

Řešení soustav lineárních rovnic

Matematické algoritmy (11MAG)

Jan Příkryl

11. přednáška 11MAG
pondělí 9. prosince 2013

verze:2013-12-10 15:33

Obsah

1	Soustavy	2
1.1	Formulace úlohy	2
1.2	Existence a jednoznačnost řešení	3
1.3	Podmíněnost soustav lineárních rovnic, odhady chyb	3
1.3.1	Vektorové normy	4
1.3.2	Maticové normy	5
1.3.3	Podmíněnost matic	6
1.3.4	Odhady chyb	7
1.3.5	Reziduum	8
2	Numerické metody	10
2.1	Transformace řešené úlohy	10
2.1.1	Permutace	11
2.1.2	Řádkové škálování	11
2.1.3	Lineární kombinace	11
2.2	Řešení soustav rovnic s trojúhelníkovou maticí	12
2.3	Gaussova eliminační metoda	14
2.4	LU rozklad matice	17
2.5	Algoritmus LU rozkladu	18
2.6	Výběr hlavního prvku	19
3	Dodatky	22
3.1	Numerické chyby při Gaussově eliminaci	22

3.2 Metoda LU rozkladu na počítači	23
3.3 Výpočetní složitost řešení lineárních soustav	23

1 Soustavy lineárních algebraických rovnic

Mnohé matematické a počítačové modely fyzikálních a technických jevů vedou v konečné fázi na řešení soustav **lineárních algebraických rovnic**. Velmi často jsou tyto soustavy rozsáhlé, mají běžně statisíce nebo miliony rovnic a neznámých. Z tohoto důvodu se jak v teoretické, tak v praktické numerické matematice (tvorbě matematického softwaru) tématice numerické lineární algebry věnovala a věnuje značná pozornost a tato oblast je dnes důkladně prozkoumána.

Používají se jak **přímé**, tak **iterační** metody řešení soustav lineárních rovnic. Přímé metody dávají přesné řešení soustavy, alespoň teoreticky, v konečném počtu kroků. Z tohoto hlediska je tedy řešení soustavy lineárních rovnic typická **numerická úloha**. Při realizaci přímé metody na počítači ovšem musíme brát v úvahu také vlastnosti počítačové aritmetiky, především zaokrouhlovací chyby.

Protože velká část používaných počítačových modelů vychází z matematických modelů formulovaných typicky jako okrajové úlohy pro diferenciální rovnice a řešené soustavy lineárních rovnic jsou pouze aproximacemi těchto okrajových úloh, je často zbytečné usilovat o jejich přesné vyřešení. Stačí nám přibližné řešení získané vhodnou iterační metodou s přesností srovnatelnou s přesností aproximace teoretické okrajové úlohy. Takové přibližné řešení se dá iteračními metodami zpravidla získat mnohem úsporněji (uvažte velikost řešených soustav!) než řešení vypočítané přímou metodou.

Ale i v jednotlivých krocích iteračních metod se může přímá metoda vyskytnout, navíc jsou tyto metody velmi užitečné při řešení soustav s maticemi menších rozměrů. Také teorie přímých metod, alespoň ta, kterou vyložíme zde, je podle našeho názoru snazší než je tomu u metod iteračních a patří víceméně ke všeobecnému vzdělání inženýra. V této části kurzu se proto budeme zabývat přímými metodami a ohledně iteračních metod, na které nám již nezbude čas a prostor, čtenáře musíme odkázat na literaturu.

Pro detailnější obeznámení s pojmy, uváděnými níže, doporučuji i zde konzultovat knihu Michaela T. Heatha [4] nebo monografii Nicka Highama [5]. Čtenář může případně použít i některou z českých učebnic či mnoha skript o numerické matematice, která v posledních letech vyšla – například [1, 2, 3] (části skriptu [3] jsou dostupné i on-line). Mnohé ze zde použitých materiálů jsme převzali právě z [4].

1.1 Formulace úlohy

Zopakujeme nejprve stručně formulaci úlohy a některé základní informace o řešení soustav a o jejich maticích, které by čtenář měl již znát z kurzu lineární algebry. V lineární algebře jsme se dozvěděli, že pohodlný způsob, jak vyjádřit lineární zobrazení (transformaci) mezi dvěma konečně-rozměrnými vektorovými prostory, spočívá v použití **matic**. V tomto textu se budeme zabývat pouze soustavami n rovnic o n neznámých, zmíněné vektorové prostory tedy budou mít stejnou dimenzi a příslušné matice budou čtvercové. Poznamenejme, že pod pojmem vektor zde rozumíme vždy *sloupcový* vektor, pokud výslovně neuvedeme jinak. V maticovém zápisu má soustava lineárních algebraických rovnic tvar

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

kde \mathbf{A} je čtvercová matice **řádu** n (nebo také typu $n \times n$), a \mathbf{x}, \mathbf{b} jsou n -složkové vektory.

Taková soustava lineárních rovnic vlastně před nás klade otázku „Je možné vektor \mathbf{b} vyjádřit jako lineární kombinaci sloupců matice \mathbf{A} ?“. Nebo, což je totéž, „Leží vektor \mathbf{b} v prostoru $\text{span}(\mathbf{A}) = \{\mathbf{Ax} : \mathbf{x} \in \mathbb{R}^n\}$?“

Pokud tomu tak je, soustava má zřejmě řešení a nazývá se **konzistentní**. Řešení soustavy může i nemusí existovat, a pokud existuje, může i nemusí být jediné. Detaily jsou studovány v lineární algebře, ale zde se omezíme pouze na soustavy rovnic s regulárními maticemi, kde je otázka existence a jednoznačnosti řešení jednoduše vymezena. Pro jednoduchost se omezíme také pouze na soustavy lineárních algebraických rovnic s reálnými koeficienty, jejichž řešeními jsou pak nutně opět reálné vektory. Ovšem komplexní soustavy lineárních rovnic mohou být zpracovány zcela obdobně.

1.2 Existence a jednoznačnost řešení

Soustavy lineárních algebraických rovnic se čtvercovými regulárními maticemi mají vždy řešení (pro každou pravou stranu \mathbf{b}) a toto řešení je jediné.

Definice 1 (Regulární matice). Čtvercová matice \mathbf{A} řádu n je **regulární**, jestliže vyhovuje kterékoli z následujících ekvivalentních podmínek:

1. k matici \mathbf{A} existuje inverzní matice \mathbf{A}^{-1} (pro takovou matici platí $\mathbf{AA}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$, kde \mathbf{I} je jednotková matice řádu n);
2. $\det(\mathbf{A}) \neq 0$ (tj. determinant matice je nenulový);
3. hodnost matice \mathbf{A} je rovna n (**hodnost matice** je maximální počet lineárně nezávislých sloupců nebo řádků, které matice obsahuje);
4. soustava $\mathbf{Az} = \mathbf{o}$, kde \mathbf{o} je nulový vektor, má pouze triviální řešení (řešením je opět pouze nulový vektor).

Matice, která není regulární, se nazývá **singulární**.

Existence a jednoznačnost řešení soustavy (1) jednoduchým způsobem závisí na regularitě matice \mathbf{A} : Jestliže je matice \mathbf{A} regulární, pak existuje její inverzní matice \mathbf{A}^{-1} a soustava (1) má vždy právě jedno řešení $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ nezávisle na hodnotě pravé strany \mathbf{b} . Jestliže je naproti tomu matice \mathbf{A} singulární, pak je řešitelnost soustavy (1) závislá na pravé straně \mathbf{b} ; pro některou pravou stranu nemusí mít soustava žádné řešení, ale pokud existuje alespoň jedno řešení, pak jich existuje nekonečně mnoho. Řešení tedy v takovém případě není určeno jednoznačně.

Geometricky ve dvou dimenzích každá z rovnic soustavy 2×2 představuje rovnici přímky a řešením soustavy je tedy společný bod obou přímek. Jsou-li tyto dvě přímky různoběžné, mají právě jeden průsečík, řešení existuje a je jediné (regulární případ), Jsou-li obě přímky rovnoběžky, pak buď nemají průsečík vůbec nebo splývají a pak je zde nekonečně mnoho společných bodů, řešení (singulární případ).

1.3 Podmíněnost soustav lineárních rovnic, odhady chyb

Poté, co jsme zhruba popsali otázky existence a jednoznačnosti řešení lineárních soustav, se nyní zaměříme na citlivost řešení \mathbf{x} na poruchy ve vstupních datech řešené numerické úlohy, což jsou

zde matice \mathbf{A} a vektor pravých stran \mathbf{b} . Abychom takové vektorové a maticové poruchy mohli měřit, budeme potřebovat pro vektory a matice nějak zobecnit pojem velikosti, jako který nám u reálných a komplexních čísel slouží jejich absolutní hodnota. Příslušným zobecněním, které by již čtenář mohl znát z kurzu lineární algebry, je pojem **normy**. Pro jistotu zde příslušné informace uvedeme.

1.3.1 Vektorové normy

Budeme zde používat vektorové normy, které jsou vesměs speciálními případy tzv. p -norm, které jsou pro kladné celé číslo $p > 0$ a n -složkový vektor \mathbf{x} definovány jako

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}.$$

Důležité speciální případy představují

- *jedničková norma*

$$\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|,$$

- *euklidovská norma*

$$\|\mathbf{x}\|_2 = \left(\sum_{i=1}^n |x_i|^2 \right)^{1/2},$$

která odpovídá obvyklému pojmu vzdálenosti v euklidovských prostorech (nebo chcete-li, délce vektoru), a

- ∞ -norma (čteno nekonečno norma)

$$\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|,$$

která je limitní hodnotou p -normy při $p \rightarrow \infty$ a říká se jí také *Čebyševova norma*.

Při práci se všemi těmito normami dostáváme kvalitativně podobné výsledky, ovšem jednotlivé normy se můžou analyticky nebo výpočetně zpracovávat snáze nebo hůře. Při analýze podmíněnosti soustav se většinou pracuje s jedničkovou nebo Čebyševovou normou. Euklidovská norma se v této souvislosti používá méně, ale nachází účelnou aplikaci v jiných oblastech numerické lineární algebry.

Dá se ukázat, že při pevně daném n se libovolné dvě z uvedených norem liší nanejvýš nevelkou multiplikativní konstantou, která sice závisí na n , ale nezávisí na konkrétním měřeném vektoru. Tím pádem jsou všechny tři normy ekvivalentní v tom smyslu, že pokud je jedna z nich malá, jsou úměrně malé i ty zbývající. V dané situaci tedy můžeme při našich úvahách volit tu z norem, s níž se nám pohodlně pracuje. Z tohoto důvodu budeme v další části tohoto textu index u norem používat pouze tam, kde to bude hrát nějakou roli, a tam, kde bude jedno, která norma se použije, budeme index vynechávat.

Každá vektorová p -norma má pro libovolné dva vektory \mathbf{x}, \mathbf{y} tyto důležité vlastnosti:

1. $\|\mathbf{x}\| \geq 0$ a $\|\mathbf{x}\| = 0$ právě pro $\mathbf{x} = \mathbf{o}$.

2. $\|\alpha \mathbf{x}\| = |\alpha| \cdot \|\mathbf{x}\|$ pro každý skalár (číslo) α .
3. $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ (trojúhelníková nerovnost).

Pro výpočet p -normem vektorů je v Matlabu k dispozici funkce `norm`.

1.3.2 Maticové normy

Budeme potřebovat také nějaký způsob, jak posuzovat „velikost“ matic. Jsou zde sice možné rozličné definice normem, ale ty normy, které budeme používat, budou definovány prostřednictvím již zavedených vektorových normem.

Definice 2 (Maticová norma matice \mathbf{A}). Při dané vektorové normě budeme maticovou normu matice \mathbf{A} definovat jako

$$\|\mathbf{A}\| = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|}.$$

O takové maticové normě se říká, že je **indukovaná** příslušnou vektorovou normou nebo že je této normě **podřízená**. Intuitivně se na tuto definici můžeme dívat tak, že norma matice měří to, jak transformace danou maticí může maximálně protáhnout ten který vektor, když délku vektorů měříme použitou vektorovou normou.

S některými maticovými normami se pracuje snáze než s jinými.

Příklad 3 (Maticová norma odpovídající jedničkové vektorové normě). Tato norma se rovná jednoduše maximální hodnotě sloupcových součtů prvků matice bráných v absolutních hodnotách,

$$\|\mathbf{A}\|_1 = \max_j \sum_{i=1}^n |a_{ij}|.$$

Příklad 4 (Maticová norma odpovídající Čebyševově vektorové normě). Maticová norma odpovídající Čebyševově vektorové normě je prostě maximální řádkový součet prvků matice bráných v absolutních hodnotách,

$$\|\mathbf{A}\|_\infty = \max_i \sum_{j=1}^n |a_{ij}|.$$

Maticová norma odpovídající euklidovské vektorové normě se dá definovat pomocí vlastních čísel matic, její explicitní výpočet je tudíž složitější a my se jí zda zabývat nebudeme.

Takto definované maticové normy mají pro libovolné matice \mathbf{A}, \mathbf{B} tyto důležité vlastnosti:

1. $\|\mathbf{A}\| \geq 0$ a $\|\mathbf{A}\| = 0$ právě když $\mathbf{A} = \mathbf{O}$ (nulová matice).
2. $\|\alpha \mathbf{A}\| = |\alpha| \cdot \|\mathbf{A}\|$ pro každý skalár (číslo) α .
3. $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$.
4. $\|\mathbf{A}\mathbf{B}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{B}\|$.
5. $\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{x}\|$ pro každý vektor \mathbf{x} .

Poznamenejme, že obecně se za maticovou normu považuje jakákoli funkce matic, která má první tři právě uvedené vlastnosti. Normy, které mají navíc poslední dvě vlastnosti, se pak nazývají **submultiplikativní** nebo **konzistentní**. Zdůrazňujeme znovu, že maticové normy indukované vektorovými p -normami jsou s nimi vždy konzistentní, mají tedy všech pět výše uvedených vlastností.

Námi popsané maticové normy se v Matlabu dají snadno počítat pomocí funkce `norm`.

1.3.3 Podmíněnost matic

Číslo podmíněnosti regulární čtvercové matice \mathbf{A} se v dané maticové normě definuje jako

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

Pro singulární matici \mathbf{A} se úmluvou klade $\text{cond}(\mathbf{A}) = \infty$. Při hlubším studiu se dá ukázat, že takto zavedené číslo podmíněnosti je konzistentní s definicí podanou v Přednášce 8, a to v tom smyslu, že ohraničuje poměr mezi relativní změnou řešení soustavy lineárních rovnic a danou relativní změnou ve vstupních datech.

Z definice je zřejmé, že hodnota čísla podmíněnosti závisí na použité maticové normě, což se někdy označuje tím, že se použije příslušný index; píšeme pak například $\text{cond}_1(A)$ nebo $\text{cond}_\infty(A)$.

Z toho, co jsme říkali v Odstavci 1.3.1 ale vyplývá, že hodnoty čísla podmíněnosti se při různých normách mohou lišit maximálně nevelkou multiplikativní konstantou (která závisí na n) a jsou tedy jako měřítka podmíněnosti stejně užitečné. Jako zajímavost uvádíme, že se dá ukázat, že číslo podmíněnosti také měří poměr mezi maximálním relativním protažením a maximálním relativním zkrácením libovolných dvou n -složkových vektorů poté, co na ně aplikujeme matici \mathbf{A} . Z definice čísla podmíněnosti matic lze vcelku snadno odvodit následující jeho důležité vlastnosti, které platí při použití jakékoli námi zavedené normy:

1. Pro všechny matice \mathbf{A} platí $\text{cond}(\mathbf{A}) \geq 1$.
2. Pro jednotkovou matici je $\text{cond}(\mathbf{I}) = 1$.
3. Pro jakoukoli matici \mathbf{A} a každé číslo α různé od nuly je $\text{cond}(\alpha\mathbf{A}) = \text{cond}(\mathbf{A})$.
4. Pro každou diagonální matici $\mathbf{D} = \text{diag}(\mathbf{d})$ platí $\text{cond}(\mathbf{D}) = (\max_i |d_i|) / (\min_j |d_j|)$.

Číslo podmíněnosti je mírou toho, jak blízko je daná matice k tomu, aby byla singulární: matice s velkým číslem podmíněnosti (co to znamená, řekneme v následujícím odstavci) je téměř singulární, zatímco matice, jejíž číslo podmíněnosti je řádu jednotek, má k singularitě daleko. Z definice je také zřejmé, že regulární matice a její inverze mají stejné číslo podmíněnosti.

Všimněme si také toho, že přestože determinant singulární matice je nulový, není hodnota determinantu sama o sobě měřítkem toho, jak blízko nebo daleko má daná matice \mathbf{A} k singulární matici. Vezměme si jako příklad matici $0,1\mathbf{I}$, jejíž determinant je roven $0,1^n$, což už u malé matice s $n = 30$ dává jako hodnotu determinantu 10^{-30} , tedy velmi malé číslo. Přitom je tato matice velmi dobře podmíněná, její číslo podmíněnosti je 1 v jakékoli normě.

Brzy uvidíme, že užitečnost čísla podmíněnosti se ukáže při posuzování přesnosti řešení soustav lineárních rovnic. V definici čísla podmíněnosti ovšem vystupuje inverzní matice, takže počítat

jeho hodnotu by byla obvykle netriviální záležitost. Ve skutečnosti by výpočet čísla podmíněnosti z jeho definice vyžadoval podstatně více výpočetní práce než samotné řešení posuzované soustavy lineárních rovnic. V praxi lze naštěstí jako vedlejší produkt při řešení soustav získat alespoň dobrý odhad čísla podmíněnosti, souhlasící s jeho skutečnou hodnotou přinejmenším řádově. V Matlabu je pro tento účel k dispozici funkce `cond` a některé další funkce.

Příklad 5 (Odhad čísla podmíněnosti). Uvažujme matici

$$\mathbf{A} = \begin{bmatrix} 0,913 & 0,659 \\ 0,457 & 0,330 \end{bmatrix}.$$

Odhadneme její čísla podmíněnosti v Matlabu; použijeme k tomu funkci `cond`. Dostaneme

$$\text{cond}_1(\mathbf{A}) = \text{cond}_\infty(\mathbf{A}) \approx 1,6958 \cdot 10^4,$$

$$\text{cond}_2(\mathbf{A}) \approx 1,2485 \cdot 10^4.$$

Odhady se řádově shodují. Vzhledem k nevelkému řádu matice a počtu uvedených platných cifer můžeme danou matici považovat za špatně podmíněnou. Jak uvidíme v příštím odstavci, pokud by prvky této matice byly naměřeny s chybou velikosti řádově 10^{-4} , nemohli bychom při řešení soustav s touto maticí zaručit ve výsledku ani jednu platnou číslici.

1.3.4 Odhady chyb

Kromě toho, že je číslo podmíněnosti spolehlivým indikátorem blízkosti k singularní matici, nám dává také užitečné kvantitativní odhady chyby pro vypočítaná řešení lineárních soustav. Uvedeme zde pouze výsledky, jakkoli matematika za nimi skrytá není nikterak obtížná. Případné zájemce v tomto směru odkazujeme například na [4, 5].

Uvažujme nejprve poruchy v pravé straně soustavy lineárních rovnic. Nechť \mathbf{x} je řešení regulární soustavy lineárních rovnic $\mathbf{A}\mathbf{x} = \mathbf{b}$ a nechť $\hat{\mathbf{x}}$ je řešením soustavy $\mathbf{A}\hat{\mathbf{x}} = \mathbf{b} + \Delta\mathbf{b}$ s porušenou pravou stranou. Označme $\Delta\mathbf{x} = \hat{\mathbf{x}} - \mathbf{x}$. Pak se dá odvodit odhad

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \text{cond}(\mathbf{A}) \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}.$$

Číslo podmíněnosti matice je tedy jakýsi „amplifikační faktor“, který pomáhá odhadnout maximální relativní změnu řešení způsobenou danou relativní poruchou ve vektoru pravých stran (porovnejte to s obecně zavedeným pojmem čísla podmíněnosti z Přednášky 8).

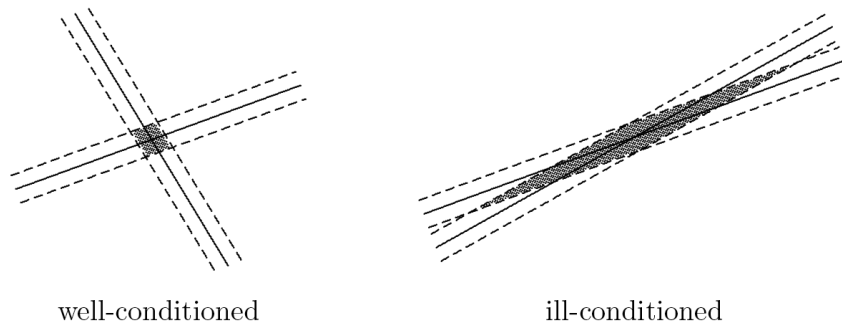
Podobný výsledek se dá odvodit pro relativní změny v prvcích matice \mathbf{A} . Jestliže $\mathbf{A}\mathbf{x} = \mathbf{b}$ a $(\mathbf{A} + \mathbf{E})\hat{\mathbf{x}} = \mathbf{b}$, pak

$$\frac{\|\Delta\mathbf{x}\|}{\|\hat{\mathbf{x}}\|} \leq \text{cond}(\mathbf{A}) \frac{\|\mathbf{E}\|}{\|\mathbf{A}\|}.$$

Obecně se pak dá říci, že při současných změnách prvků matice i pravé strany platí (až na veličiny vyšších řádů)

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \text{cond}(\mathbf{A}) \left(\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|} + \frac{\|\mathbf{E}\|}{\|\mathbf{A}\|} \right).$$

Vidíme tedy opět, že relativní změna řešení je ohraničená součinem čísla podmíněnosti a relativní změny v datech úlohy.



Obrázek 1: Dobře podmíněná (vlevo) a špatně podmíněná (vpravo) soustava dvou lineárních rovnic o dvou neznámých.

Geometrická interpretace popsaných výsledků ve dvou dimenzích je ta, že jsou-li přímky definované dvěma rovnicemi téměř rovnoběžné, pak pokud mohou být tyto přímky zatíženy zao-krouhlovacími chybami nebo chybou měření, není jejich průsečík definován ostře. Pokud ale tyto přímky vůbec nejsou rovnoběžné, jsou třeba téměř na sebe kolmé, je jejich průsečík relativně ostře vymezen. Oba tyto případy ilustrujeme na Obrázku 1, kde čárkované přímky vyznačují oblast nepřesnosti pro každou plně vytaženou přímku, takže průsečík skutečných přímek (které přesně neznáme) by mohl být kdekoli ve vystínovaném rovnoběžníku. Velké číslo podmíněnosti je tedy spojeno s velkou nepřesností získaného řešení.

Máme-li tedy shrnout to, jsme se zatím naučili, pak v případě, že jsou vstupní data přesná na strojovou přesnost (relativní chyba v Matlabu tedy cca 10^{-16}), můžeme za rozumný odhad relativní chyby ve vypočítaném řešení považovat

$$\frac{\|\hat{\mathbf{x}} - \mathbf{x}\|}{\|\mathbf{x}\|} \lesssim \text{cond}(\mathbf{A}) \epsilon_{\text{mach}}$$

kde ϵ_{mach} je relativní přesnost aritmetiky počítače (angl. *unit roundoff*, definici a výpočet máte popsány v podkladech pro 8. cvičení a také v [4] ■ **dopsat do přednášky** ■). Jednoduchý způsob interpretace získaných výsledků je ten, že vypočítané řešení soustavy ztrácí během řešení zhruba $\log_{10}(\text{cond}(\mathbf{A}))$ desítkových číslic přesnosti vůči přesnosti vstupních dat. Podíváme-li se na matici z Příkladu 5, vidíme, že její číslo podmíněnosti je řádově 10^4 , takže při řešení soustavy dvou rovnic s touto maticí nemůžeme v výsledku očekávat jedinou správnou číslici, pokud prvky matice nejsou přesné na více než čtyři platné číslice a pokud řešení nepočítáme v aritmetice používající alespoň čtyři platné desítkové číslice.

Jako kvantitativní míra citlivosti tak číslo podmíněnosti matice hraje při řešení soustav lineárních rovnic stejnou roli, jako tomu bylo u obecně zavedeného čísla podmíněnosti v Přednášce 8. Důležitý rozdíl je zde ale v tom, že maticové číslo podmíněnosti nemůže nikdy být menší než 1.

1.3.5 Reziduum

Jedním ze způsobů, jak ověřit řešení rovnice, je dosadit je do řešené rovnice a podívat se, jak se po dosazení shoduje její levá a pravá strana. **Reziduum** přibližného řešení $\hat{\mathbf{x}}$ lineární soustavy rovnic $\mathbf{Ax} = \mathbf{b}$ je rozdíl

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\hat{\mathbf{x}}.$$

Pokud je \mathbf{A} regulární, platí pro chybu teoreticky, že $\|\Delta\mathbf{x}\| = \|\hat{\mathbf{x}} - \mathbf{x}\| = 0$ tehdy a jen tehdy, je-li $\|\mathbf{r}\| = 0$. V praxi však nemusí být obě tyto veličiny malé současně. Všimněme si především toho,

že vynásobíme-li soustavu $\mathbf{Ax} = \mathbf{b}$ libovolnou nenulovou konstantou, její řešení se nezmění, ale reziduum samé se násobí stejným číslem. Reziduum tedy můžeme udělat libovolně velké nebo malé podle toho, jakým faktorem soustavu vynásobíme (přeškálujeme ji, změním měřítko), a tím pádem je velikost rezidua nesmyslné kritérium, pokud ji nějak nevztáhneme k velikosti dat úlohy a řešení. Z tohoto důvodu se zavádí pojem **relativní reziduum**, které se definuje jako

$$R(\hat{\mathbf{x}}) = \frac{\|\mathbf{r}\|}{\|\mathbf{A}\| \cdot \|\hat{\mathbf{x}}\|}.$$

Abychom uvedli relativní reziduum do vztahu s chybou získaného přibližného řešení, všimneme si, že

$$\|\Delta\mathbf{x}\| = \|\hat{\mathbf{x}} - \mathbf{x}\| = \|\mathbf{A}^{-1}(\mathbf{A}\hat{\mathbf{x}} - \mathbf{b})\| = \|\mathbf{A}^{-1}\mathbf{r}\| \leq \|\mathbf{A}^{-1}\| \cdot \|\mathbf{r}\|.$$

Vydělíme-li obě strany získané nerovnosti $\|\hat{\mathbf{x}}\|$ a použijeme definici čísla podmíněnosti matice \mathbf{A} , dostáváme odsud odhad

$$\frac{\|\Delta\mathbf{x}\|}{\|\hat{\mathbf{x}}\|} \leq \text{cond}(\mathbf{A}) \frac{\|\mathbf{r}\|}{\|\mathbf{A}\| \cdot \|\hat{\mathbf{x}}\|} = \text{cond}(\mathbf{A}) \cdot R(\hat{\mathbf{x}}).$$

Malé relativní reziduum tedy znamená malou relativní chybu vypočítaného řešení tehdy a jen tehdy, když je \mathbf{A} dobře podmíněná matice.

Pro algoritmy přímých metod řešení soustav lineárních rovnic se dá ukázat, že jimi vypočítané řešení $\hat{\mathbf{x}}$ je *přesným* řešením nějaké porušené soustavy

$$(\mathbf{A} + \mathbf{E})\hat{\mathbf{x}} = \mathbf{b}.$$

Popravdě řečeno je takových porušených soustav celá řada, stačí si uvědomit, že při daných \mathbf{A} a $\hat{\mathbf{x}}$ máme ve výše uvedeném vztahu určit n^2 prvků matice \mathbf{E} z n lineárních rovnic. Ukazuje se však, že mezi poruchami \mathbf{E} se dá vybrat porucha s minimální maticovou normou, a té se pak říká *zpětná chyba* vypočítaného řešení (na rozdíl od *přímé chyby* $\Delta\mathbf{x}$). Odhady zpětné chyby jsou pro algoritmy přímých metod k dispozici a můžeme u nich tedy říci, že jimi vypočítané numerické řešení je přesným řešením původní soustavy porušené o poruchu, jejíž velikost (v normě) umíme odhadnout shora.

Výše provedené úvahy o relativním reziduu nám snadno ale dávají dolní odhad takové zpětné chyby. Je totiž

$$\|\mathbf{r}\| = \|\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}\| = \|\mathbf{E}\hat{\mathbf{x}}\| \leq \|\mathbf{E}\| \cdot \|\hat{\mathbf{x}}\|,$$

odkud plyne po vydělení normou \mathbf{A} nerovnost

$$\frac{\|\mathbf{r}\|}{\|\mathbf{A}\| \cdot \|\hat{\mathbf{x}}\|} \leq \frac{\|\mathbf{E}\|}{\|\mathbf{A}\|}.$$

Velké relativní reziduum tak implikuje velkou zpětnou chybu, což znamená, že algoritmus použitý k výpočtu řešení nebyl stabilní a získané řešení nemůže být kvalitní. Na druhé straně, stabilní algoritmy nutně produkují malá relativní rezidua bez ohledu na podmíněnost úlohy, a tím pádem malé reziduum vypovídá o kvalitě získaného řešení jen málo.

Příklad 6 (Malé reziduum). Uvažujme soustavu lineárních rovnic

$$\mathbf{Ax} = \begin{bmatrix} 0,913 & 0,659 \\ 0,457 & 0,330 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0,254 \\ 0,127 \end{bmatrix} = \mathbf{b},$$

s jejíž maticí jsme se již setkali v Příkladu 5. Řekněme, že máme k dispozici dvě přibližná řešení

$$\hat{\mathbf{x}}_1 = \begin{bmatrix} -0,0827 \\ 0,5 \end{bmatrix} \quad \text{a} \quad \hat{\mathbf{x}}_2 = \begin{bmatrix} 0,999 \\ -1,001 \end{bmatrix}.$$

Normy odpovídajících reziduí jsou

$$\|\mathbf{r}_1\|_1 = 2,1 \cdot 10^{-4} \quad \text{a} \quad \|\mathbf{r}_2\|_1 = 2,4 \cdot 10^{-2}.$$

Které z obou řešení tedy máme považovat za přesnější? Díky mnohem menšímu reziduu bychom mohli mít sklon k tomu, abychom řekli, že je to $\hat{\mathbf{x}}_1$. Ale přesné řešení této soustavy je (dá se to snadno ověřit) vektor

$$\mathbf{x} = \begin{bmatrix} 1 \\ -1 \end{bmatrix},$$

takže $\hat{\mathbf{x}}_2$ je ve skutečnosti mnohem přesnější než $\hat{\mathbf{x}}_1$. Důvodem této možná překvapivé skutečnosti je, že matice \mathbf{A} je špatně podmíněná, jak jsme ostatně viděli v Příkladu 5. Díky jejímu velkému číslu podmíněnosti zde malé reziduum neznamená, že máme přesné řešení.

2 Numerické metody pro řešení soustav lineárních rovnic

Přímé numerické metody pro řešení soustav lineárních algebraických rovnic, kterými se nyní budeme zabývat, nejsou vlastně ničím jiným, než zobecněním postupů známých ze střední školy. Tam jsme u soustavy dvou rovnic o dvou neznámých tím či oním způsobem soustavu upravili: z jedné rovnice jsme vyloučili (eliminovali) druhou neznámou a převedli ji tak na rovnici pro jedinou neznámou. Tuto rovnici jsme vyřešili, výsledek dosadili do některé z původních rovnic a z ní pak vypočítali zbývající neznámou. Ukazuje se, že všechny takové úpravy se dají přehledně zapsat ve formě maticových transformací. Numerické metody, které zde budeme popisovat, vlastně provádějí na danou soustavu (její matici) určitou posloupnost úprav, které se dají chápat jako maticové transformace a jejichž cílem je danou soustavu převést na soustavu jednodušší (s maticí speciálního tvaru), která je s původní soustavou ekvivalentní, ale řeší se podstatně přímočařeji.

2.1 Transformace řešení úlohy

Prozatím zde pouze popíšeme maticové transformace, které nemění řešení soustavy, a ukážeme, že k nim mimo jiné patří ty transformace, které si lze představit jako elementární operace s rovnicemi soustavy, speciálně jako vynásobení rovnice nenulovou konstantou nebo jako přičtení násobku jedné rovnice k rovnici jiné. Naším cílem bude posléze převést řešenou soustavu (její matici) ne takový tvar, v němž ji půjde už snadno a přímo vyřešit. Abychom nebyli příliš tajemní, poznamenáváme již zde, že naším cílem bude získat ekvivalentní soustavu, jejíž matice bude trojúhelníková.

Je především jasné, že pokud obě strany soustavy $\mathbf{Ax} = \mathbf{b}$ s regulární maticí vynásobíme zleva jakoukoli regulární maticí \mathbf{M} , na řešení soustavy to nebude mít vliv. Abychom se o tom přesvědčili, všimněme si, že řešení transformované soustavy

$$\mathbf{MAz} = \mathbf{Mb}$$

je dáno jako

$$\mathbf{z} = (\mathbf{MA})^{-1}\mathbf{Mb} = \mathbf{A}^{-1}\mathbf{M}^{-1}\mathbf{Mb} = \mathbf{A}^{-1}\mathbf{b} = \mathbf{x}.$$

2.1.1 Permutace

Důležitým příkladem regulární transformace je skutečnost, že řádky matice \mathbf{A} a odpovídající složky ve vektoru \mathbf{b} lze přeházet, aniž bychom změnili řešení \mathbf{x} . Intuitivně je to zřejmé: všechny rovnice soustavy musí být splněny současně, takže na pořadí, v jakém je zapíšeme, nezáleží. Formálně se takové přehazování řádků dá v maticovém tvaru zapsat jako násobení obou stran soustavy zleva **permutační maticí** \mathbf{P} , což je čtvercová matice (opět řádu n), mající v každém řádku a sloupci právě jednu jedničku a všechny ostatní prvky rovné nule (je to vlastně jednotková matice s přeházenými řádky a sloupci). Podívejme se, jak permutační matice může například přeházet řádky sloupcového vektoru:

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} v_3 \\ v_1 \\ v_2 \end{bmatrix}.$$

Permutační matice je vždy regulární; ve skutečnosti je její inverzí prostě její transpozice, $\mathbf{P}^{-1} = \mathbf{P}^T$. Pro pořádek poznamenejme, že *transpozicí* matice \mathbf{M} , kterou značíme \mathbf{M}^T , rozumíme matici, jejímiž sloupci jsou řádky matice \mathbf{M} . Je-li tedy $\mathbf{N} = \mathbf{M}^T$, platí $n_{ij} = m_{ji}$.

2.1.2 Řádkové škálování

Jiným jednoduchým, ale důležitým případem regulární maticové transformace je **diagonální škálování**. Připomínáme, že matice \mathbf{D} je **diagonální**, jestliže v ní pro všechna $i \neq j$ platí $d_{ij} = 0$, což znamená, že jedinými prvky, které v ní mohou být nenulové, jsou prvky na *hlavní diagonále* $d_{ii}, i = 1, \dots, n$. Determinantem diagonální matice je součin jejích prvků na hlavní diagonále, takže regulární diagonální matice má na hlavní diagonále pouze nenulové prvky.

Vynásobíme-li obě strany soustavy $\mathbf{Ax} = \mathbf{b}$ zleva regulární diagonální maticí,

$$\mathbf{DAx} = \mathbf{Db}$$

je to totéž, jako bychom každý z řádků matice soustavy a pravé strany vynásobili odpovídajícím diagonálním prvkem d_{ii} , a mluvíme zde proto o **řádkovém škálování**. Řádky, kterým v použité diagonální matici odpovídají na hlavní diagonále jedničky, se přitom nemění. Řádkové škálování může být v praxi někdy užitečné, protože nemění řešení soustavy a přitom může pozitivně ovlivnit chování použitého algoritmu na počítači.

2.1.3 Lineární kombinace řádků matice

Další ekvivalentní úprava soustavy lineárních rovnic spočívá v tom, že provedeme určitou lineární kombinaci řádků soustavy a přičteme ji k některé jiné rovnici. Speciální případ, který zde popíšeme, protože hraje významnou roli v numerických metodách řešení soustav, je ten, kdy určitý násobek jednoho řádku matice přičteme k jinému řádku (nebo jej od něj odečteme).

Ukážeme, že taková operace se dá opět zapsat jako transformace jistou regulární maticí, takže nemění řešení soustavy (koneckonců se to zdá být zřejmé intuitivně). Především je vhodné si zopakovat definici násobení matic a uvědomit si, že prvek na místě ij ve výsledné matici není nic jiného než skalární součin i -tého řádku prvního činitele s j -tým sloupcem činitele druhého. Pak

už bychom měli snadno pochopit, že přičtení m -násobku řádku i v dané matici k jejímu j -tému řádku se dá maticově zapsat tak, že danou matici vynásobíme maticí \mathbf{M} , která se od jednotkové matice liší pouze tím, že má navíc na místě ji číslo m . Matice M je tedy trojúhelníková matice, která má například pro $i < j$ tvar podobný

$$\mathbf{M} = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & m & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

Jestliže je v uvedené matici prvek m ve sloupci k a řádku $k + 1$, pak \mathbf{MA} realizuje přičtení m -násobku řádku k v matici \mathbf{A} k řádku $k + 1$ v téže matici.

Příklad 7 (Lineární kombinace řádků matice). Mějme transformační matici \mathbf{M} a matici \mathbf{A} zadané jako

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & m & 1 \end{bmatrix} \quad \text{a} \quad \mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & 1 \\ -1 & 2 & 1 \end{bmatrix}.$$

Potom

$$\mathbf{MA} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & 1 \\ -1 + 0m & 2 - 1m & 1 + 1m \end{bmatrix}$$

Matice podobného typu (třeba i s více nenulovými prvky m v některém sloupci) budeme v dalším nazývat **elementární eliminační matice**. Všimněme si, že elementární eliminační matice mají na diagonále samé jedničky, takže jejich determinant je roven jedné a matice jsou regulární. Popsané kombinování řádků soustavy tedy nemění její řešení.

2.2 Řešení soustav rovnic s trojúhelníkovou maticí

První otázka, kterou si při konstrukci přímých metod řešení soustav lineárních rovnic klademe je, jaké soustavy lze řešit nejsnáze. Na ně se pak budeme obecné soustavy snažit ekvivalentními úpravami převést.

Představme si, že máme soustavu $\mathbf{Ax} = \mathbf{b}$, v níž se najde rovnice, která obsahuje jenom jednu neznámou (jednu složku hledaného řešení). To znamená, že v daném řádku matice \mathbf{A} je jedním jeden nenulový prvek. Z takové rovnice se pak dá snadno (prostým dělením) vypočítat příslušná složka řešení. Teď si představme, že v soustavě je ještě jiná rovnice, v níž vystupují pouze dvě neznámé, a že jedna z nich je ta, kterou jsme právě vypočetali. Jestliže nyní do této druhé rovnice dosadíme hodnotu právě vypočítané složky řešení, můžeme z ní opět snadno vypočítat tu druhou. Pokud je struktura matice i nadále podobná, to jest pokud nám v každé rovnici přibude pouze jedna nová neznámá, dají se tak postupně snadno vypočítat všechny složky řešení.

Matice s touto speciální vlastností se říká **trojúhelníková**, a to z důvodu, který bude brzy zřejmý. Protože soustavy lineárních rovnic s trojúhelníkovými maticemi se snadno řeší, jsou právě ony cílem úprav prováděných v přímých numerických metodách na obecných soustavách rovnic.

Pro výpočetní účely se ukazuje jako užitečné zavést trojúhelníkové matice dvou speciálních tvarů. Řekneme, že matice \mathbf{L} je **dolní trojúhelníková**, jestliže všechny její prvky nad hlavní diagonálou jsou nulové (tj. $\ell_{ij} = 0$ pro $i < j$). Podobně řekneme, že matice \mathbf{U} je **horní trojúhelníková**, jestliže všechny její prvky pod hlavní diagonálou jsou nuly (tj. $u_{ij} = 0$ pro $i > j$). Poznamenáváme, že trojúhelníkovou matici definovanou výše v obecném smyslu můžeme vždy permutacemi řádků nebo sloupců převést na dolní nebo horní trojúhelníkovou matici.

Příklad 8 (Horní a dolní trojúhelníková matice).

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}, \quad \text{a} \quad \mathbf{U} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Jakkoli je způsob řešení soustav s trojúhelníkovými maticemi nyní již snad zřejmý, považujeme za užitečné algoritmy pro jejich řešení popsat podrobněji. Proces řešení dolní trojúhelníkové soustavy $\mathbf{L}\mathbf{x} = \mathbf{b}$ postupným dosazováním se nazývá **přímá substitute** a matematicky se dá zapsat jako

$$x_1 = b_1/\ell_{11}, \quad x_i = \left(b_i - \sum_{j=1}^{i-1} \ell_{ij}x_j \right) / \ell_{ii}, \quad i = 2, \dots, n.$$

Formální zápis příslušného algoritmu může vypadat například následujícím způsobem.

Algoritmus 1 Přímá substitute pro soustavu s dolní trojúhelníkovou maticí

```

for  $j = 1$  to  $n$  do {cyklus přes sloupce}
  if  $\ell_{jj} = 0$  then {stop při singulární matici}
    return error
  end if
   $x_j \leftarrow b_j / \ell_{jj}$  {výpočet složky řešení}
  for  $i = j + 1$  to  $n$  do
     $b_i \leftarrow b_i - \ell_{ij}x_j$  {aktualizace pravé strany}
  end for
end for

```

Podobně se u horní trojúhelníkové soustavy $\mathbf{U}\mathbf{x} = \mathbf{b}$ proces řešení postupným dosazováním nazývá **zpětná substitute** a matematicky se dá zapsat jako

$$x_n = b_n/u_n, \quad x_i = \left(b_i - \sum_{j=i+1}^n u_{ij}x_j \right) / u_{ii}, \quad i = n-1, \dots, 1.$$

Uvádíme jeden ze způsobů algoritmické realizace zpětné substitute.

V obou těchto algoritmech jsme zvolili pořadí indexů v cyklech tak, že se k prvkům matic přistupuje po sloupcích (a ne po řádcích) a že operace, kterou provádí vnitřní cyklus, je skalár krát vektor plus vektor, operace zkracovaná **saxpy** (z angl. *scalar times a vector plus a vector*) a používaná v základních knihovnách numerické lineární algebry. Ukládání maticí po sloupcích je typické pro některé programovací jazyky, jako je například Matlab nebo Fortran. Naproti tomu třeba v jazyku C se matice ukládají po řádcích a při větších soustavách by výše popsané algoritmy bylo třeba přepsat tak, aby se v nich k maticím přistupovalo po řádcích. To nepředstavuje zásadní problém, zmiňujeme se o tom pouze proto, abychom čtenáře upozornili, že způsob implementace matematického algoritmu může podstatně ovlivnit jeho efektivitu, a to v závislosti na použitém programovacím jazyku a výpočetním systému. Poznamenáváme ještě, že výskyt

Algoritmus 2 Zpětná substituce pro soustavu s horní trojúhelníkovou maticí

```
for  $j = n$  to 1 do {zpětný cyklus přes sloupce}
  if  $u_{jj} = 0$  then {stop při singulární matici}
    return error
  end if
   $x_j \leftarrow b_j / u_{jj}$  {výpočet složky řešení}
  for  $i = 1$  to  $j - 1$  do
     $b_i \leftarrow b_i - u_{ij}x_j$  {aktualizace pravé strany}
  end for
end for
```

nulového prvku na diagonále bude znamenat selhání uvedených algoritmů, což ovšem není překvapivé, protože trojúhelníková matice s nulovým prvkem na diagonále má nulový determinant a je tedy singulární.

Příklad 9 (Soustava rovnic s trojúhelníkovou maticí). Mějme soustavu rovnic s horní trojúhelníkovou maticí

$$\begin{bmatrix} 1 & 2 & 2 \\ 0 & -1 & -6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -6 \\ 1 \end{bmatrix}.$$

Z poslední rovnice je dáno $x_3 = 1$. Tuto hodnotu pak můžeme dosadit do druhé rovnice a z ní vypočítat $-x_2 - 6 = -6$, $x_2 = 0$. Nakonec do první rovnice dosadíme jak x_3 tak x_2 a vypočítáme z ní $x_1 + 0 + 2 = 3$, $x_1 = 1$.

2.3 Gaussova eliminační metoda

Naší strategií nyní je navrhnout regulární lineární transformaci, která bude převádět danou obecnou soustavu lineárních algebraických rovnic na ekvivalentní soustavu s trojúhelníkovou maticí, již pak budeme moci snadno řešit postupnými substitucemi. Budeme tedy potřebovat transformaci, která by nahrazovala vybrané nenulové prvky dané matice nulami (vyeliminovala je). Uvidíme, že se toho dá dosáhnout prováděním vhodných lineárních kombinací řádků dané matice.

Základem přímých numerických metod pro řešení obecných soustav lineárních algebraických rovnic je tak postup, který nyní popíšeme na příkladu soustavy čtyř rovnic o čtyřech neznámých a který se nazývá **Gaussova eliminační metoda**. Uvažujme tedy soustavu lineárních rovnic

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 &= b_3 \\ a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 &= b_4 \end{aligned}$$

Položíme

$$m_{i1} = a_{i1}/a_{11}, \quad i = 2, 3, 4,$$

a odečteme m_{i1} krát první rovnici od i -té rovnice pro $i = 2, 3, 4$. Dostaneme tak upravenou ekvivalentní soustavu

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\ a'_{22}x_2 + a'_{23}x_3 + a'_{24}x_4 &= b'_2 \\ a'_{32}x_2 + a'_{33}x_3 + a'_{34}x_4 &= b'_3 \\ a'_{42}x_2 + a'_{43}x_3 + a'_{44}x_4 &= b'_4 \end{aligned}$$

kde

$$a'_{ij} = a_{ij} - m_{i1}a_{1j} \quad \text{a} \quad b'_i = b_i - m_{i1}b_1.$$

Vidíme, že neznámá x_1 byla z posledních tří rovnic vylimitována. Protože při eliminaci se čísla m_{i1} násobí první rovnice říká se jim **multiplikátory** (z angl. *multiply*, násobit). Nyní položíme

$$m_{i2} = a'_{i2}/a'_{22}, \quad i = 3, 4,$$

a odečteme m_{i2} krát druhou rovnici od i -té rovnice ($4i = 3, 4$). Výsledkem je soustava

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\ a'_{22}x_2 + a'_{23}x_3 + a'_{24}x_4 &= b'_2 \\ a''_{33}x_3 + a''_{34}x_4 &= b''_3 \\ a''_{43}x_3 + a''_{44}x_4 &= b''_4 \end{aligned}$$

kde

$$a''_{ij} = a'_{ij} - m_{i2}a'_{2j} \quad \text{a} \quad b''_i = b'_i - m_{i2}b'_2.$$

A nakonec položíme

$$m_{i3} = a''_{i3}/a''_{33}, \quad i = 4,$$

a odečteme m_{i3} krát třetí rovnici od rovnice čtvrté. Výsledkem je soustava s horní trojúhelníkovou maticí

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 &= b_1 \\ a'_{22}x_2 + a'_{23}x_3 + a'_{24}x_4 &= b'_2 \\ a''_{33}x_3 + a''_{34}x_4 &= b''_3 \\ a'''_{44}x_4 &= b'''_4 \end{aligned}$$

kde

$$a'''_{ij} = a''_{ij} - m_{i3}a''_{3j} \quad \text{a} \quad b'''_i = b''_i - m_{i3}b''_{33}.$$

Protože má tato soustava horní trojúhelníkovou matici, lze ji snadno vyřešit zpětnou substitucí. Postupu, kterým jsme původní soustavu lineárních rovnic převedli na ekvivalentní soustavu s trojúhelníkovou maticí se také říká **přímý chod** a řešení výsledné trojúhelníkové soustavy **zpětný chod** Gaussovy eliminace. Prvkům matic, které postupně objevují na hlavní diagonále, tedy v našem případě $a_{11}, a'_{22}, a''_{33}, a'''_{44}$, se říká **hlavní prvky** nebo také **pivoty**.

Popsaný postup je proveditelný pouze, pokud jsou při něm všechny hlavní prvky nenulové. Pokud během eliminace narazíme na nulový hlavní prvek, postupujeme tak, že rovnici s tímto prvkem prohodíme z některou ze zbývajících rovnic, která má v daném sloupci nenulový koeficient. Pokud by ovšem v daném sloupci matice už byly všechny prvky pod diagonálou nulové

(včetně hlavního prvku), není těžké ukázat, že by to znamenalo, že matice řešené soustavy je singularní (a tedy by soustava buď neměla řešení nebo by jich měla nekonečně mnoho).

Gaussovu eliminační metodu ilustrujeme na jednoduchém konkrétním příkladu řešení soustavy tří rovnic o třech neznámých.

Příklad 10 (Gaussova eliminace). Budeme řešit soustavu lineárních rovnic

$$\begin{aligned}x_1 + 2x_2 + 2x_3 &= 3, \\4x_1 + 4x_2 + 2x_3 &= 6, \\4x_1 + 6x_2 + 4x_3 &= 10,\end{aligned}$$

která se dá maticově zapsat jako

$$\mathbf{Ax} = \begin{bmatrix} 1 & 2 & 2 \\ 4 & 4 & 2 \\ 4 & 6 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 10 \end{bmatrix} = \mathbf{b}.$$

Pro účely Gaussovy eliminace je vhodné zapsat data úlohy ve formě tak zvané **rozšířené matice** soustavy

$$\left[\begin{array}{ccc|c} 1 & 2 & 2 & 3 \\ 4 & 4 & 2 & 6 \\ 4 & 6 & 4 & 10 \end{array} \right]$$

a postup Gaussovy eliminace provádět na této matici; je to tak přehlednější. Po prvním kroku eliminace, během kterého odečteme čtyřnásobek prvního řádku rozšířené matice od druhého a třetího řádku, rozšířená matice přejde na

$$\left[\begin{array}{ccc|c} 1 & 2 & 2 & 3 \\ 0 & -4 & -6 & -6 \\ 0 & -2 & -4 & -2 \end{array} \right].$$

Abychom nyní vynulovali člen pod diagonálou v druhém sloupci získané rozšířené matice, odečteme druhý řádek vynásobený jednou polovinou od třetího řádku a dostaneme tak matici

$$\left[\begin{array}{ccc|c} 1 & 2 & 2 & 3 \\ 0 & -4 & -6 & -6 \\ 0 & 0 & -1 & 1 \end{array} \right].$$

To je již rozšířená matice trojúhelníkové soustavy rovnic, z níž můžeme zpětnou substitucí dostat hledané řešení jako

$$\mathbf{x} = \begin{bmatrix} -1 \\ 3 \\ -1 \end{bmatrix}.$$

Algoritmus, který jsme výše obecně popsali pro soustavy čtyř rovnic, se dá snadno zobecnit pro soustavy libovolného řádu. Ukážeme nyní, že výše odvozený postup Gaussovy eliminační metody se dá elegantně zapsat pomocí maticových transformací, v nichž opět figurují trojúhelníkové matice.

2.4 LU rozklad matice

Ukážeme opět na příkladu obecné soustavy 4×4 , že během Gaussovy eliminace vlastně konstruujeme dolní trojúhelníkovou matici \mathbf{L} a horní trojúhelníkovou matici \mathbf{U} takové, že $\mathbf{A} = \mathbf{LU}$.

Položíme $\mathbf{A}_1 = \mathbf{A}$ a dále

$$\mathbf{M}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -m_{21} & 1 & 0 & 0 \\ -m_{31} & 0 & 1 & 0 \\ -m_{41} & 0 & 0 & 1 \end{bmatrix},$$

kde čísla m_{ij} jsou multiplikátory zavedené v předchozím odstavci. Pak se dá snadno ověřit (vzpomeňte si na Odstavec 2.1.3), že platí

$$\mathbf{A}_2 = \mathbf{M}_1 \mathbf{A}_1 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & a'_{32} & a'_{33} & a'_{34} \\ 0 & a'_{42} & a'_{43} & a'_{44} \end{bmatrix},$$

což je matice soustavy, která výše vznikla po prvním kroku Gaussovy eliminační metody. Nyní položíme

$$\mathbf{M}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -m_{32} & 0 & 0 \\ 0 & -m_{42} & 0 & 1 \end{bmatrix}.$$

Opět nyní není těžké ukázat, že

$$\mathbf{A}_3 = \mathbf{M}_2 \mathbf{A}_2 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & a''_{43} & a''_{44} \end{bmatrix},$$

což je matice soustavy, kterou jsme výše obdrželi po provedení druhého kroku eliminace. Konečně položíme

$$\mathbf{M}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -m_{43} & 1 \end{bmatrix}.$$

Potom

$$\mathbf{U} = \mathbf{M}_3 \mathbf{A}_3 = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & 0 & a'''_{44} \end{bmatrix}$$

je horní trojúhelníková matice výsledné soustavy získané Gaussovou eliminační metodou. Jak jsme již poznamenali v Odstavci 2.1.3, matice \mathbf{M}_k , které se během právě popsaného postupu vyskytly, se v literatuře nazývají **elementární eliminační matice**.

Zapíšeme-li vše maticově, vidíme, že pro horní trojúhelníkovou matici \mathbf{U} platí

$$\mathbf{U} = \mathbf{M}_3 \mathbf{M}_2 \mathbf{M}_1 \mathbf{A}.$$

Položíme-li tedy

$$\mathbf{L} = \mathbf{M}_1^{-1} \mathbf{M}_2^{-1} \mathbf{M}_3^{-1},$$

pak

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

Protože inverze dolní trojúhelníkové matice je opět dolní trojúhelníková a protože součin dolních trojúhelníkových matic je opět dolní trojúhelníková matice, je výše sestavená matice \mathbf{L} dolní trojúhelníková matice, a ukázali jsme tak tedy, že Gaussova eliminační metoda vlastně realizuje LU rozklad matice \mathbf{A} na součin dolní a horní trojúhelníkové matice. Této skutečnosti se hojně využívá při realizaci algoritmů Gaussovy eliminace na počítači.

Dosud jsme se nijak nezabývali výpočtem prvků matice \mathbf{L} . Jejich stanovení je však překvapivě jednoduché, neboť se dá ukázat, že matice \mathbf{L} má na hlavní diagonále jedničky a pod diagonálou má v pozici $l_{ij}, i > j$, multiplikátor m_{ij} . Není tak těžké to ověřit. Poznamenáváme opět, že zde popsané úvahy o trojúhelníkovém rozkladu matice nezávisí na řádu matice a dají se snadno zobecnit pro obecné n .

2.5 Algoritmus LU rozkladu

Pokud již máme k dispozici rozklad $\mathbf{A} = \mathbf{LU}$, můžeme snadno převést úlohu řešit soustavu $\mathbf{Ax} = \mathbf{b}$ na řešení dvou soustav rovnic s trojúhelníkovými maticemi. Je to okamžitě vidět z řešené soustavy, kam za \mathbf{A} dosadíme její trojúhelníkový rozklad. Nejprve vyřešíme soustavu

$$\mathbf{Lz} = \mathbf{b},$$

jejíž řešení \mathbf{z} je vlastně eliminací transformovaná pravá strana původní soustavy, a pak soustavu

$$\mathbf{Ux} = \mathbf{z},$$

jejímž řešením je řešení původní soustavy rovnic. Poznamenáváme ještě, že pokud jsme pro danou matici jednou stanovili její trojúhelníkový rozklad, můžeme tímto způsobem řešit s menší pracností i více soustav rovnic, které se liší pouze pravou stranou. To bývá někdy v praxi užitečné.

Zbývá nám pro případné zájemce popsat algoritmus výpočtu matic \mathbf{L} a \mathbf{U} . Na počítači zpravidla prvky těchto matic ukládáme postupně na místa původně obsazená prvky matice \mathbf{A} . Co do počtu míst to vychází, protože víme, že diagonální prvky matice \mathbf{L} jsou jedničky a nemusíme je tedy ukládat. Příslušný algoritmus má řadu variant, uvádíme zde jednu z možností.

Algoritmus 3 LU rozklad Gaussovou eliminací

```
for  $k = 1$  to  $n - 1$  do {cyklus přes sloupce}
  if  $a_{kk} = 0$  then
    return {stop, je-li hlavní prvek nulový}
  end if
  for  $i = k + 1$  to  $n$  do {výpočet multiplikátorů pro daný sloupec}
     $a_{ik} \leftarrow a_{ik}/a_{kk}$ 
  end for
  for  $j = k + 1$  to  $n$  do
    for  $i = k + 1$  to  $n$  do {aplikujeme transformaci na zbývající submatici}
       $a_{ij} \leftarrow a_{ij} - a_{ik}a_{kj}$ 
    end for
  end for
end for
```

2.6 Výběr hlavního prvku

Gaussova eliminační metoda v sobě nese jeden zřejmý problém, o kterém jsme se již zmínili, ale také jeden další, méně zřetelný problém. Možným zřejmým problémem je to, že celý proces selže, pokud se během eliminace vyskytne nulový hlavní prvek, protože by se při výpočtu multiplikátorů v odpovídajícím sloupci dělilo nulou. Ale řešení tohoto problému je téměř stejně zřejmé: pokud je v kroku k eliminace na diagonále nulový hlavní prvek, prohodíme řádek k v soustavě (tedy v její matici i v její pravé straně, jinak řečeno, řádek k v rozšířené matici soustavy) s nějakým z pod ním ležících řádků, jehož prvek ve sloupci k je nenulový. Z Odstavce 2.1.1 již víme, že tím se řešení soustavy nezmění. S nenulovým diagonálním prvkem jako hlavním prvkem pak celý postup pokračuje jako obvykle. Takové přehazování řádků se pak nazývá **výběr hlavního prvku** nebo také **pivotace**.

Příklad 11 (Pivotace a singularita). Případná nutnost provádět výběr hlavního prvku a přehazovat řádky nemá nic společného s tím, zda daná matice je singulární či nikoliv. Tak například matice

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

je zřejmě regulární (spočítejte si determinant), ale pokud nepřehodíme řádky, nemá žádný LU rozklad. Naproti tomu singulární matice

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

má (bez přehazování řádků) LU rozklad

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

Ale co dělat v případě, že na hlavní diagonále ani pod ní ve sloupci k není žádný nenulový prvek? Pak v daném kroku není třeba provádět nic, protože všechny prvky, které bychom měli eliminovat jsou nulové, a přejdeme tedy prostě k dalšímu sloupci (je tedy $\mathbf{M}_k = \mathbf{I}$). Všimněme si toho, že tento krok zanechá na diagonále nulový prvek, takže výsledná matice \mathbf{U} bude singulární, nicméně LU rozklad lze stejně dokončit. Znamená to ovšem, že s takovou maticí selže zpětný chod, protože se v něm vyskytuje dělení každým z diagonálních prvků matice \mathbf{U} , ale to není až tolik překvapivé, protože v takovém případě je nutně singulární už původní matice \mathbf{A} (stačí spočítat si její determinant z determinantů jejích trojúhelníkových faktorů, což jsou součiny jejích diagonálních prvků).

Oním méně zřetelným problémem je to, že v počítačové aritmetice nám na diagonále nemusí vyjít přesná nula, ale pouze nějaké velmi malé číslo, což nás vede k následující úvaze.

Jako hlavní prvek se při výpočtu multiplikátorů dá v zásadě využít jakékoli nenulové číslo, ale v počítačové aritmetice s konečnou přesností bychom se měli zajímat také o to, jak minimalizovat šíření zaokrouhlovacích chyb během výpočtu. Konkrétně je naším cílem omezit velikosti multiplikátorů, takže se již existující zaokrouhlovací chyby vzniklé v předchozích krocích eliminace nebudou zesilovat při následujících transformacích realizovaných elementárními eliminačními maticemi. Absolutní hodnota multiplikátorů zaručeně nikdy nepřevyší jedničku, jestliže v každém sloupci vybereme za hlavní prvek ten z prvků na diagonále nebo pod ní, který má největší absolutní hodnotu. Takové strategii se říká **částečný výběr hlavního prvku** a v praxi je tento výběr pivota podstatný pro to, abychom mohli sestojit numericky stabilní implementaci

Gaussovy eliminace pro obecné matice. Poznamenejme pro pořádek, že slovo *částečný* je zde použito proto, že vhodný hlavní prvek hledáme pouze v prvním sloupci dosud nevyeliminované submatice. Mohli bychom jej vybírat i v celé této submatici, pak bychom ovšem možná museli permutovat i některé sloupce. Navíc tento způsob volby pivota (**úplný výběr hlavního prvku**) nepřináší žádné podstatné výhody.

Ukážeme nyní na extrémním, ale jednoduchém příkladu, jak se může špatná volba pivota při výpočtu projevit. Podobný efekt bychom mohli pozorovat s méně přehnanou volbou pivota u větších soustav či matic.

Příklad 12 (Malé hlavní prvky). Pracujeme-li v počítačové aritmetice, která má pouze konečnou přesnost, musíme se vyhýbat nejen nulovým hlavním prvkům, ale také hlavním prvkům, které jsou *malé* vzhledem k ostatním prvkům daného sloupce, a to proto, abychom zabránili nepřijatelnému růstu chyb. Ukážeme to na velmi zjednodušeném příkladu, kde symbol \approx znamená výsledek počítačové operace. Označíme zde ϵ jakékoli kladné číslo, které je menší než strojové epsilon ϵ_{mach} , tedy jakékoli dostatečně malé číslo x , pro které platí $1 + x \approx 1$. Pokud chce čtenář být konkrétní, může vzít v jednoduché aritmetice $\epsilon = 10^{-8}$ a v dvojnásobné přesnosti, se kterou normálně pracuje Matlab, vzít $\epsilon = 10^{-16}$.

Vezměme nyní matici

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

To je regulární matice, jejíž determinant je zhruba roven -1 a která je dobře podmíněná, má číslo podmíněnosti řádu jednotek. Pokud nepřehodíme řádky, bude hlavním prvkem ϵ , takže výsledný multiplikátor je $-1/\epsilon$ a

$$\mathbf{L} = \begin{bmatrix} 1 & 0 \\ 1/\epsilon & 1 \end{bmatrix}.$$

Jako výsledná horní trojúhelníková matice po eliminaci nám na počítači vyjde

$$\mathbf{U} = \begin{bmatrix} \epsilon & 1 \\ 0 & 1 - 1/\epsilon \end{bmatrix} \approx \begin{bmatrix} \epsilon & 1 \\ 0 & -1/\epsilon \end{bmatrix}.$$

Vynásobíme-li ale nyní matice \mathbf{L} a \mathbf{U} získané na počítači, dostaneme

$$\mathbf{LU} = \begin{bmatrix} \epsilon & 1 \\ 1 & 0 \end{bmatrix} \neq \mathbf{A}.$$

Použití malého prvku a tím pádem velkého multiplikátoru způsobilo velkou ztrátu informace ve výsledné matici \mathbf{U} .

Přehodíme-li řádky, bude nyní hlavní prvek roven 1 a odpovídající multiplikátor bude $-\epsilon$, takže

$$\mathbf{L} = \begin{bmatrix} 1 & 0 \\ \epsilon & 1 \end{bmatrix}.$$

Jako výsledná horní trojúhelníková matice nám na počítači tentokrát vyjde

$$\mathbf{U} = \begin{bmatrix} 1 & 1 \\ 0 & 1 - \epsilon \end{bmatrix} \approx \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

Vynásobením takto získaných matic \mathbf{L} a \mathbf{U} nyní dostaneme

$$\mathbf{LU} = \begin{bmatrix} 1 & 1 \\ \epsilon & 1 \end{bmatrix},$$

což je přesně původní matice \mathbf{A} s přehozenými řádky.

Dá se ukázat, že přehazování řádků během přímého chodu Gaussovy eliminace se dá shrnout jako provedení jedné permutační transformace na původní matici a že tedy na rozdíl od přímého LU rozkladu matice \mathbf{A} , který nemusí vždy existovat (viz výše), ke každé čtvercové matici \mathbf{A} existuje permutační matice \mathbf{P} tak, že k permutované matici lze LU rozklad nalézt, tedy

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

Algoritmus LU rozkladu popsany v předchozím odstavci se při výběru hlavních prvků příliš nemění, je třeba si pouze uvědomit, že nyní místo soustavy $\mathbf{Ax} = \mathbf{b}$ řešíme ekvivalentní soustavu $\mathbf{PAx} = \mathbf{Pb}$. Po získání LU rozkladu matice \mathbf{PA} (matice \mathbf{P} vlastně vzniká teprve během tohoto rozkladu) nejprve řešíme soustavu $\mathbf{Lz} = \mathbf{Pb}$ s dolní trojúhelníkovou maticí přímou substitucí a poté řešíme zpětnou substitucí soustavu $\mathbf{Ux} = \mathbf{z}$ s horní trojúhelníkovou maticí. V numerické lineární algebře se ukazuje, že existují třídy matic, pro které je výběr hlavního prvku zbytečně provádět a kde tedy výše uvedená permutační matice je jednotková. Patří se například pozitivně definitní symetrické matice, s nimiž se často setkáváme v odborné praxi. Pro nedostatek místa zde explicitně algoritmus LU rozkladu s výběrem hlavního prvku neuvádíme.

Příklad 13 (Gaussova eliminace s částečnou pivotací). V Příkladu 10 jsme nepoužívali permutace řádků a v důsledku toho byly některé multiplikátory větší než jedna. Jako ilustraci nyní tento příklad zopakujeme, ale budeme tentokrát používat částečný výběr hlavního prvku.

Soustava, kterou jsme řešili v Příkladu 10, je

$$\mathbf{Ax} = \begin{bmatrix} 1 & 2 & 2 \\ 4 & 4 & 2 \\ 4 & 6 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 10 \end{bmatrix} = \mathbf{b}$$

a její rozšířená matice je

$$\left[\begin{array}{ccc|c} 1 & 2 & 2 & 3 \\ 4 & 4 & 2 & 6 \\ 4 & 6 & 4 & 10 \end{array} \right].$$

Největší prvek v prvním sloupci je 4, takže prohodíme první dva řádky a permutovaná rozšířená matice bude

$$\left[\begin{array}{ccc|c} 4 & 4 & 2 & 6 \\ 1 & 2 & 2 & 3 \\ 4 & 6 & 4 & 10 \end{array} \right].$$

Abychom vynulovali prvky pod diagonálou v prvním sloupci, odečteme od druhého řádku 0,25 krát první řádek a od třetího řádku první řádek krát jedna. jako transformovanou matici tak dostaneme

$$\left[\begin{array}{ccc|c} 4 & 4 & 2 & 6 \\ 0 & 1 & 1,5 & 1,5 \\ 0 & 2 & 2 & 4 \end{array} \right].$$

Poslední prvek pod diagonálou v druhém sloupci je 2, takže přehodíme poslední dva řádky a dostaneme tak opět permutovanou matici

$$\left[\begin{array}{ccc|c} 4 & 4 & 2 & 6 \\ 0 & 2 & 2 & 4 \\ 0 & 1 & 1,5 & 1,5 \end{array} \right].$$

Abychom vynulovali subdiagonální prvek ve druhém sloupci, odečteme od třetího řádku 1/2-násobek druhého řádku a dostaneme tak konečný tvar vyeliminované matice jako

$$\left[\begin{array}{ccc|c} 4 & 4 & 2 & 6 \\ 0 & 2 & 2 & 4 \\ 0 & 0 & 0,5 & -0,5 \end{array} \right].$$

Převodli jsme tak původní soustavu rovnic na ekvivalentní soustavu rovnic s horní trojúhelníkovou maticí, kterou můžeme nyní vyřešit zpětnou substitucí a dostáváme tak stejně jako v Příkladu 10

$$\mathbf{x} = \begin{bmatrix} -1 \\ 3 \\ -1 \end{bmatrix}.$$

Pro zájemce uvádíme ještě výsledek ve formě LU rozkladu. Výsledná permutační matice je

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

a máme

$$\mathbf{PA} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0,25 & 0,5 & 1 \end{bmatrix} \begin{bmatrix} 4 & 4 & 2 \\ 0 & 2 & 2 \\ 0 & 0 & 0,5 \end{bmatrix} = \mathbf{LU}.$$

3 Dodatky

3.1 Numerické chyby při Gaussově eliminaci

Jak jsme již uvedli v Odstavci 1.3.5, platí pro relativní reziduum vypočítaného řešení nerovnost

$$\frac{\|\mathbf{r}\|}{\|\mathbf{A}\| \cdot \|\hat{\mathbf{x}}\|} \leq \frac{\|\mathbf{E}\|}{\|\mathbf{A}\|},$$

kde \mathbf{E} je zpětná chyba v matici \mathbf{A} . Ale jak velká asi bude $\|\mathbf{E}\|$ při praktických výpočtech? V numerické lineární algebře se ukazuje, že pro LU rozklad Gaussovou eliminací platí odhad tvaru

$$\frac{\|\mathbf{E}\|}{\|\mathbf{A}\|} \leq \rho^n \epsilon_{\text{mach}},$$

kde ρ se nazývá **růstový faktor** a ϵ_{mach} je strojové epsilon charakterizující přesnost počítačové aritmetiky. V Matlabu je standardně strojové epsilon k dispozici jako proměnná `eps` a je řádové velikosti 10^{-16} . Pokud jde o růstový faktor, je to poměr největšího prvku matice \mathbf{U} k největšímu prvku matice \mathbf{A} braný v absolutních hodnotách.

Pokud neprovádíme výběr hlavního prvku, může být tento faktor ρ libovolně velký, a tím pádem není Gaussova eliminační metoda bez pivotace obecně numericky stabilní (nemusí poskytovat malou zpětnou chybu). Pokud provádíme částečný výběr hlavního prvku, může být růstový faktor teoreticky velikosti 2^{n-1} (protože v nejhorším možném případě se velikost prvků může v každém kroku eliminace zdvojnásobit), ale takové chování Gaussovy eliminační metody je v praxi nesmírně vzácné. V prakticky řešených příkladech dochází při částečném výběru hlavních

prvků pouze k malému nebo žádnému růstu mezivýsledků, takže se dá přibližně brát jako odhad zpětné chyby

$$\frac{\|\mathbf{E}\|}{\|\mathbf{A}\|} \lesssim n \epsilon_{\text{mach}}.$$

Tento vztah znamená, že při řešení soustav lineárních rovnic Gaussovou eliminací s částečným výběrem hlavního prvku téměř vždy dostáváme velmi malá relativní rezidua, a to bez ohledu na to, jak špatně podmíněná řešená soustava je. Malé relativní reziduum tak nutně neznamená, že jsme získali přesný výsledek; to platí pouze u dobře podmíněných soustav. Doporučujeme čtenáři, aby se v této souvislosti znovu podíval na Příklad ??.

3.2 Metoda LU rozkladu na počítači

Existuje několik způsobů jak implementovat metodu LU rozkladu na počítači, které se liší v první řadě tím, jak se přistupuje k prvkům matice \mathbf{A} (po řádcích nebo po sloupcích). Účinnost těchto implementací se na daném počítači může výrazně lišit a jedna a ta samá implementace se může na různých výpočetních systémech chovat různě. Některé z variant Gaussovy eliminační metody si dokonce vysloužily vlastní pojmenování, takže máme například Croutovu metodu nebo Doolittlovu metodu. Speciální varianta metody LU rozkladu pro řešení soustav rovnic s třídiagonálními maticemi bývá v inženýrské praxi často nazývána Thomasovým algoritmem. Jakkoli se ale různé implementace LU rozkladu mohou významně lišit co do výkonnosti, produkují pro danou matici \mathbf{A} v podstatě tentýž výsledný rozklad.

Pokud jde o hospodaření s pamětí, je běžné, že vytvářené trojúhelníkové faktory postupně přepisují původní prvky matice \mathbf{A} . Prostorově to přesně vychází, protože matice \mathbf{L} má na diagonále jedničky a ty není třeba ukládat.

Aby se minimalizoval fyzický pohyb dat, neprovádí se přehazování řádků při výběru hlavních prvků explicitně. Místo toho řádky zůstávají na svých původních místech, ale k zaznamenávání jejich pořadí se používá pomocný celočíselný vektor indexů. Je totiž zřejmé, že konečný efekt všech záměn řádků je pouze nějaká permutace čísel $1, \dots, n$.

V Matlabu je algoritmus metody LU rozkladu s částečným výběrem hlavního prvku (včetně řešení obou trojúhelníkových soustav) zabudován do operátoru `\` (zpětné lomítko, angl. backslash). Řešení soustavy rovnic $\mathbf{Ax} = \mathbf{b}$ zapisujeme v Matlabu jako `x = A\b`. Matlab má také k dispozici funkci `inv` pro výpočet inverzní matice. Výslovně ale upozorňujeme, že počítat řešení soustavy jako `x = inv(A)*b` je zcela nesprávné. Takovýto způsob výpočtu je totiž podstatně pracnější než použití zpětného lomítka a je také horší z hlediska numerické stability, zejména u větších matic. Zápis `x = A^-1*b` je tedy užitečný pouze v teoretických úvahách na papíře.

3.3 Výpočetní složitost řešení lineárních soustav

Výpočet LU rozkladu matice řádu n Gaussovou eliminací vyžaduje zhruba $n^3/3$ násobení v pohyblivé řádové čárce a zhruba stejný počet sčítání. Řešení vzniklých trojúhelníkových soustav pro jednu pravou stranu si vyžádá asi n^2 násobení a podobný počet sčítání. Jestliže tedy řád matice n roste, začne v celkové výpočetní práci stále více převládat konstrukce LU rozkladu.

Z hlediska výpočetní složitosti je výpočet inverzní matice samotné asi třikrát náročnější než řešení soustavy s jednou pravou stranou. Dalším postupem, kterému je třeba se při výpočtech řešení soustav vyhnout, je Cramerovo pravidlo, v němž se každá složka řešení počítá jako podíl dvou determinantů. Pro větší matice (už jistě od řádu dvacet) je tento postup přímo astrono-

micky náročný na výpočetní práci a Cramerovo pravidlo se tak dá použít buď u matic velmi malého řádu nebo v teoretické lineární algebře.

Reference

- [1] MÍKA, Stanislav. *MVŠT. Numerické metody algebry*. Praha: SNTL, 1982, 169 s.
- [2] MÍKA, Stanislav. *MVŠT. Numerické metody algebry*. 2. opravené vyd. Praha: SNTL, 1985, 169 s.
- [3] MÍKA, Stanislav a Marek BRANDNER. *Numerické metody I*. Plzeň: FAV ZČU, 2000, 169 s.
- [4] HEATH, M. T. *Scientific Computing: An Introductory Survey*. 2nd Edition. New York: McGraw-Hill, 2002, 563 s.
- [5] HIGHAM, N. J. *Accuracy and Stability of Numerical Algorithms*. 2nd Edition. Philadelphia: SIAM, 2002, 680 s.