \\ 0 & 0 & -2 & -7 \\ 0 & 0 & 0 & 15 \end{pmatrix} \begin{pmatrix} x_{11} \\ x_{21} \\ x_{31} \\ x_{41} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ -4 \\ 7 \end{pmatrix}$$

Řešením této soustavy získáme první sloupec inverzní matice k matici A

$x_{11} = -\frac{25}{12}, \quad x_{21} = -\frac{17}{60}, \quad x_{31} = \frac{11}{30}, \quad x_{41} = \frac{7}{15}$

Tedy zatím

$$A^{-1} = \begin{pmatrix} -\frac{25}{12} & x_{12} & x_{13} & x_{14} \\ -\frac{17}{60} & x_{22} & x_{23} & x_{24} \\ \frac{11}{30} & x_{32} & x_{33} & x_{34} \\ \frac{7}{15} & x_{42} & x_{43} & x_{44} \end{pmatrix}$$

Budeme-li postupovat tímto naznačeným postupem, dostaneme po několika krocích celou matici A^{-1}

$$A^{-1} = \begin{pmatrix} -\frac{25}{12} & -\frac{5}{4} & \frac{11}{12} & \frac{11}{12} \\ -\frac{17}{60} & -\frac{1}{20} & \frac{7}{60} & \frac{19}{60} \\ \frac{11}{30} & \frac{3}{10} & -\frac{1}{30} & -\frac{7}{30} \\ \frac{7}{15} & \frac{1}{5} & -\frac{2}{15} & \frac{1}{15} \end{pmatrix} = \frac{1}{60} \begin{pmatrix} -125 & -75 & 55 & 55 \\ -17 & -3 & 7 & 19 \\ 22 & 18 & -2 & -14 \\ 28 & 12 & -8 & 4 \end{pmatrix}$$

♠

M-file 2.5. Pro výpočet inverzní matice pomocí LU rozkladu jsem v softwaru Matlab naprogramovala m-file, který nejdříve ověří, zda je zadaná matice čtvercová, poté k nalezení LU rozkladu zavolá funkci 'LUPGauss' a dále využije také funkce 'primyChod' a 'zpetnyChod'. Na závěr tento m-file vytvoří inverzní matici.

```
function A=inverze(B,eps)
%VSTUP:  B... čtvercová matice
%        eps... požadovaná přesnost
%VÝSTUP: A...inverzni matice k matici B
[m,n]=size(B);
if m~=n
    error('ERROR - matice A není čtvercová')
else
    [L,U,P]=LUPGauss(B,eps);
    I=eye(m);
    for i=1:m
        y=primyChod(L,I(:,i));
        x=zpetnyChod(U,y);
        A(:,i) = x;
    end
end
```

♠

Příklad 2.13. Najděte inverzní matici k matici B

$$B = \begin{pmatrix} 1 & 2 & -1 \\ 2 & 3 & 1 \\ 1 & -1 & 1 \end{pmatrix}$$

Vyřešíme tento příklad pomocí m-filu 2.5 tak, že do Matlabu zadáme příkaz

```
>> B=[1 2 -1;2 3 1;1 -1 1];
>> [A]=inverze(B,10^-16)
```

Matlab nám vrátí inverzní matici A k matici B

```
A =

    0.5714   -0.1429    0.7143
   -0.1429    0.2857   -0.4286
   -0.7143    0.4286   -0.1429
```

♠

Nyní se budeme věnovat otázce, jak pomocí LU rozkladu dané matice A, vypočítat determinant matice A. Jak už víme, čtvercovou matici A lze rozložit na součin dolní trojúhelníkové matice L, která má na diagonále jedničky, a horní trojúhelníkové matice U. Odtud plyne vztah pro výpočet determinantu matice A pomocí LU rozkladu. Mohou nastat dva případy. Jestliže

(i) počítáme GEM bez výběru pivota, pak platí

$$det A = det L \cdot det U = \prod_{i=1}^{n} u_{ii}$$

(ii) počítáme GEM s částečným výběrem pivota, pak platí

$$det PA = det P \cdot det A = (-1)^r \prod_{i=1}^{n} u_{ii}$$

kde r je počet výměn řádků během výpočtu GEM.

Příklad 2.14. Určete determinant matice A.

$$A = \begin{pmatrix} 1 & -3 & 1 & 4 \\ -1 & 5 & 2 & -3 \\ 2 & -2 & 6 & 3 \\ 0 & 2 & -1 & 2 \end{pmatrix}$$

Odkážeme-li se na příklad 2.11, kde jsme již našli rozklad matice $A = LU$, můžeme vynásobením prvků na diagonále matice U zjistit, že

$$detA = \prod_{i=1}^{4} u_{ii} = 1 \cdot 2 \cdot (-2) \cdot 15 = -60$$

♠

M-file 2.6. Pro výpočet determinantu zadané matice jsem v programu Matlab sestrojila m-file, který vypočte determinant zadané matice pomocí LU rozkladu.

```
function [detA] = determinant(A,eps)
%VSTUP:  A... ctvercova matice
%        eps... požadovaná přesnost
%VÝSTUP: detA...determinant matice A
[L,U,P]=LUPGauss(A,eps);
[k]=size(U);
detA=1;
for i=1:k
    detA=detA * U(i,i);
end
```

♠

Příklad 2.15. Pomocí LU rozkladu vypočtěte determinant matice A

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

Vyřešíme tento příklad pomocí m-filu 2.6 tak, že do Matlabu zadáme příkaz

```
>> A=[1 2 1 2 3;1 2 2 1 -1;3 -2 1 0 -1;1 3 5 -1 -2;-1 2 3 1 1];
>> [detA] = determinant(A,10^-16)
```

Matlab nám vrátí výsledek

```
detA =

  -210
```

♠

2.1 Choleského rozklad

Tato podkapitola se zabývá rozkladem matice, která je hermitovská. Tento rozklad rozloží zadanou matici na součin horní trojúhelníkové matice a matice k ní hermitovské. Rozklad hermitovské matice je možné nalézt také pomocí Gaussovy eliminační metody a LU rozkladu, což ale z důvodu šetření aritmetickými operacemi a paměťmi počítače není výhodné, neboť tím, že je daná matice hermitovská se sníží počet aritmetických operací na polovinu. Zmiňovaný rozklad nám příblíží následující věta 2.4.

Věta 2.4. *Nechť matice* $A \in C_n$ *je hermitovská a pozitivně definitní. Pak existuje jediná horní trojúhelníková matice* $T \in C_n$ *s kladnými prvky na diagonále tak, že platí*

$$A = T^*T$$

Důkaz: viz. [1], strana 228.

Poznámka 2.8. Rozklad uvedený ve větě 2.4 se nazývá **Choleského rozklad**, někdy se také setkáváme s názvem metoda druhých odmocnin.

Uveďme si algoritmus, tzv. Choleského algoritmus, podle kterého můžeme vypočíst prvky matice T.

Algoritmus 2.2. Nechť A, $A = (a_{ij})$ je matice splňující předpoklady věty 2.4. Položme

$$t_{11} = \sqrt{a_{11}}$$

Pro $j = 2, \ldots, n$:

$$t_{1j} = \frac{a_{1j}}{t_{11}}$$

Pro $i = 2, \ldots, n$:

$$t_{ii} = \sqrt{a_{ii} - \sum_{l=1}^{i-1} (\overline{t_{li}}) t_{li}}$$

Pro $j > i$:

$$t_{ij} = \frac{1}{t_{ii}} \left(a_{ij} - \sum_{l=1}^{i-1} (\overline{t_{li}}) t_{lj} \right)$$

Pro $i > j$:

$$t_{ij} = 0$$

Poznámka 2.9. Pro reálné symetrické matice stačí předpoklad silné regularity, o čemž se můžeme dočíst ve skriptu [4].V tomto případě se na diagonále matice T může objevit komplexní číslo. A je-li reálná matice A navíc pozitivně definitní, pak všechny prvky matice T jsou reálné.

M-file 2.7. K ověření, zda je daná matice komplexní, hermitovská, silně regulární a pozitivně definitní jsem sestrojila tyto čtyři následující funkce, které využijeme v dalších m-filech

(i)
```
function v = komplexni(A)
%VSTUP:  A... matice
%VÝSTUP: v... pravdivostní hodnota
%            v=1 ... pro matice, které jsou komplexní
%            v=0 ... pro matice, které nejsou komplexní
v = 0;
for i = 1:length(A)
    for j = 1:length(A)
        if (imag(A(i,j)) ~= 0)
```

```
            v = 1;
            break;
        end
    end
end
```

(ii)
```
function v = Hermitovska(A)
%VSTUP:  A... matice
%VÝSTUP: v... pravdivostní hodnota
%           v=1 ... pro matice, které jsou hermitovské
%           v=0 ... pro matice, které nejsou hermitovské

[m,n] = size(A);
if (m == n)
    v = 1;
    pom = A == A';
    for i=1:m
        for j=1:m
            if (pom(i,j) ~= 1)
                v = 0;
                break;
            end
        end
        if (v == 0)
            break;
        end
    end
else
        v = 0;
```

```
end
```

(iii)
```
function v = TestSilneRegularity(A)
%VSTUP:  A... matice
%VÝSTUP: v... pravdivostní hodnota
%           v=1 ... pro matice, které jsou silně regulární
%           v=0 ... pro matice, které nejsou silně regulární
v = 1;
for i = 1:length(A)
    if (det(A(1:i,1:i)) == 0)
        v = 0;
        break;
    end
end
```

(iv)
```
function v = TestPozitivniDefinitnosti(A)
%VSTUP:  A... matice
%VÝSTUP: v... pravdivostní hodnota
%           v=1 ... pro matice, které jsou pozitivně definitní
%           v=0 ... pro matice, které nejsou pozitivně definitní
v = 1;
for i = 1:length(A)
    if (det(A(1:i,1:i)) <= 0)
    v = 0;
        break;
    end
end
```

♠

V následujících dvou příkladech si ukážeme jaké prvky bude mít matice T, jestliže zadaná matice bude pozitivně definitní nebo pouze silně regulární. Pro lepší porozumění Choleského metody si tyto dva příklady předvedeme na reálné symetrické matici, neboť rozklad reálné matice je jednodušší než rozklad komplexní matice.

V obou případech by podle poznámky 2.9 stačilo ověřit silnou regularitu. Ale jelikož má výsledná matice T pouze reálné prvky, když je pozitivně definitní, je dobré si tuto skutečnost předem zjistit.

Příklad 2.16. Najděte Choleského rozklad matice

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 2 & 7 & 2 \\ 4 & 2 & 35 \end{pmatrix}$$

Na první pohled vidíme, že matice A je symetrická. Nyní si ověříme, jestli je i pozitivně definitní. V Matlabu zavoláme na pomoc funkci 'TestPozitivniDefinitnosti', která nám zjistí, zda je tato matice pozitivně definitní. Tedy zadáme do Matlabu

```
>> A=[1 2 4;2 7 2; 4 2 35];
>> v = TestPozitivniDefinitnosti(A)


v =


     1
```

A vzhledem k tomu, že nám software Matlab vrátil pravdivostní hodnotu rovnu jedné, tak je daná matice pozitivně definitní a tím pádem i silně regulární, což plyne z věty 1.3. Tedy můžeme očekávat, že všechny prvky matice T budou reálné.

V dalších krocích postupujeme přesně podle algoritmu 2.2, tj.

$t_{11} = \sqrt{a_{11}} = \sqrt{1} = 1$
$t_{12} = \frac{a_{12}}{t_{11}} = \frac{2}{1} = 2$
$t_{13} = \frac{a_{13}}{t_{11}} = \frac{4}{1} = 4$
$t_{22} = \sqrt{a_{22} - t_{12}^2} = \sqrt{7 - 2^2} = \sqrt{3}$
$t_{23} = \frac{1}{t_{22}}(a_{23} - t_{12}t_{13} = \frac{1}{\sqrt{3}}(2 - 2 \cdot 4) = -2\sqrt{3}$
$t_{33} = \sqrt{a_{33} - (t_{13}^2 + t_{23}^2)} = \sqrt{35 - (4^2 + (-2\sqrt{3})^2)} = \sqrt{7}$
$t_{21} = t_{31} = t_{32} = 0$

Výsledky vložíme do matice a dostáváme

$$T = \begin{pmatrix} 1 & 2 & 4 \\ 0 & \sqrt{3} & -2\sqrt{3} \\ 0 & 0 & \sqrt{7} \end{pmatrix}$$

Snadno, například v Matlabu ověříme, že platí $T^T T = A$

$$T^T T = \begin{pmatrix} 1 & 0 & 0 \\ 2 & \sqrt{3} & 0 \\ 4 & -2\sqrt{3} & \sqrt{7} \end{pmatrix} \begin{pmatrix} 1 & 2 & 4 \\ 0 & \sqrt{3} & -2\sqrt{3} \\ 0 & 0 & \sqrt{7} \end{pmatrix} = \begin{pmatrix} 1 & 2 & 4 \\ 2 & 7 & 2 \\ 4 & 2 & 35 \end{pmatrix} = A$$

♠

Následuje příklad, kdy zadaná matice není pozitivně definitní a ukážeme, že matice T bude mít obecně komplexní prvky.

Příklad 2.17. Najděte rozklad matice A Choleského metodou

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

Postupujeme tak, že ověříme, jestli je tato matice silně regulární pomocí funkce 'TestSilneRegularity'. Tedy zadáme do Matlabu

```
>>  A=[1 1 2;1 4 3;2 3 1];

>> v = TestSilneRegularity(A)

 v =

       1
```

Vzhledem k tomu, že nám Matlab vrátil pravdivostní hodnotu rovnu jedné, tak je matice A silně regulární a my můžeme začít hledat prvky matice T podle algoritmu 2.2. Tedy

$$\begin{aligned}
&t_{11} = \sqrt{a_{11}} = \sqrt{1} = 1\\
&t_{12} = \tfrac{a_{12}}{t_{11}} = \tfrac{1}{1} = 1\\
&t_{13} = \tfrac{a_{13}}{t_{11}} = \tfrac{2}{1} = 2\\
&t_{22} = \sqrt{a_{22} - t_{12}^2} = \sqrt{4 - 1^2} = \sqrt{3}\\
&t_{23} = \tfrac{1}{t_{22}}(a_{23} - t_{12}t_{13}) = \tfrac{1}{\sqrt{3}}(3 - 1 \cdot 2) = \tfrac{1}{\sqrt{3}}\\
&t_{33} = \sqrt{a_{33} - (t_{13}^2 + t_{23}^2)} = \sqrt{1 - (2^2 + (\tfrac{1}{\sqrt{3}})^2)} = \sqrt{1 - \tfrac{13}{3}} = i\sqrt{\tfrac{10}{3}}\\
&t_{21} = t_{31} = t_{32} = 0
\end{aligned}$$

Výsledky vložíme do matice a dostáváme

$$T = \begin{pmatrix} 1 & 1 & 2 \\ 0 & \sqrt{3} & \frac{1}{\sqrt{3}} \\ 0 & 0 & i\sqrt{\frac{10}{3}} \end{pmatrix}$$

Snadno v Matlabu ověříme, že platí $T^T T = A$

$$T^T T = \begin{pmatrix} 1 & 0 & 0 \\ 1 & \sqrt{3} & 0 \\ 2 & \frac{1}{\sqrt{3}} & i\sqrt{\frac{10}{3}} \end{pmatrix} \begin{pmatrix} 1 & 1 & 2 \\ 0 & \sqrt{3} & \frac{1}{\sqrt{3}} \\ 0 & 0 & i\sqrt{\frac{10}{3}} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 4 & 3 \\ 2 & 3 & 1 \end{pmatrix} = A$$

Vidíme, že když nebyla zadaná matice pozitivně definitní, tak se na diagonále matice T vyskytlo komplexní číslo. Prvky mimo diagonálu nám vyšly pouze reálné

♠

M-file 2.8. Na základě algoritmu 2.2 jsem v Matlabu sestavila m-file, který nejdříve ověří, zda je vložená matice hermitovská a poté, zda je komplexní a zároveň pozitivně definitní nebo zda je reálná a silně regulární. Následně spočítá prvky matice T.

```
function T = CholeskehoRozklad(A)
%VSTUP:  A... čtvercová matice
%VÝSTUP: T... výsledná matice z choleského rozkladu
```

```
if (0 == Hermitovska(A))
    error('Matice není hermitovska')
elseif (1 == komplexni(A) && 1 == TestPozitivniDefinitnosti(A)) ||
        (0 == komplexni(A) && 1 == TestSilneRegularity(A))
    T(1,1) = sqrt(A(1,1));
    n = length(A);
    for j = 2:n
        T(1,j) = A(1,j) / T(1,1);
    end
    for i = 2:n-1
        suma = 0;
        for k = 1:i-1
            if (1 == komplexni(A))
                suma = suma + conj(T(k,i))*T(k,i);
            else
                suma = suma + T(k,i)^2;
            end
        end
        T(i,i) = sqrt(A(i,i) - suma);
    end
    for i = 2:n
        for j = 3:n
            if (j > i)
               sum = 0;
              for k = 1:i-1
                  if (1 == komplexni(A))
                      sum = sum + conj(T(k,i)) * T(k,j);
                  else
```

```
                    sum = sum + T(k,i) * T(k,j);
                end
            end
            T(i,j) = ((A(i,j) - sum)/ T(i,i));
        else
            sumsum = 0;
            for p = 1:i-1
                if (1 == komplexni(A))
                    sumsum = sumsum + conj(T(p,i))*T(p,i);
                else
                    sumsum = sumsum + T(p,i)*T(p,i);
                end
            end
            T(i,i) = sqrt(A(i,i) - sumsum);
        end
    end
end
else
    error('bud' je zadaná matice komplexní, ale není pozitivně definitní,
           nebo není zadaná matice komplexní a ani silně regulární')
end
```

♠

Nyní si ukážeme Choleského rozklad matice, která je komplexní.

Příklad 2.18. Najděte Choleského rozklad matice A

$$A = \begin{pmatrix} 1 & 2-i & 3 \\ 2+i & 10 & i \\ 3 & -i & 30 \end{pmatrix}$$

Vyřešíme tento příklad pomocí m-filu 2.8 tak, že do Matlabu zadáme příkaz

```
>> A=[1 2-i 3;2+i 10 i;3 -i 30];
>> T = CholeskehoRozklad(A)
```

Matlab nám vrátí následující výsledek

```
T =

   1.0000             2.0000 - 1.0000i   3.0000
        0             2.2361            -2.6833 - 0.8944i
        0                  0             3.6056
```

♠

Poznámka 2.10. Stejně jako pro *LU* rozklad má sofware Matlab příkaz i pro Choleského rozklad 'chol'. Narozdíl však od m-filu 2.8 je u toho příkazu zapotřebí, aby zadaná matice (i reálná) byla pozitivně definitní. Pokud taková nebude, Matlab vrátí chybu a rozklad nespočítá. Ukážeme si užití a funkčnost tohoto příkazu na následujících příkladech 2.19 a 2.20.

Příklad 2.19. Najděte choleského rozklad matice

$$A = \begin{pmatrix} 1 & 2 & 1 & -1 \\ 2 & 5 & 4 & -1 \\ 1 & 4 & 6 & 0 \\ -1 & -1 & 0 & 12 \end{pmatrix}$$

Nejdříve pomocí příkazu 'TestPozitivniDefinitnosti' ověříme, zda je zadaná matice pozitivně definitní. Zadáme tedy do Matlabu příkaz

```
>> A=[1 2 1 -1;2 5 4 -1;1 4 6 0;-1 -1 0 12];
>> v = TestPozitivniDefinitnosti(A)


v =

     1
```

A vzhledem k tomu, že nám software Matlab vrátil pravdivostní hodnotu rovnu jedné, tak je zadaná matice pozitivně definitní. Nyní můžeme přistoupit k samotnému Choleského rozkladu. Zadáme do Matlabu příkaz

```
>> A=[1 2 1 -1;2 5 4 -1;1 4 6 0;-1 -1 0 12];
>> T=chol(A)
```

Matlab nám vrátí následující výsledek

```
T =

     1     2     1    -1
     0     1     2     1
     0     0     1    -1
     0     0     0     3
```

Příkazem

```
>> T'*T
```

můžeme zkontrolovat, zda jsme počítali správně.

♠

Příklad 2.20. Najděte choleského rozklad matice

$$A = \begin{pmatrix} 1 & 2 & 1 & -1 \\ 2 & 5 & 4 & -1 \\ 1 & 4 & 6 & 0 \\ -1 & -1 & 0 & 1 \end{pmatrix}$$

Nejdříve pomocí příkazu 'TestPozitivniDefinitnosti' ověříme, zda je zadaná matice pozitivně definitní. Zadáme tedy do Matlabu příkaz

```
>> A=[1 2 1 -1;2 5 4 -1;1 4 6 0;-1 -1 0 1];
>> v = TestPozitivniDefinitnosti(A)

v =

     0
```

A vzhledem k tomu, že nám software Matlab vrátil pravdivostní hodnotu rovnu nule, tak zadaná matice není pozitivně definitní. Zadáme-li do Matlabu příkaz

```
>> A=[1 2 1 -1;2 5 4 -1;1 4 6 0;-1 -1 0 1];
>> T=chol(A)
```

tak nám Matlab vrátí následující výsledek

```
>> T=chol(A)
??? Error using ==> chol
Matrix must be positive definite.
```

Použijeme m-file 2.8. Zadáme do Matlabu příkaz

```
>> A=[1 2 1 -1;2 5 4 -1;1 4 6 0;-1 -1 0 1];
>> T = CholeskehoRozklad(A)
```

a dostaneme výsledek

```
T =

   1.0000              2.0000              1.0000             -1.0000
        0              1.0000              2.0000              1.0000
        0                   0              1.0000             -1.0000
        0                   0                   0          0 + 1.4142i
```

Tedy vidíme, že narozdíl od příkazu 'chol' m-file 2.8 dokáže najít choleského rozklad reálné matice, která není pozitivně definitní.

2.2 Croutova metoda

Podívejme se nyní blíže na LU rozklad matice, která je třídiagonální, neboli (1,1)-pásová matice. Máme zadánu následující matici $A \in C_n$ tvaru

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 & \dots & \dots & 0 \\ a_{21} & a_{22} & a_{23} & \dots & \dots & 0 \\ 0 & a_{32} & a_{33} & & & \vdots \\ \vdots & & & \ddots & & 0 \\ \vdots & & & & a_{n-1,n-1} & a_{n-1,n} \\ 0 & \dots & \dots & 0 & a_{n,n-1} & a_{nn} \end{pmatrix}$$

a hledáme LU rozklad matice A, tj. hledáme matice L a U ve tvaru

$$L = \begin{pmatrix} l_{11} & 0 & 0 & \dots & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & \dots & 0 \\ 0 & l_{32} & l_{33} & & & \vdots \\ \vdots & & & \ddots & & 0 \\ \vdots & & & & l_{n-1,n-1} & 0 \\ 0 & \dots & \dots & 0 & l_{n,n-1} & l_{nn} \end{pmatrix}, \quad U = \begin{pmatrix} 1 & u_{12} & 0 & \dots & \dots & 0 \\ 0 & 1 & u_{23} & \dots & \dots & 0 \\ 0 & 0 & 1 & & & \vdots \\ \vdots & & & \ddots & & 0 \\ \vdots & & & & 1 & u_{n-1,n} \\ 0 & \dots & \dots & 0 & 0 & 1 \end{pmatrix} \tag{5}$$

Pro určení $(2n-1)$ prvků matice L a $(n-1)$ prvků matice U (dohromady $(3n-2)$ prvků), nám pomůže následující algoritmus, který vznikl porovnáním prvků matice A s odpovídajícími prvky součinu LU

Algoritmus 2.3. Nechť je dána třídiagonální matice $A \in C_n, A = (a_{ij})$. Položme

$$a_{11} = l_{11}$$

Pro $i = 2, \dots, n$ položme:

$$a_{i,i-1} = l_{i,i-1}$$

Pro $i = 2, \dots, n$ vypočteme:

$$a_{i,i} = l_{i,i-1}u_{i-1,i} + l_{ii}$$

pro $i = 1, \dots, n-1$ vypočteme:

$$a_{i,i+1} = l_{i,i}u_{i,i+1}$$

Poznámka 2.11. Uvedený algoritmus 2.3 se nazývá **Croutova metoda**.

Věta 2.5. *Nechť $A \in C_n$ je třídiagonální matice s vlastnostmi:*

(i) $a_{i,i-1}a_{i,i+1} \neq 0$ *pro* $i = 2, \dots, n-1$

(ii) A je řádkově diagonálně dominantní.

Pak matice A je regulární a hodnoty $l_{ii}, i = 1, \dots, n$, vypočtené algoritmem 2.3 jsou různé od nuly.

Důkaz: viz. [1].

Poznámka 2.12. Jsou-li splněny předpoklady věty 2.5, pak matici A lze rozložit na součin dolní a horní trojúhelníkové matice tvaru (5), což ilustruje následující příklad.

Příklad 2.21. Nalezněte LU rozklad třídiagonální matice A pomocí Croutovy metody

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

Vidíme, že daná matice je řádkově diagonálně dominantní, třídiagonální a že $a_{21}a_{23} = 1 \cdot 1 \neq 0$ a $a_{32}a_{34} = 1 \cdot 1 \neq 0$, tudíž jsou splněny předpoklady věty 2.5

a můžeme k výpočtu LU rozkladu použít Croutovu metodu.

Počítáme

$l_{11} = a_{11} = 4$
$u_{12} = \frac{a_{12}}{l_{11}} = \frac{1}{4}$
$l_{21} = a_{21} = 1$
$l_{22} = a_{22} - l_{21}u_{12} = 2 - 1 \cdot \frac{1}{4} = \frac{7}{4}$
$u_{23} = \frac{a_{23}}{l_{22}} = \frac{1}{\frac{7}{4}} = \frac{4}{7}$
$l_{32} = a_{32} = 1$
$l_{33} = a_{33} - l_{32}u_{23} = 5 - 1 \cdot \frac{4}{7} = \frac{31}{7}$
$u_{34} = \frac{a_{34}}{l_{33}} = \frac{1}{\frac{31}{7}} = \frac{7}{31}$
$l_{43} = a_{43} = 2$
$l_{44} = a_{44} - l_{43}u_{34} = 3 - 2 \cdot \frac{7}{31} = \frac{79}{31}$

Našli jsme matice L a U ve tvaru

$$L = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 1 & \frac{7}{4} & 0 & 0 \\ 0 & 1 & \frac{31}{7} & 0 \\ 0 & 0 & 2 & \frac{79}{31} \end{pmatrix}$$

$$U = \begin{pmatrix} 1 & \frac{1}{4} & 0 & 0 \\ 0 & 1 & \frac{4}{7} & 0 \\ 0 & 0 & 1 & \frac{7}{31} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

pro kontrolu nasobíme

$$LU = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 1 & \frac{7}{4} & 0 & 0 \\ 0 & 1 & \frac{31}{7} & 0 \\ 0 & 0 & 2 & \frac{79}{31} \end{pmatrix} \begin{pmatrix} 1 & \frac{1}{4} & 0 & 0 \\ 0 & 1 & \frac{4}{7} & 0 \\ 0 & 0 & 1 & \frac{7}{31} \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 5 & 1 \\ 0 & 0 & 2 & 3 \end{pmatrix} = A$$

♠

M-file 2.9. K výpočtu LU rozkladu matice A pomocí Croutovy metody jsem sestavila m-file v matematickém softwaru Matlab. Tento program na začátku ověří, zda zadaná matice je čtvercová. Poté zkontroluje, je-li matice třídiagonální a pomocí funkce 'podminka', jestli splňuje podmínku (i) z věty 2.5. Na závěr ověří pomocí funkce 'RadkoveDiagonalneDominantni', je-li matice A řádkově diagonálně dominantní. Pokud jsou všechny tyto podmínky splněny, tak daná matice vyhovuje požadavkům věty 2.5 a m-file vypočte prvky matic L a U.

(i)
```
function r = RadkoveDiagonalneDominantni(A)
%VSTUP:  A... matice
%VÝSTUP: r... pravdivostní hodnota
%            r=1 ... pro matice, které jsou řádkově
                     diagonálně dominantní
%            r=0 ... pro matice, které nejsou řádkově
                     diagonálně dominantní
[m,n]= size(A);
if (m ~= n)
    error('zadaná matice není čtvercová')
else
    temp = 0;
    r = 1;
    for i=1:n
       for j=1:n
           if (i ~= j)
             temp = temp + abs(A(i,j));
             if (abs(A(i,i)) <= temp)
                r = 0;
                break;
             end
             temp = 0;
           end
       end
    end
```

```
end
```

(ii)
```
function r = podminka(A,eps)
%VSTUP:  A... matice
%        eps... požadovaná přesnost
%VÝSTUP: r... pravdivostní hodnota
%           r=1 ... pro matice, které splňují
                    předpoklad (i) věty 2.5.
%           r=0 ... pro matice, které nesplňují
                    předpoklad (i) věty 2.5.
[m,n]= size(A);
if (m ~= n)
    error('zadaná matice není čtvercová')
else
    temp = 0;
    r=1;
    for i=2:n-1
        if (abs(A(i,i-1)*A(i,i+1)) < eps)
            r=0;
            break;
        end
    end
end
```

(iii)
```
function [L,U] = CroutovaMetoda(A,eps)
%VSTUP:  A... čtvercová matice
%        eps... požadovaná přesnost
%VÝSTUP: L...dolní trojúhelníková matice
%        U...horní trojúhelníková matice
```

```
[k,l]=size(A);
n = length(A);
for r=2:n-1
   if (0 == podminka(A))
      error('ERROR - Croutova metoda nelze použít,
                     neni splnena podminka (i) vety 2.5.')
   end
end
if k~=l
    error('ERROR - matice A není čtvercová')
else
    stop=1;
    for k=1:n
        for j=1:n
           if (abs(k-j)>1)
              if (abs(A(k,j))>= eps | abs(A(j,k))>= eps)
                 stop=0;
              end
           end
        end
    end
        if (1 == RadkoveDiagonalneDominantni(A))
           if (stop == 1)
              U=zeros(n);
              L=zeros(n);
              L(1,1)  = A(1,1);
              U(1,2) = (A(1,2)./L(1,1));
              U(1,1)=1;
```

```
        for i=2:n
            if (i<n)
                U(i,i)=1;
                L(i,i-1)=A(i,i-1);
                L(i,i)=A(i,i)-(L(i,i-1).*U(i-1,i));
                U(i,i+1) = (A(i,i+1)./L(i,i));
            else
                U(i,i)=1;
                L(i,i-1)=A(i,i-1);
                L(i,i)=A(i,i)-(L(i,i-1).*U(i-1,i));
            end
        end

    else
        disp('matice neni tridiagonalni')
    end

else
        disp('Matice neni řádkově diagonalne dominantni')
end
end
```

♠

Příklad 2.22. Nalezněte LU rozklad matice A pomocí Croutovy metody

$$A = \begin{pmatrix} 6 & 3 & 0 & 0 & 0 \\ 4 & 5 & 2 & 0 & 0 \\ 0 & 4 & 12 & 8 & 0 \\ 0 & 0 & 1 & 13 & 1 \\ 0 & 0 & 0 & 2 & 7 \end{pmatrix}$$

Vyřešíme tento příklad pomocí m-filu 2.9 tak, že do Matlabu zadáme příkaz

```
>> A=[6 3 0 0 0;4 5 2 0 0;0 4 12 8 0;0 0 1 13 1;0 0 0 2 7];
>> [L,U] = CroutovaMetoda(A,10^-16)
```

Matlab nám vrátí následující výsledek

```
L =

    6.0000         0         0         0         0
    4.0000    3.0000         0         0         0
         0    4.0000    9.3333         0         0
         0         0    1.0000   12.1429         0
         0         0         0    2.0000    6.8353



U =

    1.0000    0.5000         0         0         0
         0    1.0000    0.6667         0         0
         0         0    1.0000    0.8571         0
         0         0         0    1.0000    0.0824
         0         0         0         0    1.0000
```

♠

Nyní si ukážeme aplikaci m-filu 2.9 na komplexní matici.

Příklad 2.23. Nalezněte LU rozklad matice A pomocí Croutovy metody

$$A = \begin{pmatrix} 5 & 2 & 0 & 0 & 0 \\ i & 6 & 2 & 0 & 0 \\ 0 & 2 & 7 & 3 & 0 \\ 0 & 0 & 1 & 14 & 5 \\ 0 & 0 & 0 & 1 & 6 \end{pmatrix}$$

Vyřešíme tento příklad pomocí m-filu 2.9 tak, že do Matlabu zadáme příkaz

```
>> A=[ 5 2 0 0 0;i 6 2 0 0 ;0 2 7 3 0;0 0 1 14 5; 0 0 0 1 6 ];
>> [L,U] = CroutovaMetoda(A,10^-16)
```

Matlab nám vrátí matice L a U

```
L =


   5        0            0                  0                  0
   i    6 - 0.4i         0                  0                  0
   0        2     6.3363 - 0.0442i          0                  0
   0        0            1           13.5266 - 0.0033i         0
   0        0            0                  1            5.6304 - 0.0001i



U =


   1      0.4            0                  0                  0
   0        1     0.3319 + 0.0221i          0                  0
   0        0            1           0.4734 + 0.0033i          0
   0        0            0                  1            0.3696 + 0.0001i
   0        0            0                  0                  1
```

♠

3 Skeletový rozklad

V této kapitole se budeme zabývat součinovým rozkladem matice, který je známý pod pojmem skeletový rozklad. S tímto rozkladem se můžeme setkat hlavně při určování pseudoinverze, což se dále využívá při řešení systému lineárních rovnic. Nejdříve uveďme větu, která nám skeletový rozklad přiblíží.

Definice 3.1. **Skeletovým rozkladem matice** $A \in C_{m,n}$ nazýváme každou uspořádanou dvojici matic $(B; C)$, pro kterou platí

(i) $B \in C_{m,h}$

(ii) $C \in C_{h,n}$

(iii) $A = BC$

kde $h = h(A) = h(B) = h(C)$

Věta 3.1. *Ke každé matici $A \in C_{m,n}$ existuje alespoň jeden skeletový rozklad.*

Důkaz: viz [3], strana 97. Tento důkaz uvedeme, jelikož je v něm obsažena konstrukce, jak k dané matici najít její skeletový rozklad. Ke každé matici $A \in C_{m,n}, A = (a_{ij})$ existuje regulární matice $T \in C_m$ taková, že $TA = S$, kde $S \in C_{m,n}$ je ve stupňovitém tvaru. Je-li $h(A) = h$, je prvních h řádků matice S lineárně nezávislých a posledních $(m - h)$ řádků nulových. Pokud je matice B tvořená prvními h sloupci matice T^{-1} a C matice tvořená nenulovými řádky matice S, pak $A = BC$. $\square$

Postup nalezení skeletového rozkladu ukážeme blíže na příkladu

Příklad 3.1. Najděte skeletový rozklad matice A

$$A = \begin{pmatrix} 1 & 2 & 4 & -1 & 1 \\ 1 & 3 & 7 & 1 & 3 \\ 2 & 7 & 17 & 4 & 8 \\ -1 & -1 & -1 & 3 & 1 \end{pmatrix}$$

Budeme postupovat podle důkazu věty 3.1. Nejdříve hledáme matici S ve stupňovitém tvaru. Tedy prvním krokem bude převedení matice A na stupňovitý tvar a to tak, že na ní aplikujeme Gaussovu eliminační metodu, tedy v prvním kroku vynulujeme prvky na pozicích a_{i1} pro $i = 2, 3, 4$ a dostáváme matici

$$\begin{pmatrix} 1 & 2 & 4 & -1 & 1 \\ 0 & 1 & 3 & 2 & 2 \\ 0 & 3 & 9 & 6 & 6 \\ 0 & 1 & 3 & 2 & 2 \end{pmatrix}$$

Nyní tuto matici vhodným násobením upravíme na ekvivalentní tvar a dostáváme matici

$$\begin{pmatrix} 1 & 2 & 4 & -1 & 1 \\ 0 & 1 & 3 & 2 & 2 \\ 0 & 1 & 3 & 2 & 2 \\ 0 & 1 & 3 & 2 & 2 \end{pmatrix}$$

Vidíme, že poslední tři řádky jsou lineárně závislé, tedy hodnost této matice je dvě. Tuto matici, kterou označíme S, můžeme psát ve tvaru

$$S = \begin{pmatrix} 1 & 2 & 4 & -1 & 1 \\ 0 & 1 & 3 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Z důkazu věty 3.1 víme, že existuje regulární matice T taková, že $TA = S$ Tedy

$$TA = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -2 & -3 & 1 & 0 \\ 1 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 4 & -1 & 1 \\ 1 & 3 & 7 & 1 & 3 \\ 2 & 7 & 17 & 4 & 8 \\ -1 & -1 & -1 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 4 & -1 & 1 \\ 0 & 1 & 3 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} = S$$

kde prvky na hlavní diagonále matice T jsou jedničky, prvky nad hlavní diagonálou jsou nuly a prvky pod hlavní diagonálou jsou multiplakátory získané z matice A postupem vysvětleným u Gaussovy eliminační metody.

Jelikož $h(A) = 2$, dostáváme matici B ve tvaru

$$B = \begin{pmatrix} 1 & 1 & 2 & -1 \\ 0 & 1 & 3 & 1 \end{pmatrix}^T$$

a matici C, která je tvořena nenulovými řádky matice A

$$C = \begin{pmatrix} 1 & 2 & 4 & -1 & 1 \\ 0 & 1 & 3 & 2 & 2 \end{pmatrix}$$

Matice B a C tvoří skeletový rozklad matice A. Přesvědčíme se o tom jednoduchou zkouškou

$$BC = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 2 & 3 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 4 & -1 & 1 \\ 0 & 1 & 3 & 2 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 4 & -1 & 1 \\ 1 & 3 & 7 & 1 & 3 \\ 2 & 7 & 17 & 4 & 8 \\ -1 & -1 & -1 & 3 & 1 \end{pmatrix} = A$$

♠

M-file 3.1. V matematickém programu Matlab jsem vytvořila m-file, který nejprve převede danou matici na schodovitý tvar, pomocí funkce 'SchodTvar'. Zároveň se multiplikátory, kterými jsme násobili řádky, uloží do matice B a poté určí matici C.

(i)
```
function [A,B]=SchodTvar(A,eps)
%VSTUP:  A... matice
%        eps... požadovaná přesnost
%VÝSTUP: A...matice upravená na schodovitý tvar
%        B...matice sestavená z multiplikátorů
r=rank(A);
[m,n]=size(A);
B=eye(m,r);
P=eye(m);
```

```
i = 1;
while i<m
     if abs(A(i,:)) <= eps
        for j=m:-1:i+1
            if abs(A(j,:)) > eps
                temp = A(j,:);
                A(j,:)=A(i,:);
                A(i,:)=temp;
                if i<j
                    tempB = B(i,1:i-1);
                    B(i,1:i-1) = B(j,1:i-1);
                    B(j,1:i-1) = tempB;
                elseif i>j
                    tempB = B(i,1:j-1);
                    B(i,1:j-1) = B(j,1:j-1);
                    B(j,1:j-1) = tempB;
                end
                break;
            end
        end
    end
     if (abs(A(i,i)) <= eps)
        for j=1:m
           if (abs(A(i,j)) > eps) & (abs(A(j,i)) > eps)
              temp = A(j,:);
              A(j,:)=A(i,:);
              A(i,:)=temp;
              if i<j
```

```
                tempB = B(i,1:i-1);
                B(i,1:i-1) = B(j,1:i-1);
                B(j,1:i-1) = tempB;
             elseif i>j
                tempB = B(i,1:j-1);
                B(i,1:j-1) = B(j,1:j-1);
                B(j,1:j-1) = tempB;
             end
             break;
          end
       end
    end
    if (abs(A(i,i)) <=eps)
       i = i + 1;
       continue;
    end
    for j=i+1:m
       B(j,i)=A(j,i)/A(i,i);
       A(j,:)=A(j,:) - B(j,i)*A(i,:);
    end
    i = i + 1;
end
```

(ii)
```
function [B,C] = SkeletovyRozklad(A, eps)
%VSTUP:  A... matice
%        eps... požadovaná přesnost
%VÝSTUP: B...matice ze skeletova rozkladu matice A
%        C...matice ze skeletova rozkladu matice A
```

```
[m,n]=size(A);
[A,B]=SchodTvar(A);
temp = [];
for i=1:m
    temp(i) = 0;
    for j=1:n
        if abs(A(i,j)) >= eps
            temp(i) = 1;
        end
    end
end
temp;
temp2 = 0;
for i=1:m
    if temp(i) == 0
        A(i-temp2,:) = [];
        temp2 = temp2 + 1;
    end
end
C=A;
```

♠

Příklad 3.2. Najděte skeletový rozklad matice A

$$A = \begin{pmatrix} 1 & 2 & -1 & 1 \\ 1 & 3 & 4 & 2 \\ -1 & 0 & 11 & 1 \\ 2 & 7 & 13 & 5 \end{pmatrix}$$

Vyřešíme tento příklad pomocí m-filu 3.1 tak, že do Matlabu zadáme příkaz

```
>> A=[1 2 -1 1;1 3 4 2;-1 0 11 1;2 7 13 5];
>> [B,C]=SkeletovyRozklad(A,10^-16)
```

Matlab nám vrátí výsledné matice

```
B =

     1     0
     1     1
    -1     2
     2     3


C =

     1     2    -1     1
     0     1     5     1
```

Zadáme pro kontrolu do Matlabu příkaz $B * C$ a dostaneme původní matici A

```
>> B*C

ans =

     1     2    -1     1
     1     3     4     2
    -1     0    11     1
     2     7    13     5
```

♠

Nyní si na příkladě ukážeme funkčnost m-filu 3.1 i pro komplexní matice.

Příklad 3.3. Najděte skeletový rozklad matice A

$$A = \begin{pmatrix} i & 2 & 1 \\ 2 & 1-2i & 5 \\ i & -3i & 9 \end{pmatrix}$$

Do Matlabu zadáme příkaz

```
>> A=[i 2 1; 2 1-2i 5;i -3i 9];
>> [B,C] = SkeletovyRozklad(A, eps)
```

Matlab nám vrátí výsledné matice

```
B =

   1.0000                  0                  0
        0 - 2.0000i   1.0000                  0
   1.0000            -1.6000 + 0.2000i   1.0000


C =

        0 + 1.0000i   2.0000              1.0000
        0             1.0000 + 2.0000i   5.0000 + 2.0000i
        0                  0            16.4000 + 2.2000i
```

Zadáme-li pro kontrolu do Matlabu příkaz $B * C$, dostaneme původní matici A

```
>> B*C
ans =
        0 + 1.0000i   2.0000              1.0000
   2.0000             1.0000 - 2.0000i   5.0000
        0 + 1.0000i  -0.0000 - 3.0000i   9.0000 + 0.0000i
```

♠

Poznámka 3.1. Ačkoli je m-file 3.1 vytvořen přesně podle návodu, který je uveden v důkazu věty 3.1, tak není dostatečně efektivní. Používá se v něm převod na schodovitý tvar, což u velkých matic může být zdlouhavé. Navíc tento m-file může zkolabovat například na stejném problému velkých multiplikátorů jako v příkladu 2.9. Z toho důvodu jsem v softwaru Matlab naprogramovala m-file 3.2.

M-file 3.2. Tento nový m-file na výpočet skeletového rozkladu pracuje tak, že zjistí všechny lineárně nezávislé sloupce matice A a ty přiřadí do matice B. Matice C se pak vypočte maticovým dělením $C = B\backslash A$.

```
function [B,C] = NovySkeletovyRozklad(A, eps)
%VSTUP:  A... matice
%        eps... požadovaná přesnost
%VÝSTUP: B...matice ze skeletova rozkladu matice A
%        C...matice ze skeletova rozkladu matice A
h=rank(A);
B(:,1) = A(:,1);
[m,n] = size(A);
k=rank(B) + 1;
for i=2:n
    B2 = [B, A(:,i)];
    if (rank(B2) == k)
        B = B2;
        k=rank(B) + 1;
    end
end
C=B\A;
```

♠

Nefunkčnost m-filu 3.1 a funkčnost m-filu 3.2 pro konkrétní matice si ukážeme na následujícím příkladě.

Příklad 3.4. Nalezněte skeletový rozklad matice

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 1 & 2 & 1 \end{pmatrix}$$

Nejdříve použijeme m-file 3.1, tj. zadáme do Matlabu

```
>> A=[ 1 2 3;2 4 6;1 2 1];

>> [B,C] = SkeletovyRozklad(A, 10^-16)
```

Matlab nám vrátí výsledek

```
B =

     1     0
     2     1
     1     0



C =

     1     2     3
     0     0    -2
```

Pro kontrolu zadáme

```
>> B*C

ans =

     1     2     3
     2     4     4
     1     2     3
```

A vidíme, že matice $BC \neq A$. Program nám tedy nespočítal skeletový rozklad správně, což je způsobeno tím, že matici převádíme na schodovitý tvar. Zkusíme proto m-file 3.2, tj. zadáme do Matlabu

```
>> A=[ 1 2 3;2 4 6;1 2 1];
>> [B,C] = NovySkeletovyRozklad(A, 10^-16)
```

dostáváme výsledek

```
B =

     1     3
     2     6
     1     1

C =

    1    2    0
    0    0    1
```

Nyní pro kontrolu opět zadáme

```
>> B*C

ans =
       1               2               3
       2               4               6
       1               2               1
```

Nyní vidíme, že $BC = A$ a tedy že m-file 3.2 nám našel správný skeletový rozklad.

♠

Poznámka 3.2.

(i) Je-li $(B; C)$ skeletový rozklad matice A, pak zřejmě obě matice B^*B, CC^* jsou hermitovské , pozitivně definitní, hodnosti $h = h(A)$. A to plyne z toho, že nerovnost $x^*B^*Bx \geq 0$ platí pro všechna x a $x^*B^*Bx = 0$ tehdy a jen tehdy, je-li $Bx = 0$. A případ, kdy $Bx = 0$ nastává právě tehdy, když $x = o$.

(ii) Skeletový rozklad matice A není určen jednoznačně. Známe-li však jeden skeletový rozklad, můžeme snadno popsat jiný, jak je řečeno v následující větě.

Věta 3.2. *Je-li $(B; C)$ skeletový rozklad matice A s hodností h, pak $(B_1; C_1)$ je skeletovým rozkladem téže matice v tom a jen v tom případě, existuje-li regulární matice $Z \in C_h$ taková, že*

$$BZ = B_1$$

$$Z^{-1}C = C_1$$

Důkaz: viz. [3], strana 98.

Nyní si ukážeme aplikaci předchozí věty 3.2. Pomocí jednoho, již známého, skeletového rozkladu najdeme druhý skeletový rozklad.

Příklad 3.5. Nechť je dána matice A z příkladu 3.1, která má hodnost $h = 2$ a nechť máme k dispozici již nalezené matice B a C. Tedy

$$A = \begin{pmatrix} 1 & 2 & 4 & -1 & 1 \\ 1 & 3 & 7 & 1 & 3 \\ 2 & 7 & 17 & 4 & 8 \\ -1 & -1 & -1 & 3 & 1 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & 1 & 2 & -1 \\ 0 & 1 & 3 & 1 \end{pmatrix}^T \qquad C = \begin{pmatrix} 1 & 2 & 4 & -1 & 1 \\ 0 & 1 & 3 & 2 & 2 \end{pmatrix}$$

Nyní si zvolíme nějakou čtvercovou regulární matici $Z \in C_2$, jejíž rozměr bude roven hodnosti matice A. Například

$$Z = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Podle vztahů z věty 3.2 teď vypočítáme matice B_1 a C_1

$$B_1 = BZ = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 2 & 3 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 4 & 6 \\ 11 & 16 \\ 2 & 2 \end{pmatrix}$$

$$C_1 = Z^{-1}C = \begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} 1 & 2 & 4 & -1 & 1 \\ 0 & 1 & 3 & 2 & 2 \end{pmatrix} = \begin{pmatrix} -2 & -3 & -5 & 4 & 0 \\ \frac{3}{2} & \frac{5}{2} & \frac{9}{2} & -\frac{5}{2} & \frac{1}{2} \end{pmatrix}$$

Nyní pomocí násobení zkontrolujeme, zda se opravdu jedná o skeletový rozklad, tj.

$$B_1C_1 = \begin{pmatrix} 1 & 2 \\ 4 & 6 \\ 11 & 16 \\ 2 & 2 \end{pmatrix} \begin{pmatrix} -2 & -3 & -5 & 4 & 0 \\ \frac{3}{2} & \frac{5}{2} & \frac{9}{2} & -\frac{5}{2} & \frac{1}{2} \end{pmatrix} = \begin{pmatrix} 1 & 2 & 4 & -1 & 1 \\ 1 & 3 & 7 & 1 & 3 \\ 2 & 7 & 17 & 4 & 8 \\ -1 & -1 & -1 & 3 & 1 \end{pmatrix} = A$$

♠

Skeletový rozklad se s výhodou používá při určování pseudoinverzní matice, o čemž hovoří následující věta 3.3.

Věta 3.3. *Nechť $A \in C_{m,n}$ je nenulová matice a nechť $A = BC$ je skeletový rozklad matice A. Potom pro pseudoinverzi matice A platí*

$$A^+ = C^+B^+$$

Důkaz: Je založen na ověření Moore - Penroseových podmínek. Z poznámky 1.8 plyne, že $CC^+ = CC^*(CC^*)^{-1} = I$ a $B^+B = BB^*(BB^*)^{-1} = I$. Označíme si $X = C^+B^+$ a začneme ověřovat podmínky z definice 1.16

(i) $AXA = BCC^+B^+BC = BC = A$

(ii) $XAX = C^+B^+BCC^+B^+ = C^+B^+ = X$

(iii) $(AX)^* = (BCC^+B^+)^* = (BB^+)^* = BB^+ = BCC^+B^+ = AX$

(iv) $(XA)^* = (C^+B^+BC)^* = (C^+C)^* = C^+C = C^+B^+BC = XA$

Tedy $X = A^+$. □

Vidíme, že díky skeletovému rozkladu můžeme snáze určit pseudoinverzi dané matice, jak říká věta 3.3. Ukážeme si toto využití skeletového rozkladu na příkladě.

Příklad 3.6. Určete Moore - Penroseovu inverzi matice A užítím skeletového rozkladu.

$$A = \begin{pmatrix} 1 & -1 & -2 \\ 1 & 1 & 0 \\ 3 & 2 & -1 \end{pmatrix}$$

K nalezení skeletového rozkladu matice A, použijeme m-file 3.2. Dostaneme tak matice

$$B = \begin{pmatrix} 1 & -1 \\ 1 & 1 \\ 3 & 2 \end{pmatrix}, \qquad C = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \end{pmatrix}$$

Pseudoinverzi matice B a C vypočítáme podle poznámky 1.8, tedy dostaneme:

$$B^+ = (B^*B)^{-1}B^* = \begin{pmatrix} 11 & 6 \\ 6 & 6 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 1 & 3 \\ -1 & 1 & 2 \end{pmatrix} = \tfrac{1}{30} \begin{pmatrix} 12 & 0 & 6 \\ -17 & 5 & 4 \end{pmatrix}$$

$$C^+ = C^*(CC^*)^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}^{-1} = \tfrac{1}{3} \begin{pmatrix} 2 & 1 \\ 1 & 2 \\ -1 & 1 \end{pmatrix}$$

Pro pseudoinverzi matice A pak platí:

$$A^+ = C^+B^+ = \frac{1}{3} \begin{pmatrix} 2 & -1 \\ 1 & 1 \\ -1 & 2 \end{pmatrix} \frac{1}{30} \begin{pmatrix} -5 & 5 & 10 \\ -17 & 5 & 4 \end{pmatrix} = \frac{1}{90} \begin{pmatrix} 7 & 5 & 16 \\ -22 & 10 & 14 \\ -29 & 5 & -2 \end{pmatrix}$$

♠

M-file 3.3. V softwaru Matlab jsem naprogramovala m-file 'pseuoinverze', který za využití skeletového rozkladu nalezne psedoinverzi zadané matice.

```
function S = pseudoinverze(A, eps)
%VSTUP:  A... matice
%        eps... požadovaná přesnost
%VÝSTUP: S...pseudoinverzni matice k matici A
[B,C] = SkeletovyRozklad(A, eps);
D=B'*B;
G=inv(D);
Bp=G*B';
K=C*C';
L=inv(K);
Cp=C'*L;
S=Cp*Bp;
```

♠

Příklad 3.7. Najděte pseudoinverzní matici k matici

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

Vyřešíme tento příklad pomocí m-filu 3.3 tak, že do Matlabu zadáme

```
>> A=[1 2 2 1; -2 1 0 -1; 2 4 -1 -1;2 1 3 2]
>> S = pseudoinverze(A, 10^-16)
```

Matlab nám vrátí výsledek

```
S =

   -2.0000    0.5000    0.5000    1.5000
    1.6667   -0.5000   -0.1667   -1.1667
   -3.0000    1.5000    0.5000    2.5000
    5.6667   -2.5000   -1.1667   -4.1667
```

♠

Poznámka 3.3. V matematickém programu Matlab existuje funkce 'pinv', která vrací psedoinverzní matici k zadané matici. Vyzkoušíme tuto funkci na stejné matici jako v předchozím příkladě 3.7. Do Matlabu tedy zadáme

```
>> A=[1 2 2 1; -2 1 0 -1; 2 4 -1 -1;2 1 3 2]
>> pinv(A)
```

Dostaneme výsledek

```
ans =


   -2.0000    0.5000    0.5000    1.5000
    1.6667   -0.5000   -0.1667   -1.1667
   -3.0000    1.5000    0.5000    2.5000
    5.6667   -2.5000   -1.1667   -4.1667
```

Vidíme, že obě funkce 'pseudoinverze' i 'pinv' vrací shodné matice.

Vlastnosti pseudoinverzních matic se využívají při řešení systému lineárních rovnic s maticí soustavy, která není regulární nebo čtvercová. Důvodem je fakt, že pro každou matici (čtvercovou, obdélníkovou, regulární i singulární) existuje právě jedna Moore - Penroseova inverze. Tato pseudoinverze má několik důležitých a podobných rysů jako inverze čtvercových, regulárních matic.

Uvažujme nyní systém lineárních rovnic $Ax = b$ kde $A \in C_{m,n}$, $x \in \mathbb{C}^n$ a $b \in \mathbb{C}^m$. Vektor $x^* = A^+ b$ se nazývá řešení systému $Ax = b$ vzhledem k nejmenším čtvercům. Řešením ve smyslu metody nejmenších čtverců rozumíme řešení, které minimalizuje normu $\|Ax - b\|$ a mezi všemi takovými vektory je v normě nejmenší. Pokud má systém nekonečně mnoho řešení, pak najdeme to, které je v normě nejmenší. Pokud systém řešení nemá, pak najdeme vektor, který minimalizuje normu reziduí, tj. $\|Ax - b\|$ a mezi všemi takovými vektory je nejmenší.

V následujícím příkladě si najdeme řešení systému lineárních rovnic ve smyslu metody nejmenších čtverců pomocí pseudoinverze.

Příklad 3.8. Najděte řešení systému lineárních rovnic s maticí

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

a vektorem pravé strany

$$b = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$$

Abychom našli řešení ve smyslu metody nejmenších čtverců $x^* = A^+ b$ soustavy $Ax = b$ musíme nejdříve určit pseudoinverzi matice A. Vzhledem k tomu, že již máme vytvořenou funkci 'pseudoinverze', můžeme ji použít tak, že do Matlabu zadáme

```
>> A=[4 3 2 1; 3 1 1 0;2 -1 1 1];
>> S = pseudoinverze(A, 10^-16)
```

Matlab nám vrátí pseudoinverzní matici k matici A

```
S =

  -0.1556    0.4889    0.0444
   0.2815   -0.1704   -0.3185
   0.1852   -0.2963    0.1852
   0.4074   -0.8519    0.4074
```

Nyní stačí provést násobení

$$x^* = Sb = \begin{pmatrix} -0.1556 & 0.4889 & 0.0444 \\ 0.2815 & -0.1704 & -0.3185 \\ 0.1852 & -0.2963 & 0.1852 \\ 0.4074 & -0.8519 & 0.4074 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 0.4222 \\ -0.5259 \\ 0.2593 \\ 0.3704 \end{pmatrix}$$

Vektor x^* je řešením uvažované soustavy ve smyslu metody nejmenších čtverců.

♠

4 Singulární rozklad

Singulární rozklad matice má významné využití v numerické matematice a to především při výpočtu pseudoinverzní matice při řešení systému lineárních rovnic.

Věta 4.1 (Singulární rozklad)**.** *Je-li $A \in C_{m,n}$, $h(A) = r$. Pak existuje unitární matice $U \in C_m$ a unitární matice $V \in C_n$ a diagonální matice $D \in C_r$ s kladnými diagonálními prvky $d_1, d_2, \ldots, d_r$ takovými, že $d_1 \geq d_2 \geq \ldots \geq d_r$ tak, že platí*

$$A = U \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} V^*$$

přičemž nulové bloky doplňují matici D na pravé straně ma matici typu $m \times n$.

Důkaz: viz. [2], strana 62.

Poznámka 4.1. Pro jednoduchost označíme matici $\begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix}$ písmenem S.

Poznámka 4.2. Diagonální prvky matice D z věty 4.1 se nazývají **singulární čísla** matice A a jsou to druhé odmocniny vlastních hodnot matice A^*A.

Nyní si pro lepší porozumění singulárního rozkladu ukážeme příklad, který je zároveň návodem k tomu, jak nalézt matice U a V.

Příklad 4.1. Najděte singulární rozklad matice A

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

Nejprve spočítáme matici AA^T

$$AA^T = \begin{pmatrix} -1 & 1 \\ -1 & 0 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} -1 & -1 & 1 \\ 1 & 0 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 5 \end{pmatrix}$$

Nyní určíme vlastní čísla

$$det(AA^T - \lambda I) = \begin{vmatrix} \lambda - 2 & 1 & 1 \\ 1 & \lambda - 1 & -1 \\ 1 & -1 & \lambda - 5 \end{vmatrix} = \lambda^3 - 8\lambda^2 + 14\lambda$$

Tedy řešením kubické rovnice

$$\lambda^3 - 8\lambda^2 + 14\lambda = 0$$

jsou vlastní čísla

$\lambda_1 = 4 + \sqrt{2}$
$\lambda_2 = 4 - \sqrt{2}$
$\lambda_3 = 0$

Pro λ_1 dostáváme vlastní vektor $u_1 = [0.2326, -0.1645, 0.9586]^T$
pro λ_2 dostáváme vlastní vektor $u_2 = [0.8125, 0.5745, -0.0986]^T$
a pro λ_3 dostáváme vlastní vektor $u_3 = [-0.5345, 0.8018, 0.2673]^T$

Vlastní vektory u_1, u_2, u_3 tvoří sloupce matice U, která tedy vypadá následovně

$$U = \begin{pmatrix} 0.2326 & 0.8125 & -0.5345 \\ -0.1645 & 0.5745 & 0.8018 \\ 0.9586 & -0.0986 & 0.2673 \end{pmatrix}$$

Po odmocnění vlastních čísel λ_1, λ_2 dostáváme singulární čísla, což jsou diagonální prvky matice D. Jelikož matice S musí mít podle věty 4.1 stejné rozměry jako zadaná matice A, přidáme matici D jeden nulový řádek. Tedy

$$S = \begin{pmatrix} 2.3268 & 0 \\ 0 & 1.6080 \\ 0 & 0 \end{pmatrix}$$

Na závěr vypočteme prvky matice V ze vztahu $SV = U^{-1}A$, což lze v Matlabu vyřešit příkazem $V = (U^{-1}A) \backslash S$, tedy.

$$V = \begin{pmatrix} 0.3827 & -0.9239 \\ 0.9239 & 0.3827 \end{pmatrix}$$

Pro ověření, spočítáme součin USV^T

$$USV^T = \begin{pmatrix} 0.2326 & 0.8125 & -0.5345 \\ -0.1645 & 0.5745 & 0.8018 \\ 0.9586 & -0.0986 & 0.2673 \end{pmatrix} \begin{pmatrix} 2.3268 & 0 \\ 0 & 1.6080 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 0.3827 & -0.9239 \\ 0.9239 & 0.3827 \end{pmatrix}^T =$$

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

♠

M-file 4.1. Pro nalezení singulárního rozkladu jsem v Matlabu vytvořila m-file, který najde matice U, S a V

```
function [U,S,V]=singularniRozklad(A)
%VSTUP:  A... matice
%VÝSTUP: U...matice ze singulárního rozkladu matice A
%        S...matice ze singulárního rozkladu matice A
%        V...matice ze singulárního rozkladu matice A

[k,l]=size(A);
u=min(k,l);
t=max(k,l);
S=zeros(k,l);
F=A*A';
H=A'*A;

[D2] = eig(H);
if k>l
    [U,D] = eig(F);
    [VV,D2] = eig(H);
     for i=1:t/2
```

```
        temp = U(:,i);
        U(:,i) = U(:,t-i+1);
        U(:,t-i+1)=temp;
    end
    for i=1:u
        S(i,i) = sqrt(D2(i,i));
    end
    for i=1:u/2
        temp = S(i,i);
        S(i,i) = S(u-i+1,u-i+1);
        S(u-i+1,u-i+1)=temp;
    end
    V=(inv(U)*A)\S;
else
    [UU,D] = eig(F);
    [V,D2] = eig(H);
    for i=1:u
      S(i,i) = sqrt(D(i,i));
    end
    for i=1:u/2
       temp = S(i,i);
       S(i,i) = S(u-i+1,u-i+1);
       S(u-i+1,u-i+1)=temp;
    end
    for i=1:t/2
        temp = V(:,i);
        V(:,i) = V(:,t-i+1);
        V(:,t-i+1)=temp;
```

```
    end
    U=(inv(V)*A')\S';
end
```

♠

Příklad 4.2. Najděte singulární rozklad matice A

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \\ 1 & 3 \end{pmatrix}$$

Vyřešíme tento příklad pomocí m-filu 4.1 tak, že do Matlabu zadáme příkaz

```
>> [U,S,V]=singularniRozklad([1 2;2 1; 1 3])
```

Matlab nám vrátí výsledné matice U, S, V

```
U =

    0.5251    0.0998    0.8452
    0.4397   -0.8821   -0.1690
    0.7286    0.4604   -0.5071


S =

    4.2500         0
         0    1.3920
         0         0


V =

    0.5019   -0.8649
    0.8649    0.5019
```

♠

Poznámka 4.3. V softwaru Matlab se můžeme setkat s příkazem pro výpočet singulárního rozkladu. Zadáme-li příkaz

(i) `[U,S,V] = svd(A)`

tak nám Matlab vrátí diagonální matici S a unitární matice U a V.

(ii) `s = svd(A)`

tak nám Matlab vrátí singulární čísla matice A.

Příklad 4.3. Najděte singulární rozklad matice A

$$A = \begin{pmatrix} 1 & 2 & 3 \\ -1 & 1 & 2 \\ -1 & 3 & 1 \\ 1 & -1 & 4 \end{pmatrix}$$

Vyřešíme tento příklad pomocí příkazu $svd(A)$ tak, že do Matlabu zadáme

```
>> A=[1 2 3;-1 1 2;-1 3 1; 1 -1 4];
>> [U,S,V]=svd(A)
```

Matlab nám vrátí následující matice

```
U =

   -0.6244    0.1363   -0.6604   -0.3943
   -0.3692    0.1817    0.7201   -0.5587
   -0.3277    0.7289    0.1070    0.5915
   -0.6054   -0.6459    0.1840    0.4272
```

```
S =

    5.7449         0         0
         0    3.7405         0
         0         0    1.4161
         0         0         0


V =

   -0.0928   -0.3797   -0.9205
   -0.3474    0.8787   -0.3275
   -0.9331   -0.2894    0.2134
```

♠

Jak už bylo na začátku této kapitoly zmíněno, tak singulární rozklad lze využít například při řešení systému lineárních rovnic, o čemž hovoří následující věta 4.2.

Věta 4.2. *Nechť je dána matice $A \in C_{m,n}$ a její singulární rozklad*

$$A = U \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} V^*$$

Pak matice

$$A^+ = V \begin{pmatrix} D^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^*$$

je pseudoinverzní maticí k matici A.

Důkaz: viz. [2], strana 210.

Poznámka 4.4. Ačkoli rozklad matice A z věty 4.2 nemusí být jednoznačný, tak matice A^+ jednoznačná je. Opět budeme hledat řešení x^* ve smyslu metody nejmenších čtverců.

Uveďme si nyní příklad jak nalézt řešení soustavy lineárních rovnic ve smyslu metody nejmenších čtverců pomocí singulárního rozkladu.

Příklad 4.4. Najděte řešení soustavy ve smyslu metody nejmenších čtverců s maticí

$$A = \begin{pmatrix} 1 & 1 \\ 3 & -1 \\ 0 & 1 \end{pmatrix}$$

a vektorem pravé strany

$$b = \begin{pmatrix} 1 \\ 1 \\ 3 \end{pmatrix}$$

Abychom zjistili řešení soustavy $Ax = b$ ve smyslu metody nejmenších čtverců, budeme postupovat podle věty 4.2. Nejdříve nalezneme singulární rozklad matice A pomocí m-filu 4.1. Tedy zadáme do Matlabu příkaz

```
>> A=[1 1; 3 -1;0 1];
>> [U,S,V]=singularniRozklad(A)
```

a dostaneme následující matice

```
U =


   -0.2187    0.7785    0.5883
   -0.9726   -0.1251   -0.1961
    0.0791    0.6151   -0.7845


S =


    3.2452         0
         0    1.5713
         0         0
```

```
V =

  -0.9665    0.2567
   0.2567    0.9665
```

Matice D je matice S bez posledního nulového řádku, takže vypadá následovně

$$D = \begin{pmatrix} 3.2452 & 0 \\ 0 & 1.5713 \end{pmatrix}$$

Nyní podle věty 4.2 najděme pseudoinverzní matici k matici A. Tedy

$$A^+ = V \begin{pmatrix} D^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^* =$$

$$= \begin{pmatrix} -0.2187 & 0.7785 & 0.5883 \\ -0.9726 & -0.1251 & -0.1961 \\ 0.0791 & 0.6151 & -0.7845 \end{pmatrix} \begin{pmatrix} 0.3081 & 0 & 0 \\ 0 & 0.6364 & 0 \end{pmatrix} \begin{pmatrix} -0.9665 & 0.2567 \\ 0.2567 & 0.9665 \end{pmatrix}^* =$$

$$= \begin{pmatrix} 0.1923 & 0.2692 & 0.0769 \\ 0.4615 & -0.1538 & 0.3846 \end{pmatrix}$$

Na závěr stačí najít řešení zadané soustavy ve smyslu metody nejmenších čtverců, které je ve tvaru

$$x^* = A^+ b$$

Tedy

$$x^* = A^+ b = \begin{pmatrix} 0.1923 & 0.2692 & 0.0769 \\ 0.4615 & -0.1538 & 0.3846 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 0.6922 \\ 1.4615 \end{pmatrix}$$

♠

Závěr

Cílem této bakalářské práce bylo obohatit své vědomosti o základní vlastnosti matic a především vytvořit přehled o jednotlivých typech maticových rozkladů. V této práci jsme ukázali na jakém principu funguje Gaussova eliminační metoda a z ní plynoucí LU rozklad, který se efektivně využívá u soustav lineárních rovnic s více pravými stranami. Dále jsme si také ukázali, že LU rozklad lze využít také pro výpočet determinantu a inverzní matice. Pro řešení soustav lineárních rovnic s hermitovskou maticí jsme se naučili Choleského metodu, která je oproti LU rozkladu či Gaussově eliminační metodě časově méně náročnější. Také jsme si ukázali tzv. Croutovu metodu, která se využívá u řešení soustav s třídiagonální maticí. Skeletový rozklad, který je dalším typem součinového rozkladu, je používán především pro výpočet psedoinverzní matice. Také jsme se zmínili, že při řešení soustav lineárních rovnic se ještě můžeme setkat se singulárním rozkladem. Poznatky o těchto rozkladech jsme demonstrovali na příkladech a v matematickém programu Matlab jsme sestavili m-fily, které by nám měly hledání zmíňovaných rozkladů ulehčit.

Během tvorby této bakalářské práce jsem se blíže seznámila s problematikou numerické matematiky, získala mnoho zkušeností s četbou odborné literatury a zároveň jsem zdokonalila své dovednosti s matematickým softwarem Matlab a s typografickým systémem LaTeX, kterým je práce vysázena.

Věřím, že usilí věnované tvorbě této bakalářské práce bude zajímavým přínosem nejen pro mě v dalším studiu, ale i pro všechny čtenáře této práce.

Literatura

[1] Burden R.L., Faires J.D. : Numerical analysis, Prindle,Weber and Schmidt, Boston, 1984

[2] Fiedler M. : Speciální matice a jejich použití v numerické matematice, SNTL, Praha, 1981

[3] Holenda J. : O maticích, Plzeň, 2007

[4] Horová I., Zelinka J. : Numerické metody, skriptum MU, Brno, 2004

[5] Míka S. : Numerické metody algebry, SNTL,Praha 1982

[6] Segethová J. : Základy numerické matematiky, Karolinum, Praha, 2002

[7] Schatzman M. : Numerical analysis, Oxford university press 2007

[8] Šik F. : Lineární algebra zaměřená na numerickou analýzu, skriptum MU,Brno 1998