

UNIVERZITA KARLOVA V PRAZE

FARMACEUTICKÁ FAKULTA V HRADCI  
KRÁLOVÉ

KATEDRA BIOFYZIKY A FYZIKÁLNÍ CHEMIE



ŘEŠENÍ ŘÍDKÝCH SOUSTAV LINEÁRNÍCH  
ALGEBRAICKÝCH ROVNIC

Bakalářská práce

Vedoucí bakalářské práce: Dipl.-Math. Erik Jurjen Duintjer  
Tebbens, Ph.D.

HRADEC KRÁLOVÉ 2016

Kristýna Rojková

„Prohlašuji, že tato bakalářská práce je mým původním autorským dílem a veškeré myšlenky, data a jejich zdroje, z nichž jsem pro zpracování čerpala, řádně cituji. Práce nebyla využita k získání jiného nebo stejného titulu.“

V Hradci Králové dne

Kristýna Rojková

Mé poděkování si nejvíce zaslouží vedoucí této práce pan Dipl.-Math. Erik Jurjen Duintjer Tebbens, Ph.D., protože bez jeho odborného dohledu, rad a trpělivosti by tato práce nevznikla. Dále bych chtěla poděkovat své babičce magistře Jaroslavě Padevětové za poskytnutí gramatické i stylistické opravy práce.

# Obsah

Úvod	5
<b>1 Základní pojmy</b>	<b>7</b>
1.1 Matice . . . . .	7
1.2 Maticové operace . . . . .	8
1.3 Jednotková a inverzní matice . . . . .	12
1.4 Regulární matice . . . . .	13
1.5 Lineární soustavy . . . . .	14
<b>2 LU rozklad</b>	<b>17</b>
2.1 LU rozklad . . . . .	17
2.2 Postup řešení soustavy pomocí LU rozkladu . . . . .	18
2.3 Výpočetní náklady . . . . .	24
2.4 Permutační matice a numerická stabilita výpočtu LU rozkladu . . . . .	25
2.5 Permutace a řídké matice . . . . .	33
2.6 Zaplnění versus stabilita výpočtu . . . . .	35
2.7 LU rozklad pomocí programu Scilab . . . . .	36
<b>3 Krylovovské iterační metody</b>	<b>49</b>
3.1 Krylovovské podprostory a jejich báze . . . . .	49
3.1.1 Gram-Schmidtův ortogonalizační proces . . . . .	49
3.1.2 Arnoldiho algoritmus . . . . .	51
3.2 Konkrétní Krylovovské metody . . . . .	54
3.3 Výpočetní náklady . . . . .	55
3.4 Krylovovské podprostory pomocí programu Scilab . . . . .	56

<b>4 Porovnání efektivity a závěr</b>	<b>61</b>
4.1 LU rozklad . . . . .	61
4.2 Krylovovské metody . . . . .	62
<b>Literatura</b>	<b>65</b>

# Úvod

Tato bakalářská práce se zabývá řešením soustav lineárních algebraických rovnic. Popisuje dvě populární třídy metod pro řešení řídkých soustav o vysokém počtu rovnic. Takzvané řídké matice obsahují velké množství nulových prvků, což přináší dvě výhody. První výhodou je, že při ukládání těchto matic do počítače stačí uložit pouze nenulové prvky a druhou výhodou jsou levné výpočty (nízký počet aritmetických operací) s těmito řídkými maticemi. Řídké matice v praxi často vznikají například z důsledku diskretizace diferenciálních rovnic pomocí konečných diferencí nebo prvků [8].

Práce je členěná do tří hlavních kapitol. Tyto kapitoly většinou obsahují příklady s nízkým počtem rovnic pro lepší znázornění a orientaci.

První kapitola popisuje základní pojmy. Uvede maticové operace, definuje regulární a inverzní matici a popisuje lineární soustavy. Tato kapitola tím vysvětluje pojmy, které jsou používány pro stěžejní náplň práce. Také se zde uvede testovací příklad využíváný především ve druhé kapitole.

Druhá kapitola se zabývá první třídou metod řešení lineárních soustav a to  $LU$  rozkladem matice  $A$ . Jedná se o zobecnění Gaussovy eliminace. Popisuje výpočet rozkladu a následné řešení soustavy lineárních rovnic. Důležitou částí kapitoly jsou permutační matice, které kladně ovlivňují stabilitu výpočtu a zaplnění původní matice nenulovými prvky ve výsledném rozkladu. Dále je prezentováno řešení konkrétní soustavy s 900 rovnic pomocí softwaru Scilab. Tento software nabízí několik způsobů výpočtu  $LU$  rozkladu, které jsme porovnali a určili ten nejefektivnější z pohledu počtu nenulových prvků.

Třetí kapitola je určena pro druhou třídu metod a to Krylovovské iterační metody. Obsahuje popis Krylovovských podprostorů a výpočtu jejich ortogonálních bází orto-

---

gonalizačním procesem nazývaným Arnoldiho algoritmus. Řešili jsme pomocí Krylovských metod tutéž soustavu 900x900 jako v předchozí kapitole. Při porovnávání nejznámějších Krylovských metod jsme se zajímali především o počet iterací, i když je zde více faktorů, které efektivitu ovlivňují.

Závěrečná kapitola porovnává popsané třídy metod ( $LU$  rozklad a Krylovské metody) z hlediska efektivity ve tvaru stručného přehledu. Na závěr uvedeme původní znění zadání této bakalářské práce: Práce bude zaměřena na řešení soustav lineárních rovnic s velkými maticemi, které obsahují mnoho nulových prvků. Její plánované dva díly jsou (i) řešení založená na Gaussově eliminaci a permutacích matice (ii) iterační řešení. Práce by také měla vyzdvihnout praktické situace, kdy jeden přístup je výhodnější než druhý.

# Kapitola 1

## Základní pojmy

Na začátek uvedeme základní pojmy potřebné pro následující kapitoly. Zejména uvedeme definice: matice, řádké, jednotkové a inverzní matice a maticových operací. Pokusíme se u všech pojmů pro znázornění přidat jednoduchý příklad. Čerpali jsme z knih [4] a [17].

### 1.1 Matice

**Definice 1.1.1** Matice je obdélníkové či čtvercové schéma o  $m$  řádcích a  $n$  sloupcích obsahující reálná čísla. Množinu matic o  $m$  řádcích a  $n$  sloupcích označujeme  $\mathbb{R}^{m \times n}$ . Jestliže platí  $m \neq n$ , mluvíme o obdélníkové matici. Pokud  $m = n$ , pak mluvíme o čtvercové matici řádu  $n$ . Pokud  $n = 1$ , mluvíme o sloupcovém vektoru. Pokud  $m = 1$ , mluvíme o řádkovém vektoru.

Matice je možné definovat i přes jiné obory, například komplexní čísla. Pro tuto práci však stačí uvažovat matice s reálnými čísly. Jednoduchý příklad čtvercové matice řádu 3 je

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}.$$

Obdélníková matice z  $\mathbb{R}^{3 \times 2}$  může například vypadat následovně

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}.$$

Máme-li obecnou matici  $A \in \mathbb{R}^{m \times n}$ , pak prvky této matice označujeme  $a_{i,j}$ , kde  $i = 1, \dots, m$ ,  $j = 1, \dots, n$ . Například pro obecnou matici řádu 3, můžeme použít označení

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

Velké matice vznikající z konkrétního praktického problému mají často mnoho nulových prvků.

**Definice 1.1.2** *O matici mluvíme jako o řídké tehdy, je-li výhodné použít speciální techniky uložení nebo výpočtů, které zohledňují velké množství nulových prvků.*

Výhody těchto řídkých matic jsou podrobněji popsány v sekci 2.5.

## 1.2 Maticové operace

**Definice 1.2.1** *Pro matici  $A \in \mathbb{R}^{m \times n}$  je transponovaná matice  $A^T \in \mathbb{R}^{n \times m}$  definována jako matice, jejíž sloupce jsou tvořeny řádky původní matice  $A$ .*

Například pro matici

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

je transponovanou maticí

$$A^T = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}.$$

**Definice 1.2.2** *Součet (rozdíl) matic je definován pouze pro matice stejné velikosti. Prvky výsledné matice se rovnají součtu (rozdílu) odpovídajících prvků sčítaných (odčítaných) matic. Výsledkem je matice stejné velikosti.*

Například pro

$$A = \begin{pmatrix} 1 & 2 \\ 5 & 0 \\ 7 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 9 & 0 \\ -4 & 1 \\ 6 & 2 \end{pmatrix}$$

je součet  $A + B$  roven

$$A + B = \begin{pmatrix} 1 & 2 \\ 5 & 0 \\ 7 & 1 \end{pmatrix} + \begin{pmatrix} 9 & 0 \\ -4 & 1 \\ 6 & 2 \end{pmatrix} = \begin{pmatrix} 10 & 2 \\ 1 & 1 \\ 13 & 3 \end{pmatrix}.$$



---

**Definice 1.2.3** Součin čísla a matice je definován pro libovolné číslo  $\alpha \in \mathbb{R}$  a každou matici. Prvky výsledné matice jsou vynásobeny příslušným číslem. Výsledná matice je stejné velikosti jako původní.

Například pro

$$\alpha = 2, \quad A = \begin{pmatrix} 4 & 6 & 1 \\ 2 & 5 & 8 \end{pmatrix}$$

je součin  $\alpha A$  roven

$$\alpha \cdot A = 2 \cdot \begin{pmatrix} 4 & 6 & 1 \\ 2 & 5 & 8 \end{pmatrix} = \begin{pmatrix} 8 & 12 & 2 \\ 4 & 10 & 16 \end{pmatrix}.$$

**Definice 1.2.4** Skalární součin je definován pro řádkový vektor  $v = (v_1, \dots, v_n)$  a sloupcový vektor se stejným počtem prvků  $w = (w_1, \dots, w_n)^T$  v tomto pořadí vztahem  $vw = \sum_{i=1}^n v_i w_i$ . Výsledek je číslo, které vzniklo součtem součinů odpovídajících prvků.

Například pro

$$v = (4 \quad 3 \quad 6), \quad w = \begin{pmatrix} 3 \\ 7 \\ 2 \end{pmatrix}$$

je skalární součin  $v \cdot w$  roven

$$vw = (4 \quad 3 \quad 6) \begin{pmatrix} 3 \\ 7 \\ 2 \end{pmatrix} = 4 \cdot 3 + 3 \cdot 7 + 6 \cdot 2 = 12 + 21 + 12 = 45.$$

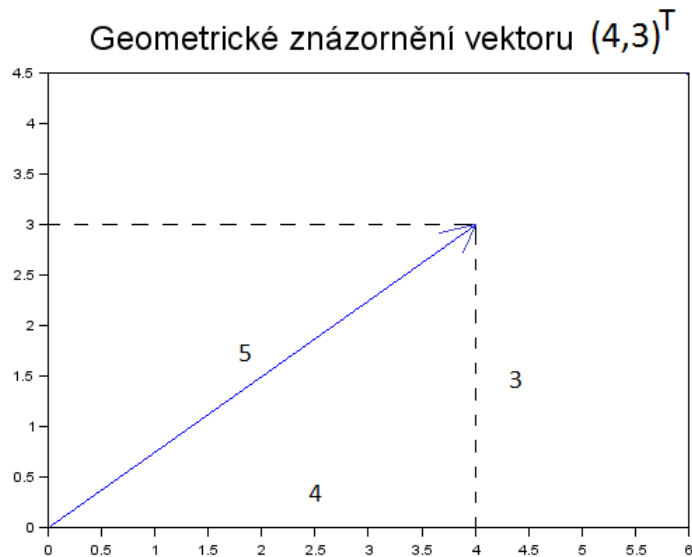
Dva a třídídimenzionální vektory můžeme geometricky interpretovat jako "šipky" v rovině nebo v prostoru. Například geometrické znázornění vektoru  $(4, 3)^T$  vidíme na obrázku 1.1. Jestliže násobíme dva navzájem geometricky kolmé neboli *ortogonální* vektory, pak jejich skalární součin je roven nule.

Máme-li například dva kolmé vektory

$$q_1 = (1 \quad 0 \quad 0), \quad q_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix},$$

pak je jejich skalární součin

$$q_1 q_2 = 1 \cdot 0 + 0 \cdot 1 + 0 \cdot 0 = 0.$$



Obrázek 1.1: Geometrické znázornění vektoru  $(4,3)^T$

**Definice 1.2.5** Velikost vektoru neboli jeho norma se spočítá jako odmocnina součtu druhých mocnin prvků vektoru. Tento výpočet vychází z Pythagorovy věty, jejíž grafické znázornění vidíme na obrázku 1.1. Je-li norma vektoru rovna jedné, pak je tento vektor normalizovaný.

Máme - li vektor

$$q = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix},$$

pak jeho norma je

$$\|q\| = \sqrt{q_1^2 + q_2^2 + q_3^2}.$$

Jestliže chceme provést skalární součin vektoru  $q$  sám sebou  $q^T q$ , pak výsledek součinu je roven kvadrátu normy:

$$q^T q = q_1^2 + q_2^2 + q_3^2 = \|q\|^2.$$

**Definice 1.2.6** Ortonormální matice  $Q_k \in \mathbb{R}^{n \times k}$  je taková matice, jejíž sloupce představující vektory jsou vzájemně kolmé a jejichž norma je rovna jedné.

**Definice 1.2.7** Součinem matic  $AB$  vznikne matice  $C$ , která obsahuje prvky  $c_{i,j}$ , odpovídající skalárnímu součinu  $i$ -tého řádku matice  $A$  a  $j$ -tého sloupce matice  $B$ .

**Poznámka:** Z definice 1.2.4 vyplývá, že násobit matice můžeme jen tehdy, jestliže počet sloupců první matice odpovídá počtu řádků druhé matice. Navíc jestliže  $A \in \mathbb{R}^{m \times n}$  a  $B \in \mathbb{R}^{n \times k}$ , pak  $AB \in \mathbb{R}^{m \times k}$ .

Například pro

$$A = \begin{pmatrix} 5 & 2 & 3 \\ 1 & 6 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 7 & 2 \\ -2 & 1 \\ 3 & 2 \end{pmatrix}$$

je součin  $AB$  roven

$$\begin{aligned} AB &= \begin{pmatrix} 5 & 2 & 3 \\ 1 & 6 & 4 \end{pmatrix} \begin{pmatrix} 7 & 2 \\ -2 & 1 \\ 3 & 2 \end{pmatrix} = \begin{pmatrix} (5 \cdot 7 - 2 \cdot 2 + 3 \cdot 3) & (5 \cdot 2 + 2 \cdot 1 + 3 \cdot 2) \\ (1 \cdot 7 - 6 \cdot 2 + 4 \cdot 3) & (1 \cdot 2 + 6 \cdot 1 + 4 \cdot 2) \end{pmatrix} \\ &= \begin{pmatrix} 40 & 18 \\ 7 & 16 \end{pmatrix}. \end{aligned}$$

**Definice 1.2.8** *Násobení dvou matic  $A$  a  $B$  je komutativní, pokud platí vztah  $AB = BA$ . Pak se též říká, že matice  $A$  a  $B$  komutují.*

Pro obdélníkové matice  $A \in \mathbb{R}^{m \times n}$  a  $B \in \mathbb{R}^{n \times m}$  je zřejmé, že matice  $AB \in \mathbb{R}^{m \times m}$  nemůže kvůli odlišné velikosti být rovna  $BA \in \mathbb{R}^{n \times n}$ . Ale ani pro čtvercové matice při všeobecném násobení tento vztah neplatí, a proto násobení matic obecně není komutativní.

Například pro matice

$$A = \begin{pmatrix} 5 & 2 & 3 \\ 1 & 6 & 4 \\ 2 & 3 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 7 & 5 & 2 \\ -2 & 1 & 6 \\ 2 & 4 & 6 \end{pmatrix}$$

lze násobit v pořadí  $A, B$  jako

$$AB = \begin{pmatrix} 5 & 2 & 3 \\ 1 & 6 & 4 \\ 2 & 3 & 4 \end{pmatrix} \begin{pmatrix} 7 & 5 & 2 \\ -2 & 1 & 6 \\ 2 & 4 & 6 \end{pmatrix} = \begin{pmatrix} 37 & 39 & 40 \\ 3 & 27 & 62 \\ 16 & 29 & 46 \end{pmatrix},$$

nebo také v pořadí  $B, A$  jako

$$BA = \begin{pmatrix} 7 & 5 & 2 \\ -2 & 1 & 6 \\ 2 & 4 & 6 \end{pmatrix} \begin{pmatrix} 5 & 2 & 3 \\ 1 & 6 & 4 \\ 2 & 3 & 4 \end{pmatrix} = \begin{pmatrix} 44 & 50 & 49 \\ 3 & 20 & 22 \\ 26 & 46 & 46 \end{pmatrix}.$$

Je vidět, že  $AB \neq BA$ .

---

**Věta 1.2.9** Pro obecně obdélníkové matice  $A, B$ , které lze násobit, platí  $(AB)^T = B^T A^T$ .

Důkaz:

Nechť je  $C = AB$ . Jestliže vynásobíme  $i$ -tý řádek matice  $A$  s  $j$ -tým sloupcem matice  $B$ , pak získáme prvek  $c_{i,j}$  matice  $C$  na pozici  $i, j$ . Prvek na pozici  $i, j$  pro transponovanou matici  $C^T$  se rovná  $c_{j,i}$ . Získáme tento prvek tím, že vynásobíme  $j$ -tý řádek matice  $A$  s  $i$ -tým sloupcem matice  $B$ . To je ale to samé jako násobit  $i$ -tý řádek matice  $B^T$  s  $j$ -tým sloupcem matice  $A^T$ , čili prvek matice  $B^T A^T$  na pozici  $i, j$ .

### 1.3 Jednotková a inverzní matice

**Definice 1.3.1** Jednotková matice  $I_n$  je matice s vlastností, že pro libovolnou matici  $A \in \mathbb{R}^{n \times m}$  platí  $I_n A = A$  a zároveň platí  $B I_n = B$  pro libovolnou matici  $B \in \mathbb{R}^{m \times n}$ .

Z definice vyplývá, že každá jednotková matice je čtvercová, označuje se proto  $I_n$ , kde  $n$  vyjadřuje řád této matice. Každá jednotková matice má na hlavní diagonále prvky rovny jedné a ostatní prvky jsou rovny nule. Hlavní diagonálou myslíme prvky  $a_{1,1}, \dots, a_{n,n}$ .

Například jednotková matice  $I_3$  vypadá takto:

$$I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

**Definice 1.3.2** Inverzní matice  $A^{-1}$  ke čtvercové matici  $A$  řádu  $n$ , pokud existuje, je jednoznačně určená matice stejného řádu, pro kterou platí  $AA^{-1} = I_n = A^{-1}A$ .

Například pro  $A = \begin{pmatrix} 1 & 2 \\ 3 & 5 \end{pmatrix}$  je inverzní matice  $A^{-1} = \begin{pmatrix} -5 & 2 \\ 3 & -1 \end{pmatrix}$ , protože platí:

$$AA^{-1} = \begin{pmatrix} 1 & 2 \\ 3 & 5 \end{pmatrix} \begin{pmatrix} -5 & 2 \\ 3 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I_2,$$

druhou rovnici nemusíme ověřit, protože z definice 1.3.2 vyplývá, že násobení matice s maticí k ní inverzní je komutativní.

**Poznámka:** Pokud je matice  $Q_k$  ortonormální, pak součin matice  $Q_k$  s její transpozicí  $Q_k^T$  je roven jednotkové matici:

$$Q_k^T Q_k = I_k.$$

---

Pokud navíc  $Q_k \in \mathbb{R}^{n \times k}$  je čtvercová, to je  $k = n$ , pak

$$Q_n^{-1} = Q_n^T.$$

**Poznámka:** Chceme-li spočítat normu součinu ortonormální matice s vektorem, pak stačí spočítat pouze normu vektoru:

$$(1.1) \quad \|Q_k v\| = \sqrt{(Q_k v)^T Q_k v} = \sqrt{v^T Q_k^T Q_k v} = \sqrt{v^T v} = \|v\|.$$

## 1.4 Regulární matice

**Definice 1.4.1** Čtvercová matice  $A$  je regulární právě tehdy, existuje-li k ní inverzní matice  $A^{-1}$ . Čtvercová matice, která není regulární, se jmenuje singulární.

V následující větě používáme pojem determinant (označeno  $\det(A)$ ). Pro její definici odkážeme na [4, Kapitola 1.1]. Tady uvedeme dva jednoduché příklady výpočtu determinantu. Jeden pro matici  $A \in \mathbb{R}^{2 \times 2}$ :

$$\det \left( \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \right) = a_{1,1} \cdot a_{2,2} - a_{1,2} \cdot a_{2,1},$$

druhý pro matici  $A \in \mathbb{R}^{3 \times 3}$ :

$$\begin{aligned} \det \left( \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix} \right) &= a_{1,1} \cdot \det \left( \begin{pmatrix} a_{2,2} & a_{2,3} \\ a_{3,2} & a_{3,3} \end{pmatrix} \right) \\ &- a_{2,1} \cdot \det \left( \begin{pmatrix} a_{1,2} & a_{1,3} \\ a_{3,2} & a_{3,3} \end{pmatrix} \right) + a_{3,1} \cdot \det \left( \begin{pmatrix} a_{1,2} & a_{1,3} \\ a_{2,2} & a_{2,3} \end{pmatrix} \right). \end{aligned}$$

Analogicky se tento postup výpočtu determinantu zobecňuje pro větší matice, viz [4, Kapitola 1.1, Věta 1.1]. V následující větě znak "  $\Leftrightarrow$  " znamená ekvivalence neboli "tehdy a právě tehdy".

**Věta 1.4.2** Matice  $A \in \mathbb{R}^{n \times n}$  je singulární  $\Leftrightarrow \det(A) = 0 \Leftrightarrow$  sloupce matice  $A$  jsou lineárně závislé. [4, Kapitola 1.1]

Singulární matice může vypadat například takto:

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

---

kde třetí sloupec je dvojnásobkem druhého sloupce, a tudíž poslední dva sloupce jsou navzájem lineárně závislé. Snadno lze ověřit, že zároveň platí  $\det(A) = 0$ .

**Věta 1.4.3** *Pro regulární matice  $A, B$  stejného řádu platí  $(AB)^{-1} = B^{-1}A^{-1}$ .*

Důkaz:

Jestliže  $X = (AB)^{-1}$ , platí  $ABX = I_n$ , kde  $n$  je řád matic  $A$  a  $B$ . Dosažením  $X = B^{-1}A^{-1}$  zjistíme, že  $ABB^{-1}A^{-1} = AI_nA^{-1} = AA^{-1} = I_n$ , tudíž  $B^{-1}A^{-1}$  je jednoznačně určená inverze matice  $AB$ .

## 1.5 Lineární soustavy

Cílem této práce je popis dvou tříd metod pro řešení soustav lineárních algebraických rovnic, neboli zkráceně lineárních soustav.

**Definice 1.5.1** *Lineární soustava je systém  $n$  rovnic spolu s  $n$  neznámých  $x_1, \dots, x_n \in \mathbb{R}$  takový, že v každé rovnici každá neznámá  $x_i$  se vyskytuje maximálně v první mocnině.*

Lineární soustava, která obsahuje tři neznámé a tři rovnice, může vypadat například takto:

$$\begin{aligned}x_1 + 2x_2 + 3x_3 &= 6 \\2x_1 + 5x_2 + 2x_3 &= 9 \\6x_1 - 3x_2 + x_3 &= 4.\end{aligned}$$

Každá soustava lze zapsat maticově ve tvaru:

$$A \cdot x = b, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

Pro předchozí příklad je

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 2 \\ 6 & -3 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 6 \\ 9 \\ 4 \end{pmatrix},$$

soustava je zapsána jako

$$(1.3) \quad \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 2 \\ 6 & -3 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ 9 \\ 4 \end{pmatrix}.$$

$A$  nazýváme maticí soustavy,  $b$  pravou stranou a  $x$  vektorem neznámých. Zde jsme použili definice rovnosti matic, viz [17, strana 63].

Podmínkou pro jednoznačnost řešení rovnice je, že matice soustavy  $A$  musí být regulární. Pokud by byla singulární, soustava by neměla řešení, nebo by měla nekonečně mnoho řešení. Pro celou práci předpokládáme, že matice soustavy  $A$  je regulární.

Ilustrujeme zde metody pro řešení lineárních soustav na dvou příkladech. První je právě uvedený, velmi jednoduchý příklad (1.3), druhý je soustava, která vzniká diskretizací diferenciální (konvexní-difúzní) rovnice, kde  $u$  je neznámá funkce dvou proměnných,  $u : \mathbb{R}^2 \rightarrow \mathbb{R}$  :

$$(1.4) \quad -\nu \Delta u + w \cdot \nabla u = 0 \quad \text{v } \Omega = (0, 1) \times (0, 1), \quad u = g \quad \text{na } \partial\Omega,$$

$\nu$  je skalární difúzní koeficient a je roven 0,001 (tuto hodnotu jsme zvolili),  $w$  je řádkový vektor  $(0 \ 1)$ ,  $\Delta$  je operátor zvaný Laplacián, který označuje součet druhých parciálních derivací funkcí dle  $x$  a  $y$

$$\Delta u : \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2},$$

$\nabla$  je operátor zvaný gradient, který označuje sloupcový vektor, jehož prvky jsou první parciální derivace funkcí dle  $x$  a  $y$

$$\nabla u : \begin{pmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{pmatrix},$$

$\Omega$  je oblast v rovině a  $g$  je funkce  $\mathbb{R}^2 \rightarrow \mathbb{R}$  daná na okraji roviny  $\partial \Omega$ .

Diskretizace diferenciální rovnice (1.4) pomocí metody konečných prvků spočívá, velmi stručně řečeno, v omezení na určení funkčních hodnot neznámé funkce  $u$  v pravidelné síti v oblasti  $\Omega$  tvořené  $N = 30$  vnitřními body v každém směru. V každém bodě této sítě aproximujeme rovnici (1.4) pomocí lineárních kombinací funkcí  $\phi_i$  (takzvané bilineární konečné prvky). Pro  $N^2$  bodů tím dostaneme lineární soustavu  $N^2$  rovnic. Jelikož aproximace Laplaciánu a gradientu pomocí  $A_d = \langle \nabla \phi_j, \nabla \phi_i \rangle$  respektive  $A_c = \langle w \cdot \nabla \phi_j, \phi_i \rangle$ ,  $1 \leq i, j \leq N^2$ , kde  $\langle \cdot, \cdot \rangle$  označuje vhodnou aproximaci derivace, používá pouze informaci o sousedních bodech. Příslušná soustava je velmi řídká (každý bod má maximálně 8 relevantních sousedů, tudíž každý řádek soustavy má nejvíce 9 nenulových prvků). Spolu se stabilizačním členem  $A_s = \langle w \cdot \nabla \phi_j, w \cdot \nabla \phi_i \rangle$ ,

---

$1 \leq i, j \leq N^2$  a námi zvoleným stabilizačním parametrem  $\hat{\delta} = -0,4$ , dostaneme matici soustavy  $A$  tvaru

$$(1.5) \quad A = \nu A_d + A_c + \hat{\delta} A_s, \in \mathbb{R}^{n \times n}.$$

Pravá strana soustavy je určena volbou funkce  $g$ . Tady používáme takzvané Raitby okrajové podmínky, viz například [19]. Pro více detailů odkážeme na články [7, 9, 11, 19, 6].



# Kapitola 2

## LU rozklad

Pravděpodobně nejstarší způsob řešení soustav lineárních algebraických rovnic

$$Ax = b,$$

je  $LU$  rozklad matice  $A$ . Je založen na takzvané Gaussově eliminaci.  $LU$  rozklad byl poprvé popsán v článku [22] v roce 1948 A. Turingem. Informace pro  $LU$  rozklad jsme čerpali z knihy [4].

### 2.1 LU rozklad

**Definice 2.1.1** Horní trojúhelníková matice je čtvercová matice, jejíž dolní trojúhelníkovou část (bez hlavní diagonály) tvoří pouze nuly, horní trojúhelníkovou část (včetně hlavní diagonály) tvoří prvky rovny nule nebo různé od nuly:

$$U = \begin{pmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \dots & u_{1,n} \\ 0 & u_{2,2} & u_{2,3} & \dots & u_{2,n} \\ 0 & 0 & u_{3,3} & \dots & u_{3,n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & u_{n,n} \end{pmatrix}.$$

**Definice 2.1.2** Dolní trojúhelníková matice je čtvercová matice, jejíž horní trojúhelníkovou část tvoří pouze nuly a dolní trojúhelníkovou část tvoří čísla rovna nebo různá od nuly:

$$L = \begin{pmatrix} l_{1,1} & 0 & 0 & \dots & 0 \\ l_{2,1} & l_{2,2} & 0 & \dots & 0 \\ l_{3,1} & l_{3,2} & l_{3,3} & \ddots & 0 \\ \vdots & \vdots & \ddots & \ddots & 0 \\ l_{n,1} & l_{n,2} & l_{n,3} & \dots & l_{n,n} \end{pmatrix}.$$

**Definice 2.1.3** LU rozklad čtvercové matice  $A$  je rozklad tvaru

$$A = LU,$$

kde  $L$  je dolní trojúhelníková matice (lower triangular) s jedničkami na hlavní diagonále a  $U$  je horní trojúhelníková matice (upper triangular).

## 2.2 Postup řešení soustavy pomocí LU rozkladu

Řešme lineární soustavu  $Ax = b$  pomocí  $LU$  rozkladu matice soustavy  $A$ .

Pro ověření, zda je matice soustavy  $A$  regulární, můžeme u malých matic spočítat její determinant, který nesmí být nulový. U velkých matic determinant z důvodů numerické stability nepočítáme. Zda je matice regulární, poznáme během procesu, jak bude jasné z poznámky za větou 2.4.2.

Matice soustavy  $A$  se musí vynásobit takovou dolní trojúhelníkovou maticí  $L^{-1}$ , aby vznikla horní trojúhelníková matice  $U$ . Pokud se toto povede, pak

$$(2.1) \quad Ax = b \Leftrightarrow L^{-1}Ax = L^{-1}b \Leftrightarrow Ux = L^{-1}b.$$

Dostaneme tedy soustavu  $Ux = L^{-1}b$ , která má v poslední rovnici jen jednu neznámou, v předposlední dvě neznámé a takto kaskádovitě budou neznámé přibývat. Tento tvar je velmi výhodný, protože se postupně mohou neznámé dopočítávat a tím bude soustava vyřešena. Analogicky počítáme pravou stranu  $y = L^{-1}b$ , která se rovná řešení soustavy  $Ly = b$ . To znamená, že v první rovnici je jedna neznámá, ve druhé rovnici dvě neznámé a takto kaskádovitě přibývají. Nyní bude následovat podrobný postup  $LU$  rozkladu  $A$ , a to prozatím pro konkrétní soustavu (1.3).

V našem konkrétním případě hledáme prvky  $m_{i,j}$  (prvky matice  $L^{-1}$ ) takové, aby

$$U = L^{-1}A = \begin{pmatrix} 1 & 0 & 0 \\ m_{2,1} & 1 & 0 \\ m_{3,1} & m_{3,2} & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 2 \\ 6 & -3 & 1 \end{pmatrix}$$

byla matice v horním trojúhelníkovém tvaru. V prvním kroku zkusíme vynulovat prvky pod hlavní diagonálou prvního sloupce matice  $A$  pomocí vhodné volby prvků  $m_{2,1}$  a  $m_{3,1}$ . V našem případě se jedná o prvky 2 a 6. Prvek  $m_{3,2}$  považujeme prozatím za nulový.

Naším cílem je najít takové číslo  $m_{2,1}$ , aby po vynásobení řádku  $(m_{2,1}, 1, 0)$  matice  $L^{-1}$  se sloupcem  $\begin{pmatrix} a_{1,1} \\ a_{2,1} \\ a_{3,1} \end{pmatrix}$  matice  $A$  byl výsledek, tedy prvek v pozici  $(2,1)$ , který označíme  $a_{2,1}^{(1)}$ , nulový. Máme tudíž podmínku

$$\begin{aligned} m_{2,1} \cdot a_{1,1} + 1 \cdot a_{2,1} + 0 \cdot a_{3,1} &= 0, \\ m_{2,1} \cdot a_{1,1} + a_{2,1} &= 0, \\ m_{2,1} &= -\frac{a_{2,1}}{a_{1,1}}. \end{aligned}$$

Po dosazení konkrétními hodnotami dostaneme:

$$m_{2,1} = \frac{-2}{1} = -2.$$

Analogický postup provedeme i pro  $m_{3,1}$ . Příslušná podmínka je

$$\begin{aligned} m_{3,1} \cdot a_{1,1} + 0 \cdot a_{2,1} + 1 \cdot a_{3,1} &= 0, \\ m_{3,1} \cdot a_{1,1} + a_{3,1} &= 0, \\ m_{3,1} &= -\frac{a_{3,1}}{a_{1,1}}. \end{aligned}$$

Po dosazení dostaneme:

$$m_{3,1} = -\frac{6}{1} = -6.$$

Získané prvky dosadíme do matice, kterou označíme

$$M^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -6 & 0 & 1 \end{pmatrix},$$

protože jsme se zabývali poddiagonálními prvky prvního sloupce matice  $A$ .

Vynásobíme získanou maticí  $M^{(1)}$  maticí  $A$  a dostaneme matici  $A^{(1)}$ , která má nulové poddiagonálové prvky prvního sloupce:

$$M^{(1)} \cdot A = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -6 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 2 \\ 6 & -3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & -4 \\ 0 & -15 & -17 \end{pmatrix} = A^{(1)}.$$

Aby byla zachována rovnost mezi pravou a levou stranou rovnice, musíme vynásobit získanou maticí  $M^{(1)}$  i pravou stranu soustavy  $b$  a získáme vektor  $b^{(1)}$ .

$$M^{(1)} \cdot b = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -6 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 6 \\ 9 \\ 4 \end{pmatrix} = \begin{pmatrix} 6 \\ -3 \\ -32 \end{pmatrix} = b^{(1)}.$$

Ve druhém kroku si všímáme poddiagonálních prvků druhého sloupce matice  $A^{(1)}$ . V našem případě prvku  $-15$ . Prvky, které jsme spočítali v prvním kroku, opomeneme a nahradíme je nulami:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & m_{3,2} & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & -4 \\ 0 & -15 & -17 \end{pmatrix} = A^{(2)}.$$

Postupujeme analogicky, jako v prvním kroku, a dostaneme hodnotu  $m_{3,2} = 15$ .

Získali jsme matici  $M^{(2)}$ , se kterou vynásobíme jak levou stranu  $A^{(1)}$ , tím dostaneme  $A^{(2)} = U$ , tak pravou stranu  $b^{(1)}$  a dostaneme  $b^{(2)}$ .

$$M^{(2)} \cdot A^{(1)} = A^{(2)} = U =$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 15 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & -4 \\ 0 & -15 & -17 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & -4 \\ 0 & 0 & -77 \end{pmatrix},$$

$$M^{(2)} \cdot b^{(1)} = b^{(2)} =$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 15 & 1 \end{pmatrix} \cdot \begin{pmatrix} 6 \\ -3 \\ -32 \end{pmatrix} = \begin{pmatrix} 6 \\ -3 \\ -77 \end{pmatrix}.$$

Nyní se původní maticový zápis (1.3) změnil na matematicky ekvivalentní tvar:

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & -4 \\ 0 & 0 & -77 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ -3 \\ -77 \end{pmatrix}.$$

Jestliže z maticového zápisu vytvoříme opět soustavu, dostaneme:

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

---

$$-77x_3 = -77.$$

Z poslední rovnice vypočítáme  $x_3$  :

$$x_3 = \frac{-77}{-77} = 1.$$

Z předposlední pak vypočítáme  $x_2$ :

$$x_2 = -3 - (-4 \cdot 1) = 1.$$

Z první nakonec spočítáme  $x_1$ :

$$x_1 = 6 - (3 \cdot 1 + 2 \cdot 1) = 1.$$

Tato soustava má řešení:  $x = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ .

Právě popsany postup můžeme zobecnit pro soustavu s  $n$  rovnic. Poznamenáme, že v popsaném postupu byly provedeny jak násobení  $L^{-1}A$ , tak násobení  $L^{-1}b$ , to znamená, že byla vytvořena druhá ekvivalentní rovnice v (2.1). Tento takzvaný přímý chod Gaussovy eliminace můžeme popsat ve tvaru pseudoalgoritmu:

1. Vložíme  $A^{(0)} := A = [a_{i,j}]$ ,  $b^{(0)} := b = [\beta_1, \dots, \beta_n]^T$

2. Pro  $k = 1 : n - 1$

3. pro  $i = k + 1 : n$

$$m_{i,k} := \frac{a_{i,k}^{(k-1)}}{a_{k,k}^{(k-1)}}$$

Pro  $j = k + 1 : n$

$$a_{i,j}^{(k)} := a_{i,j}^{(k-1)} - m_{i,k}a_{k,j}^{(k-1)}$$

$$\beta_i^{(k)} := \beta_i^{(k-1)} - m_{i,k}\beta_k^{(k-1)}$$

4. Výsledek je  $A^{(n-1)} = U$  a  $b^{(n-1)} = L^{-1}b$

Zpětný chod Gaussovy eliminace je řešení systému s horní trojúhelníkovou maticí  $U$ , to je řešení třetí ekvivalentní rovnice v (2.1). Příslušný pseudoalgoritmus může vypadat následovně:

Pseudoalgoritmus: Zpětný chod Gaussovy eliminace

---

1. Vložíme  $A^{(n-1)}, b^{(n-1)}$

2.  $\xi_n := \beta_n^{(n-1)} / a_{n,n}^{(n-1)}$

3. Pro  $i = n - 1 : -1 : 1$

$$\xi_i := \beta_i^{n-1}$$

Pro  $j = i + 1 : n$

$$\xi_i := \xi_i - a_{i,j}^{(n-1)} \xi_j$$

$$\xi_i := \xi_i / a_{i,i}^{(n-1)}$$

4. Výsledek je řešení soustavy  $x = (\xi_1 \ \dots \ \xi_n)^T$

Pro úplnost uvedeme, jak vypadá matice  $L$  spočítaného  $LU$  rozkladu pro soustavu (1.3). Po celou dobu jsme místo rozkladu  $A = LU$  používali rozklad  $L^{-1}A = U$ . Matici  $L^{-1}$  můžeme vyjádřit pro vzorový příklad (1.3) jako součin matic  $M^{(2)}$  a  $M^{(1)}$ . Jestliže platí vztah

$$L^{-1} = M^{(2)}M^{(1)},$$

pak platí

$$L = (M^{(1)})^{-1}(M^{(2)})^{-1},$$

viz věta 1.4.3.

Pro zápis matic  $M^{(1)}$  a  $M^{(2)}$  můžeme využít jednotkové matice:

$$\begin{aligned} M^{(1)} &= \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -6 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 \\ -2 \\ -6 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \\ &= I_3 + \begin{pmatrix} 0 \\ -2 \\ -6 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, \end{aligned}$$

---

analogicky pro  $M^{(2)}$ :

$$M^{(2)} = I_3 + \begin{pmatrix} 0 \\ 0 \\ 15 \end{pmatrix} (0 \ 1 \ 0).$$

Hledané inverzní matice musí být takové, aby součin původní matice a k ní inverzní matice dal jednotkovou matici. V případě  $M^{(1)}$  stačí změnit jedno znaménko:

$$\begin{aligned} & \left( I_3 + \begin{pmatrix} 0 \\ -2 \\ -6 \end{pmatrix} (1 \ 0 \ 0) \right) \left( I_3 - \begin{pmatrix} 0 \\ -2 \\ -6 \end{pmatrix} (1 \ 0 \ 0) \right) \\ &= I_3 - \begin{pmatrix} 0 \\ -2 \\ -6 \end{pmatrix} (1 \ 0 \ 0) + \begin{pmatrix} 0 \\ -2 \\ -6 \end{pmatrix} (1 \ 0 \ 0) \\ & \quad - \begin{pmatrix} 0 \\ -2 \\ -6 \end{pmatrix} (1 \ 0 \ 0) \begin{pmatrix} -2 \\ -6 \end{pmatrix} (1 \ 0 \ 0) = I_3. \end{aligned}$$

Poslední člen je nulový, protože vnitřní skalární součin je nulový. Proto inverzní matice k  $M^{(1)}$  je:

$$(M^{(1)})^{-1} = I_3 - \begin{pmatrix} 0 \\ -2 \\ -6 \end{pmatrix} (1 \ 0 \ 0) = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 6 & 0 & 1 \end{pmatrix}.$$

Analogicky pro matici  $M^{(2)}$  platí:

$$(M^{(2)})^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -15 & 1 \end{pmatrix}.$$

**Poznámka:** Takovýto výpočet inverzní matice je možné jen díky speciální struktuře matic  $M^{(1)}$  a  $M^{(2)}$ . Pro obecný důkaz odkážeme na [4, str 117, cvičení 4.1].

Nyní můžeme spočítat matici  $L$ :

$$L = (M^{(1)})^{-1} \cdot (M^{(2)})^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 6 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -15 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 6 & -15 & 1 \end{pmatrix}.$$

Tabulka 2.1: Počty jednotlivých operací pro výpočet  $LU$  rozkladu.

Operace	celkem
*	$\frac{1}{3}n^3 - \frac{1}{2}n^2 + \frac{1}{6}n$
-	$\frac{1}{3}n^3 - \frac{1}{2}n^2 + \frac{1}{6}n$
:	$\frac{1}{2}n^2 - \frac{1}{2}n$

Můžeme se všimnout, že došlo pouze k sečtení poddiagonálních prvků. Všeobecný důkaz této vlastnosti najdeme ve skriptech [4, Kapitola 4.1, rovnice (4.4)].

Pro kontrolu můžeme spočítat matici  $A$  podle vztahu  $A = LU$ :

$$LU = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 6 & -15 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & -4 \\ 0 & 0 & -77 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 2 \\ 6 & -3 & 1 \end{pmatrix} = A.$$

## 2.3 Výpočetní náklady

Nyní se díváme na počty operací během celého procesu  $LU$  rozkladu. Shrnutí všech počtů operací najdeme v tabulce 2.1, Jelikož počty úměrné  $n^2$  nebo  $n$  jsou zanedbatelné v porovnání s počty úměrnými  $n^3$ , lze říci, že celkový počet aritmetických operací potřebných pro  $LU$  rozklad je řádově  $\frac{2}{3}n^3$ . Operace odčítání a násobení jsou přibližně stejně drahé, dělení je poněkud dražší. Souhrn počtů operací při zpětném chodu najdeme v tabulce 2.2 a nachází se na úrovni  $n^2$  operací.

Tabulka 2.2: Počty jednotlivých operací při zpětném chodu.

Operace	celkem
*	$\frac{1}{2}n^2 - \frac{1}{2}n$
+	$\frac{1}{2}n^2 - \frac{3}{1}n + 1$
-	$n - 1$
:	1



---

## 2.4 Permutační matice a numerická stabilita výpočtu LU rozkladu

Během procesu  $LU$  rozkladu se dělí v  $k$ -tém kroku prvkem  $a_{k,k}^{k-1}$ , viz pseudoalgoritmus pro přímý chod v odstavci 2.2. Prvkům  $a_{k,k}^{k-1}$  se říká pivotní prvky. Může tedy dojít k dělení velmi malým číslem, a proto ke vzniku velkých číselných mezivýsledků. V průběhu výpočtu se mohou pak ztrácet platné cifry a může dojít k silné numerické nestabilitě. Může dokonce dojít k dělení nulou, pokud je nějaký pivotní prvek nulový. Tím pádem musí výpočet skončit a lze dokázat, že to se může stát, nejen když  $A$  je singulární, ale jakmile  $A$  není tzv. silně regulární. Silná regularita je regularitou všech hlavních podmatic, viz [4, 4.2]. Těmto komplikacím lze zabránit násobením takzvanou permutační maticí.

**Definice 2.4.1** Permutační matice je taková matice, kde v každém řádku a sloupci této matice je pouze jedna jednička a zbytek prvků jsou nuly. Jednotková matice není považována za permutační matici.

Jestliže budeme permutační maticí násobit matici  $A$  zleva (pak se tato matice označuje  $P$ ), dojde k obměně řádků matice  $A$ . Při násobení permutační maticí zprava ( $Q$ ) dojde k obměně sloupců matice  $A$ .

Pro úplnost uvedeme příklady násobení matice

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

zprava permutační maticí  $P$ :

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

a zleva permutační maticí  $Q$ :

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

V následujících třech příkladech uvidíme, proč je počítání s permutačními maticemi výhodné vzhledem ke stabilitě výpočtů. První je výpočet bez použití permutační

matice a se zaokrouhlením mezivýsledků na malý počet platných cifer (jak tomu je v běžných kalkulačkách<sup>1</sup>). Druhý příklad je výpočet bez použití permutační matice, ale se zaokrouhlením na vysoký počet platných cifer. Třetí příklad je výpočet s použitím permutační matice.

Ve všech případech uvažujeme modifikovanou soustavu (1.3), přesněji lineární soustavu

$$Ax = b,$$

kde

$$A = \begin{pmatrix} 10^{-9} & 2 & 3 \\ 2 & 5 & 2 \\ 6 & -3 & 1 \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad b = \begin{pmatrix} 6 \\ 4 \\ 2 \end{pmatrix}.$$

Jak nyní uvidíme, malý prvek  $10^{-9}$  na pozici (1,1) může způsobit nestabilitu. Spočítejme, podle postupu popsaného v odstavci 2.2, první krok  $LU$  rozkladu, konkrétně matice  $M^{(1)}, A^{(1)}, b^{(1)}$ .

$$M^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{2}{10^{-9}} & 1 & 0 \\ -\frac{6}{10^{-9}} & 0 & 1 \end{pmatrix}, \quad A^{(1)} = \begin{pmatrix} 10^{-9} & 2 & 3 \\ 0 & -3999999995 & -5999999998 \\ 0 & -1,2 \cdot 10^{10} & -1,8 \cdot 10^{10} \end{pmatrix},$$

přičemž  $A^{(1)}$  získáme pomocí násobení (na kalkulačce)

$$\begin{pmatrix} 1 & 0 & 0 \\ -\frac{2}{10^{-9}} & 1 & 0 \\ -\frac{6}{10^{-9}} & 0 & 1 \end{pmatrix} \begin{pmatrix} 10^{-9} & 2 & 3 \\ 2 & 5 & 2 \\ 6 & -3 & 1 \end{pmatrix} = \begin{pmatrix} 10^{-9} & 2 & 3 \\ 0 & -3999999995 & -5999999998 \\ 0 & -1,2 \cdot 10^{10} & -1,8 \cdot 10^{10} \end{pmatrix}.$$

Spočítáme  $b^{(1)}$  pomocí  $M^{(1)}$ :

$$b^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{2}{10^{-9}} & 1 & 0 \\ -\frac{6}{10^{-9}} & 0 & 1 \end{pmatrix} \begin{pmatrix} 6 \\ 4 \\ 2 \end{pmatrix} = \begin{pmatrix} 6 \\ -1,2 \cdot 10^{10} \\ -3,6 \cdot 10^{10} \end{pmatrix}.$$

Hrubé zaokrouhlení je zde vidět ve všech číslech velikosti řádově  $10^{10}$ .

Ve druhém kroku vypočítáme  $M^{(2)}, A^{(2)}, b^{(2)}$ .

$$M^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1,2 \cdot 10^{10}}{3999999995} & 1 \end{pmatrix}, \quad A^{(2)} = \begin{pmatrix} 10^{-9} & 2 & 3 \\ 0 & -3999999995 & -5999999998 \\ 0 & 0 & 16,499 \end{pmatrix},$$

<sup>1</sup>použili jsme zaokrouhlení kalkulačkou Casio - fx-85ES.

přičemž podle kalkulačky

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1,2 \cdot 10^{10}}{3999999995} & 1 \end{pmatrix} \begin{pmatrix} 10^{-9} & 2 & 3 \\ 0 & -3999999995 & -5999999998 \\ 0 & -1,2 \cdot 10^{10} & -1,8 \cdot 10^{10} \end{pmatrix} \\ = \begin{pmatrix} 10^{-9} & 2 & 3 \\ 0 & -3999999995 & -5999999998 \\ 0 & 0 & 16,499 \end{pmatrix}.$$

Spočítáme  $b^{(2)}$  pomocí  $M^{(2)}$ :

$$b^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1,2 \cdot 10^{10}}{3999999995} & 1 \end{pmatrix} \begin{pmatrix} 6 \\ -1,2 \cdot 10^{10} \\ -3,6 \cdot 10^{10} \end{pmatrix} = \begin{pmatrix} 6 \\ -1,2 \cdot 10^{10} \\ 45 \end{pmatrix},$$

Hrubé zaokrouhlení prvního kroku má v tomto kroku vliv i na "rozumné" výsledky (prvky 16,499 a 45).

Pro úplnost vypočítáme i  $L$

$$L^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1,2 \cdot 10^{10}}{3999999995} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\frac{2}{10^{-9}} & 1 & 0 \\ -\frac{6}{10^{-9}} & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{2}{10^{-9}} & 1 & 0 \\ -\frac{6}{10^{-9}} & -\frac{1,2 \cdot 10^{10}}{3999999995} & 1 \end{pmatrix},$$

tudíž

$$L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{2}{10^{-9}} & 1 & 0 \\ \frac{6}{10^{-9}} & \frac{1,2 \cdot 10^{10}}{3999999995} & 1 \end{pmatrix}.$$

Dokončíme výpočet vyřešením ekvivalentní soustavy

$$\begin{aligned} 10^{-9}x_1 + 2x_2 + 3x_3 &= 6 \\ -3999999995x_2 - 5999999998x_3 &= -1,2 \cdot 10^{10} \\ 16,499x_3 &= 45. \end{aligned}$$

To vede k řešení

$$x = \begin{pmatrix} -269998,6416 \\ -1,09115704 \\ 2,727348027 \end{pmatrix}.$$

Snadno si ověříme, že toto řešení není správné:

$$Ax = \begin{pmatrix} 10^{-9} & 2 & 3 \\ 2 & 5 & 2 \\ 6 & -3 & 1 \end{pmatrix} \begin{pmatrix} -269998,6416 \\ -1,09115704 \\ 2,727348027 \end{pmatrix} = \begin{pmatrix} 5,99946 \\ -539997,28 \\ -1619985,8 \end{pmatrix} \neq b.$$

Tato velká chyba výsledku je důsledkem pouze toho, že používáme nízký počet platných cifer při zaokrouhlení. Ve druhém příkladu používáme podstatně více platných cifer (maximálně 12). V prvním kroku dostaneme:

$$M^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{2}{10^{-9}} & 1 & 0 \\ -\frac{6}{10^{-9}} & 0 & 1 \end{pmatrix}, \quad A^{(1)} = \begin{pmatrix} 10^{-9} & 2 & 3 \\ 0 & -3999999995 & -5999999998 \\ 0 & -12000000003 & -17999999999 \end{pmatrix}$$

a platí

$$\begin{pmatrix} 1 & 0 & 0 \\ -\frac{2}{10^{-9}} & 1 & 0 \\ -\frac{6}{10^{-9}} & 0 & 1 \end{pmatrix} \begin{pmatrix} 10^{-9} & 2 & 3 \\ 2 & 5 & 2 \\ 6 & -3 & 1 \end{pmatrix} = \begin{pmatrix} 10^{-9} & 2 & 3 \\ 0 & -3999999995 & -5999999998 \\ 0 & -12000000003 & -17999999999 \end{pmatrix}.$$

Spočítáme  $b^{(1)}$  pomocí  $M^{(1)}$ :

$$b^{(1)} = \begin{pmatrix} 1 & 2 & 0 \\ -\frac{2}{10^{-9}} & 1 & 0 \\ -\frac{6}{10^{-9}} & 0 & 1 \end{pmatrix} \begin{pmatrix} 6 \\ 4 \\ 2 \end{pmatrix} = \begin{pmatrix} 6 \\ -11999999996 \\ -35999999998 \end{pmatrix}.$$

Výsledky se patrně liší od výsledků prvního kroku v prvním příkladě. Ve druhém kroku vypočítáme  $M^{(2)}$ ,  $A^{(2)}$ ,  $b^{(2)}$ .

$$M^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{12000000003}{3999999995} & 1 \end{pmatrix} \quad A^{(2)} = \begin{pmatrix} 10^{-9} & 2 & 3 \\ 0 & -3999999995 & -5999999998 \\ 0 & 0 & 22 \end{pmatrix}$$

a platí

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{12000000003}{3999999995} & 1 \end{pmatrix} \begin{pmatrix} 10^{-9} & 2 & 3 \\ 0 & -3999999995 & -5999999998 \\ 0 & -12000000003 & -17999999999 \end{pmatrix} \\ = \begin{pmatrix} 10^{-9} & 2 & 3 \\ 0 & -3999999995 & -5999999998 \\ 0 & 0 & 22 \end{pmatrix}.$$

Spočítáme  $b^{(2)}$  pomocí  $M^{(2)}$ :

$$b^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{12000000003}{3999999995} & 1 \end{pmatrix} \begin{pmatrix} 6 \\ -11999999996 \\ -35999999998 \end{pmatrix} = \begin{pmatrix} 6 \\ -11999999996 \\ 44 \end{pmatrix}.$$

Největší rozdíly v porovnání s prvním příkladem jsou v číslech 22 a 44. Pro úplnost vypočítáme  $L$

$$L^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{12000000003}{3999999995} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\frac{2}{10^{-9}} & 1 & 0 \\ -\frac{6}{10^{-9}} & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{2}{10^{-9}} & 1 & 0 \\ -\frac{6}{10^{-9}} & -\frac{12000000003}{3999999995} & 1 \end{pmatrix},$$

tudíž

$$L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{2}{10^{-9}} & 1 & 0 \\ \frac{6}{10^{-9}} & \frac{12000000003}{3999999995} & 1 \end{pmatrix}.$$

Dokončíme výpočet vyřešením ekvivalentní soustavy

$$\begin{aligned} 10^{-9}x_1 + 2x_2 + 3x_3 &= 6 \\ -3999999995x_2 - 5999999998x_3 &= -11999999996 \\ 22x_3 &= 44. \end{aligned}$$

To vede k řešení

$$x = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix},$$

které se výrazně liší od řešení v prvním příkladu a je navíc správné:

$$Ax = \begin{pmatrix} 10^{-9} & 2 & 3 \\ 2 & 5 & 2 \\ 6 & -3 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \\ 2 \end{pmatrix} = b.$$

Použitý počet platných cifer byl pro tento příklad postačující, nicméně mezivýsledky obsahují velká čísla. Ve třetím příkladu se vyhneme potenciálně velkým mezivýsledkům pomocí permutační matice  $P$ :

$$(2.2) \quad P = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

Upravíme matici  $A$  permutační maticí  $P$  do tvaru  $PA$

$$PA = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 10^{-9} & 2 & 3 \\ 2 & 5 & 2 \\ 6 & -3 & 1 \end{pmatrix} = \begin{pmatrix} 6 & -3 & 1 \\ 2 & 5 & 2 \\ 10^{-9} & 2 & 3 \end{pmatrix}.$$

---

Abychom nezměnili původní řešení rovnice, musíme vynásobit maticí  $P$  i vektor  $b$ ,

$$Pb = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 6 \\ 4 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix}.$$

Nyní máme rovnici ve tvaru  $PAx = Pb$ :

$$\begin{pmatrix} 6 & 3 & 1 \\ 2 & 5 & 2 \\ 10^{-9} & 2 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix}.$$

Celý proces  $LU$  rozkladu nyní probíhá bez rizik špatného zaokrouhlení.

V prvním kroku dostaneme

$$M^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ -\frac{10^{-9}}{6} & 0 & 1 \end{pmatrix}, \quad A^{(1)} = \begin{pmatrix} 6 & -3 & 1 \\ 0 & 6 & \frac{5}{3} \\ 0 & 2 & 3 \end{pmatrix}$$

a platí

$$\begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ -\frac{10^{-9}}{6} & 0 & 1 \end{pmatrix} \begin{pmatrix} 6 & 3 & 1 \\ 2 & 5 & 2 \\ 10^{-9} & 2 & 3 \end{pmatrix} = \begin{pmatrix} 6 & -3 & 1 \\ 0 & 6 & \frac{5}{3} \\ 0 & 2 & 3 \end{pmatrix}.$$

Spočítáme  $b^{(1)}$  pomocí  $M^{(1)}$ :

$$b^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ -\frac{10^{-9}}{6} & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 4 \\ 6 \end{pmatrix} = \begin{pmatrix} 2 \\ \frac{10}{3} \\ 6 \end{pmatrix}.$$

Výsledky se zcela liší od výsledků v prvních krocích obou předchozích příkladů.

Ve druhém kroku vypočítáme  $M^{(2)}, A^{(2)}, b^{(2)}$ :

$$M^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{3} & 1 \end{pmatrix}, \quad A^{(2)} = \begin{pmatrix} 6 & -3 & 1 \\ 0 & 6 & \frac{5}{3} \\ 0 & 0 & \frac{22}{9} \end{pmatrix}$$

a platí

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{3} & 1 \end{pmatrix} \begin{pmatrix} 6 & -3 & 1 \\ 0 & 6 & \frac{5}{3} \\ 0 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 6 & -3 & 1 \\ 0 & 6 & \frac{5}{3} \\ 0 & 0 & \frac{22}{9} \end{pmatrix}.$$

---

Spočítáme  $b^{(2)}$  pomocí  $M^{(2)}$

$$b^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{3} & 1 \end{pmatrix} \begin{pmatrix} 2 \\ \frac{10}{3} \\ 6 \end{pmatrix} = \begin{pmatrix} 2 \\ \frac{10}{3} \\ \frac{44}{9} \end{pmatrix},$$

Rozdíly v porovnání s prvním i druhým příkladem jsou patrné v číslech  $\frac{22}{9}$  a  $\frac{44}{9}$ . Pro úplnost vypočítáme  $L$

$$L^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{1}{3} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ -\frac{10^{-9}}{6} & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ -\frac{1}{3} & -\frac{10^{-9}}{6} & 1 \end{pmatrix},$$

tudíž

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ -\frac{1}{3} & -\frac{10^{-9}}{6} & 1 \end{pmatrix}.$$

Dokončíme výpočet vyřešením ekvivalentní soustavy

$$6x_1 - 3x_2 + x_3 = 2$$

$$6x_2 + \frac{5}{3}x_3 = \frac{10}{3}$$

$$\frac{22}{9}x_3 = \frac{44}{9}.$$

To vede k řešení

$$x = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix},$$

které je shodné s řešením druhého příkladu a je proto také správné.

Z těchto tří příkladů je důležité si odnést dva poznatky. První - je velmi výhodné použít vhodnou permutační matici, výpočty jsou jednodušší a je menší pravděpodobnost výskytu numerických chyb. Druhý - pokud nepoužijeme permutační matici, musíme všechna čísla zaokrouhlit na dostatečný počet platných cifer. Chyby vzniklé nesprávným zaokrouhlením se během procesu zvětšují a může při zaokrouhlení dojít k výpočtu zcela chybného řešení.

Výběr vhodné permutační matice (2.2) byl proveden na základě následující úvahy.

Permutační maticí se snažíme dosáhnout toho, aby v  $i$ -tém kroku  $LU$  rozkladu bylo na pivotní pozici  $(i, i)$  vždy největší možné číslo ze všech prvků na pozicích  $(i, i), \dots, (n, i)$  v  $i$ -tém sloupci. V našem případě bylo volbou (2.2) zaručeno nejen to, že v prvním kroku bylo největší číslo (6) na pozici (1,1), ale zároveň bylo ve druhém kroku největší číslo (také 6) na pozici (2,2). Obecně může však pro každý krok  $LU$  rozkladu existovat jiná vhodná permutace řádků, která zajistí, aby došlo v daném kroku k dělení co největším možným číslem.

Uvažujeme-li matici  $A$  řádu  $n$ , pak proces  $LU$  rozkladu pomocí řádkových permutací bude vypadat následovně. Zvolíme pro matici  $A$  vhodnou permutační matici  $P_1$  zajišťující velký pivotní prvek v prvním sloupci, pak provedeme první krok procesu  $LU$  rozkladu:

$$M^{(1)}P_1A = A^{(1)}.$$

Zvolíme další vhodnou permutační matici  $P_2$  pro matici  $A^{(1)}$  a provedeme druhý krok procesu  $LU$  rozkladu:

$$M^{(2)}P_2M^{(1)}P_1A = A^{(2)}.$$

Takto probíhá proces  $LU$  rozkladu až do tvaru:

$$M^{(n-1)}P_{n-1} \dots M^{(2)}P_2M^{(1)}P_1 \cdot A = U.$$

Lze dokázat, že součin

$$M^{(n-1)}P_{n-1} \dots M^{(2)}P_2M^{(1)}P_1$$

se rovná součinu

$$(M^{(n-1)} \cdot \dots \cdot M^{(2)}M^{(1)}) \cdot (P_{n-1} \cdot \dots \cdot P_2P_1),$$

viz [4, Kapitola 4, Odstavec 4.2.1].

S označením

$$L^{-1} = M^{(n-1)} \cdot \dots \cdot M^{(2)}M^{(1)}$$

a

$$P = P_{n-1} \cdot \dots \cdot P_2P_1$$

dostaneme celkem

$$L^{-1}PA = U$$



---

čili

$$PA = LU.$$

Výsledná permutační matice  $P$  vzniká součinem permutačních matic v průběhu procesu  $LU$  rozkladu.

Díky permutačním maticím můžeme provést  $LU$  rozklad pro libovolnou regulární matici:

**Věta 2.4.2** *Je-li matice  $A$  regulární, pak existuje permutační matice  $P$  taková, že výpočet  $LU$  rozkladu matice  $PA$  probíhá až do posledního kroku (zejména všechny diagonální prvky matice  $U$  jsou nenulové).*

Důkaz věty najdeme v literatuře [4, Kapitola 4.2].

**Poznámka:** Je-li matice  $A$  singulární, pak při výpočtu  $LU$  rozkladu s permutacemi musí v určitém kroku dojít k situaci, kdy se musí proces ukončit, protože v příslušném sloupci jsou nulové jak poddiagonální prvky, tak diagonální prvek.

## 2.5 Permutace a řídké matice

Z definice 1.1.2 víme, co jsou řídké matice. Takové matice mají dvě důležité výhody. První výhoda nastává, pokud takovou matici můžeme uložit v počítači bez uložení nulových prvků; zabere to málo místa v paměti. Druhá výhoda vzniká při násobení matice s vektory. Toto násobení je levné, jelikož při výpočtu skalárních součinů vznikají nulové prvky, se kterými nemusíme počítat.

Při výpočtu  $LU$  rozkladu řídké matice  $A$  vznikají matice  $L$  a  $U$ , které mohou mít podstatně menší počet nulových prvků. Naštěstí lze tento problém často elegantně odstranit tím, že řídkou matici  $A$  vynásobíme permutační maticí  $P$  nebo  $Q$ . Matice  $P$  a  $Q$  se pokusíme volit takové, aby docházelo při  $LU$  rozkladu matice  $PAQ$  k co nejmenšímu zaplnění nenulovými prvky a aby byl výpočet rozumně stabilní.

Uvedeme dva příklady; první bez použití permutační matice a druhý s použitím permutační matice.

Například uvažujeme řádkou matici  $A$ :

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

V prvním příkladu provedeme první krok procesu  $LU$  rozkladu

$$M^{(1)}A = A^{(1)},$$

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

Matice  $A$  obsahuje dvanáct nulových prvků a matice  $A^{(1)}$  obsahuje pouze čtyři nulové prvky. Došlo k zaplnění matice a můžeme očekávat, že i výsledný faktor  $U$  bude málo řádký.

Ve druhém příkladu použijeme například permutační matici

$$P^{(1)} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

pak

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

Po vynásobení permutační maticí  $P^{(1)}$  provedeme výpočet prvního kroku  $LU$  rozkladu

$$M^{(1)}P^{(1)}A = A^{(1)},$$

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

Matice  $A^{(1)}$  obsahuje třináct nulových prvků. Můžeme si všimnout, že matice  $A^{(1)}$  obsahuje v hlavní diagonále na druhém místě číslo (1), které můžeme pomocí permutační matice zaměnit za větší číslo (2). Proto použijeme další permutace řádků.

$$P^{(2)}A^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 2 & -\frac{1}{2} & 0 \\ 0 & 2 & 1 & \frac{3}{2} & 1 \\ 0 & 0 & 0 & -\frac{1}{2} & 2 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 & 1 & 0 \\ 0 & 2 & 1 & \frac{3}{2} & 1 \\ 0 & 0 & 2 & -\frac{1}{2} & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -\frac{1}{2} & 2 \end{pmatrix}.$$

Nyní spočítáme  $M^{(2)}, A^{(2)}$

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

Kdybychom chtěli pokračovat ve výpočtu  $M^{(3)}, A^{(3)}$ , už bychom nepotřebovali permutační matici ani z důvodu stability, ani z důvodu zaplnění.

## 2.6 Zaplnění versus stabilita výpočtu

Vliv permutačních matic se může zdát jednoznačný. Bohužel tomu tak není. Je velmi složité najít takové permutační matice, aby minimalizovaly zaplnění a zároveň posilovaly stabilitu výpočtu. Pro nalezení permutačních matic splňujících obě podmínky (nízké zaplnění a vysoká stabilita) se v současné době využívají velmi sofistikované komerční softwarové balíky. Mezi tyto softwarové balíky patří například MUMPS [1], UMFPACK [2], SuperLU [3], GPLU [14], MA48 [25], WSMP [15], PARDISO [10]. Tyto softwary často nejprve hledají vhodné permutace na základě umístění nenulových prvků bez ohledu na hodnoty prvků. Tato fáze se nazývá symbolický výpočet

---

a v této fázi je primárním cílem minimalizace zaplnění. Ve druhé fázi nazývajícím se numerický výpočet se provádí samotná Gaussova eliminace, a zde se mohou do výpočtu zakomponovat další permutace kvůli stabilitě.

## 2.7 LU rozklad pomocí programu Scilab

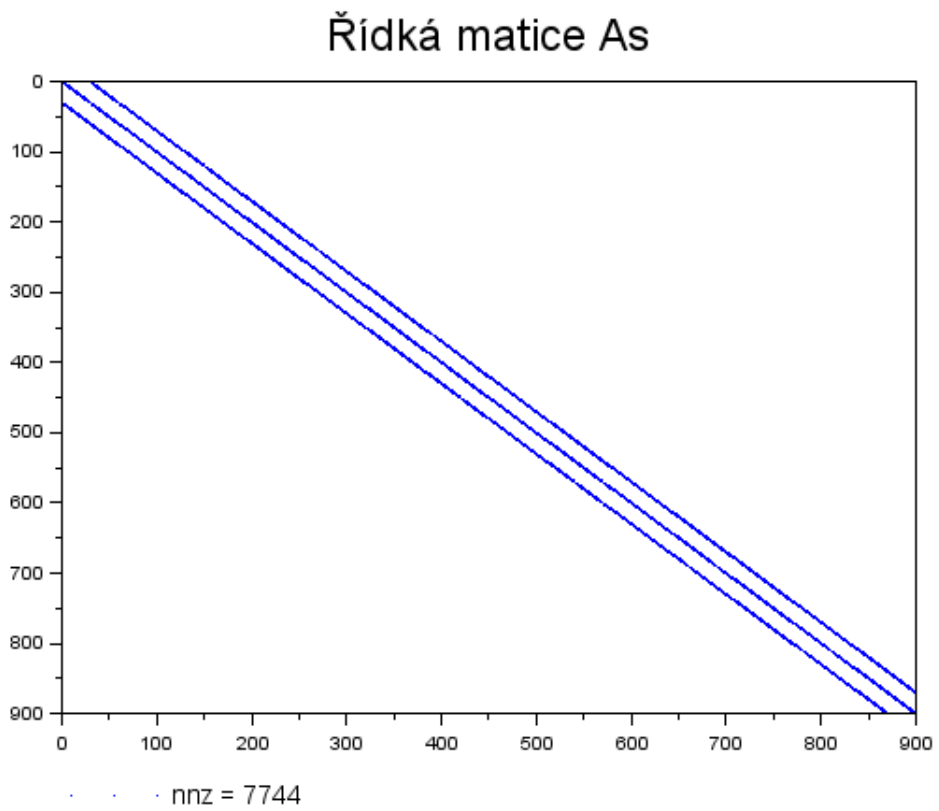
Pro soustavy s větším počtem rovnic lze využít počítačový program, například komerční software Matlab 8.5 [26] a volně dostupné balíky Octave 4.0.1 [27] nebo Scilab 5.5.2 [28]. Excel [24] má ve standardní nabídce pouze funkci pro invertování matic. Gaussova eliminace a  $LU$  rozklad v Excelu nejsou. Pro naše výpočty jsme zvolili program Scilab 5.5.2. Tento program poskytuje několik způsobů řešení rovnice  $Ax = b$  pomocí  $LU$  rozkladu. Pokusíme se najít nejvýhodnější způsob. Kritériem nejvýhodnějšího způsobu bude nejmenší počet nenulových prvků matic  $L$  a  $U$ , což má vliv na počet operací během řešení. Posouzení efektivity  $LU$  rozkladu probereme podrobněji v poslední kapitole, prozatím stačí toto kritérium.

Pokusíme se vyřešit soustavu prezentovanou na konci druhé kapitoly spojenou s konvexním-difúzním problémem s maticí  $A \in \mathbb{R}^{900 \times 900}$  a pravou stranou  $b \in \mathbb{R}^{900 \times 1}$ . Matici  $A$  uložíme jako  $As$ , což značí uložení matice jako řídké (uložíme pouze nenulové prvky).

Umístění nenulových prvků matice  $As$  je zobrazeno na obrázku 2.1. Může se zdát, že je matice  $As$  tvořena třemi diagonálami. Pokud dojde k přiblížení, zjistíme, že každá diagonála je tvořena dalšími třemi diagonálami, tudíž matice  $A$  je tvořena devíti diagonálami, viz obrázek 2.2. Důvod, proč má matice devět diagonál, jsme popsali na konci první kapitoly při popisu konvexního-difúzního problému. Dále z grafu můžeme vyčíst počet nenulových prvků ( $\text{nnz}$ ). Matice  $As$  obsahuje 7 744 nenulových prvků, přičemž celkem obsahuje  $900^2 = 810\,000$  prvků. Je tedy velmi řídká. V prvním způsobu výpočtu  $LU$  rozkladu matice  $As$  vyzkoušíme výpočet bez použití permutačních matic.

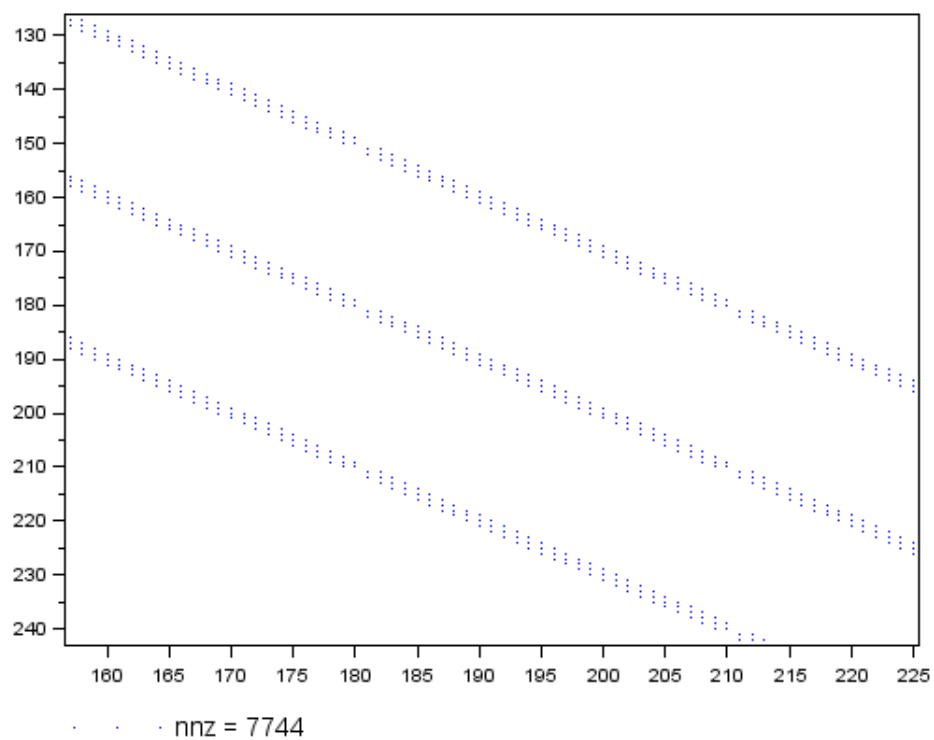
Použijeme ručně vytvořenou funkci  $[L, U] = \text{ilu0}(As, 0)$ . Příkazem  $\text{PlotSparse}(L)$  a  $\text{PlotSparse}(U)$  dostaneme grafy umístění nenulových prvků matic  $L$  a  $U$  i počet nenulových prvků.

Umístění nenulových prvků matice  $L$  můžeme vidět na obrázku 2.3 a umístění ne-

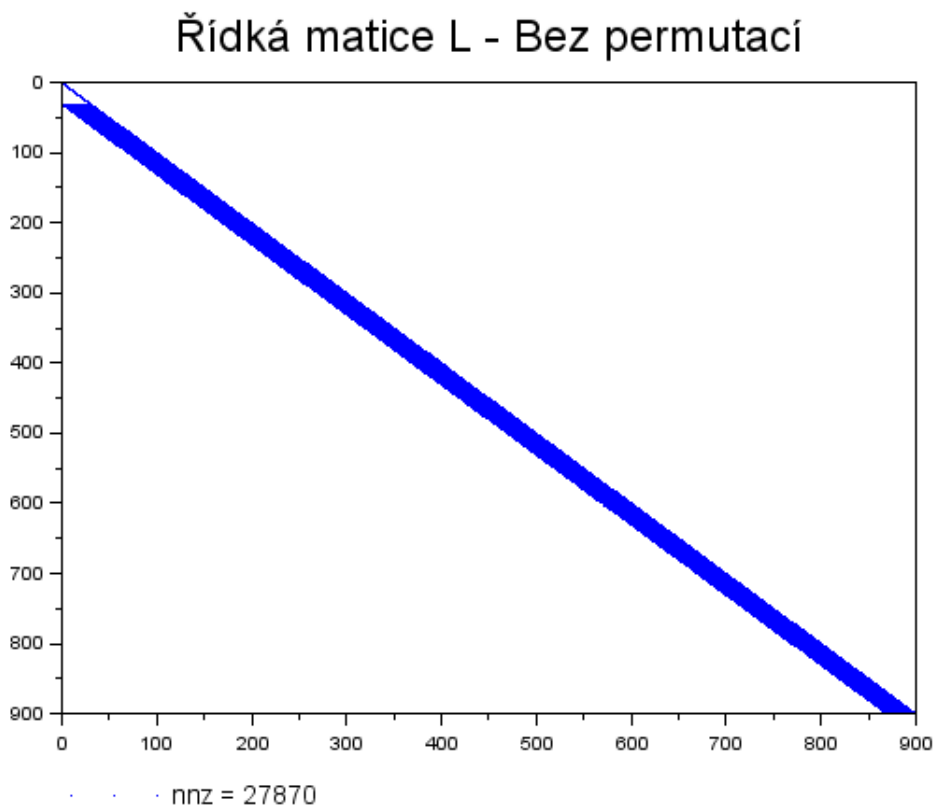


Obrázek 2.1: Řídká matice  $A_s$

## Řídká matice $A_s$

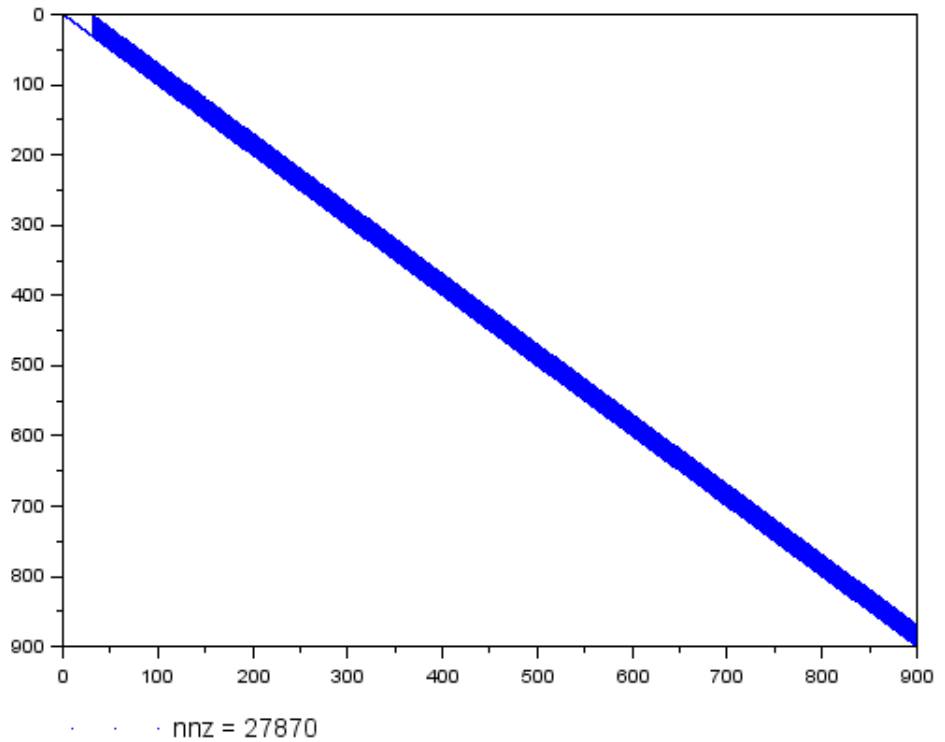


Obrázek 2.2: Řídká matice  $A_s$  - přiblížení



Obrázek 2.3: Řídká matice L - Bez permutací

## Řídká matice U - Bez permutací



Obrázek 2.4: Řídká matice U - Bez permutací

nulových prvků matice  $U$  na obrázku 2.4 .

Můžeme vidět, že zaplnění matic  $L$  a  $U$  je oproti původní matici  $As$  výrazně vyšší. Zda opravdu došlo ke správnému výpočtu, si můžeme ověřit zadáním příkazu:  $norm(L * U - As)$ ; počítá se pak standardní maticová norma (takzvaná spektrální norma). Po zadání příkazu vyšlo číslo 1.075D-17. Toto číslo je na úrovni strojové přesnosti a můžeme ho považovat za nulu. Došlo ke správnému výpočtu, jelikož součin matic  $L * U$  je roven matici  $As$ .

Chceme-li soustavu vyřešit, zadáme příkazy, které vycházejí z ekvivalencí

$$Ax = b \Leftrightarrow LUx = b \Leftrightarrow Ux = L^{-1}b \Leftrightarrow x = U^{-1}(L^{-1}b).$$

Je tedy potřeba řešit dvě soustavy, jednu s maticí  $L$  a jednu s maticí  $U$ . Zadáme proto příkaz  $y = L \setminus b$  a poté  $x = U \setminus y$ . Pro kontrolu zadáme  $norm(As * x - b)$  a opět je výsledek roven nule.



Ve druhém způsobu vyzkoušíme výpočet s použitím permutačních matic na základě kódu ze Scilabu. Zadáme příkaz  $hand = lu\text{fact}(As, [eps, reps])$ , kde parametr  $eps$  označuje minimální velikost, kterou prvky musí mít, aby mohly být kandidátem pro permutaci takovou, že tento prvek bude na diagonální pozici (a během Gaussovy eliminace se bude dělit tímto prvkem). Program vybere největšího z těchto kandidátů pro pivotní prvek. Pro úplnost je třeba uvést, že tento program umožňuje kvůli stabilitě nejen řádkové, ale i sloupcové permutace a hledá největší prvek nejen ve sloupci pod aktuálním diagonálním prvkem, ale v celé submatici zprava a dole od diagonálního prvku. Parametr  $reps$  klade další podmínku na takové permutace. Kandidátní prvek  $\tilde{a}$  může být permutován na diagonální pozici pouze tehdy, když  $\tilde{a} \geq reps \cdot a_{i,i}$ , kde  $a_{i,i}$  je aktuální diagonální prvek. Hodnota  $reps$  musí být mezi jedničkou a nulou. Čím bližší k jedničce, tím bude proces stabilnější, ale může mít tendenci k zaplňování matic. Naopak čím bude bližší nule, tím méně se dbá na stabilitu, ale matice budou zpravidla řidší.

Nejprve si zvolíme parametry  $eps$  a  $reps$   $[0, 0.1]$ . Zadáme příkaz  $hand = lu\text{fact}(As, [0, 0.1])$ , poté  $[P, L, U, Q] = lu\text{get}(hand)$ . Došlo k  $LU$  rozkladu matice  $PAQ$ .

Pomocí příkazu  $PlotSparse(L)$  a  $PlotSparse(U)$  získáme umístění nenulových prvků v maticích a počet nenulových prvků.

Umístění nenulových prvků matice  $L$  jsou zobrazeny na obrázku 2.5 a umístění nenulových prvků matice  $U$  jsou zobrazeny na obrázku 2.6 .

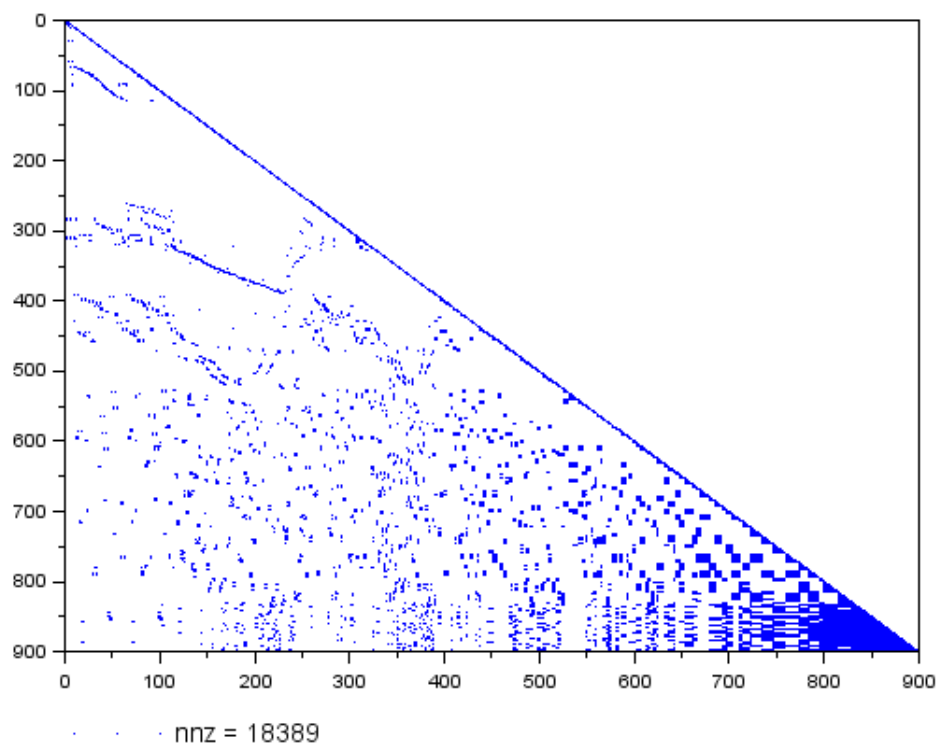
Porovnáme-li počet nenulových prvků v prvním a ve druhém případě, vidíme výrazný pokles. V prvním případě mají faktory  $L$  a  $U$  počet nnz 27 870 a ve druhém případě mají faktory  $L$  a  $U$  počet nnz 18 389. Pokles nenulových prvků je zapříčiněn vhodnou volbou permutačních matic.

Ve třetím způsobu změním parametry na  $[0, 0.5]$  a zadáme příkaz  $hand = lu\text{fact}(As, [0, 0.5])$ , poté  $[P, L, U, Q] = lu\text{get}(hand)$ . Pro zobrazení grafického umístění nenulových prvků matic  $L$  a  $U$  zadáme opět  $PlotSparse(L)$  a  $PlotSparse(U)$ . Nenulové prvky matice  $L$  jsou zobrazeny na obrázku 2.7 . Nenulové prvky matice  $U$  jsou zobrazeny na obrázku 2.8 .

Můžeme si všimnout, jak změna parametrů  $[eps, reps]$  výrazně ovlivní počet nenulových prvků. Kvůli stabilitě jsme v tomto pokusu dostali více nenulových prvků než *bez* permutací.

---

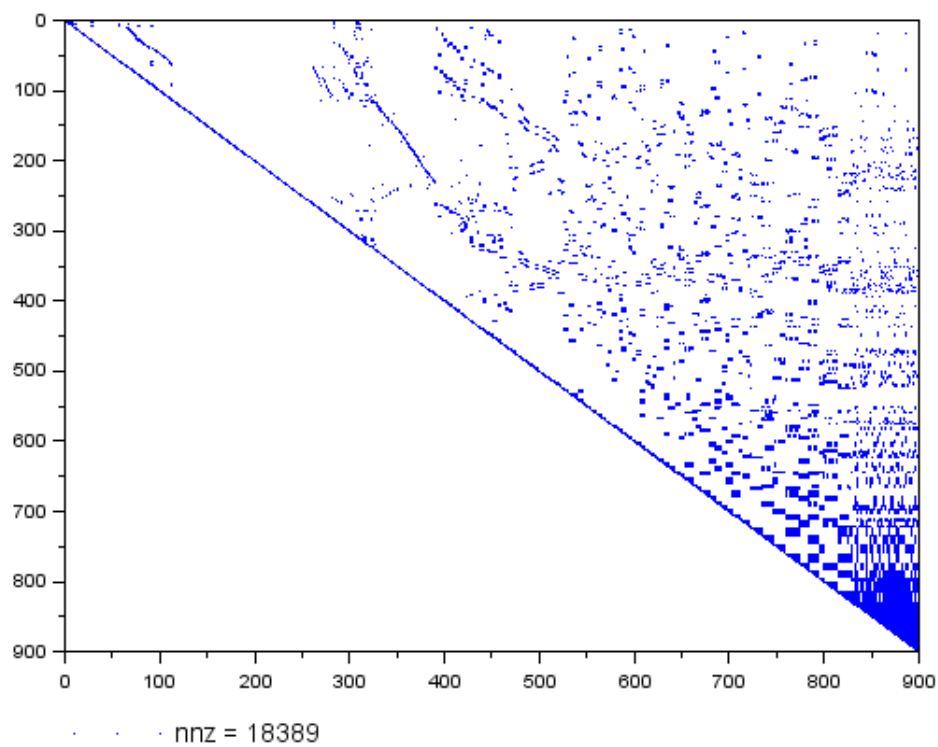
### Řídká matice L - s permutací - parametry [0,0.1]



Obrázek 2.5: Řídká matice L - s permutací - parametry [0,0.1]

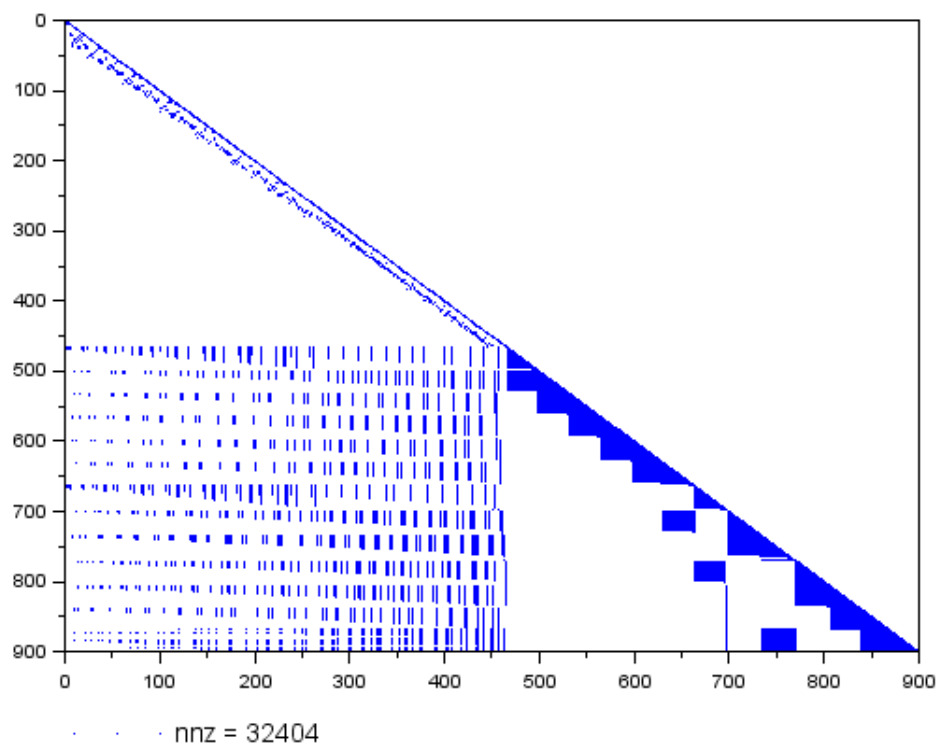
---

## Řídká matice U - s permutací - parametry [0,0.1]



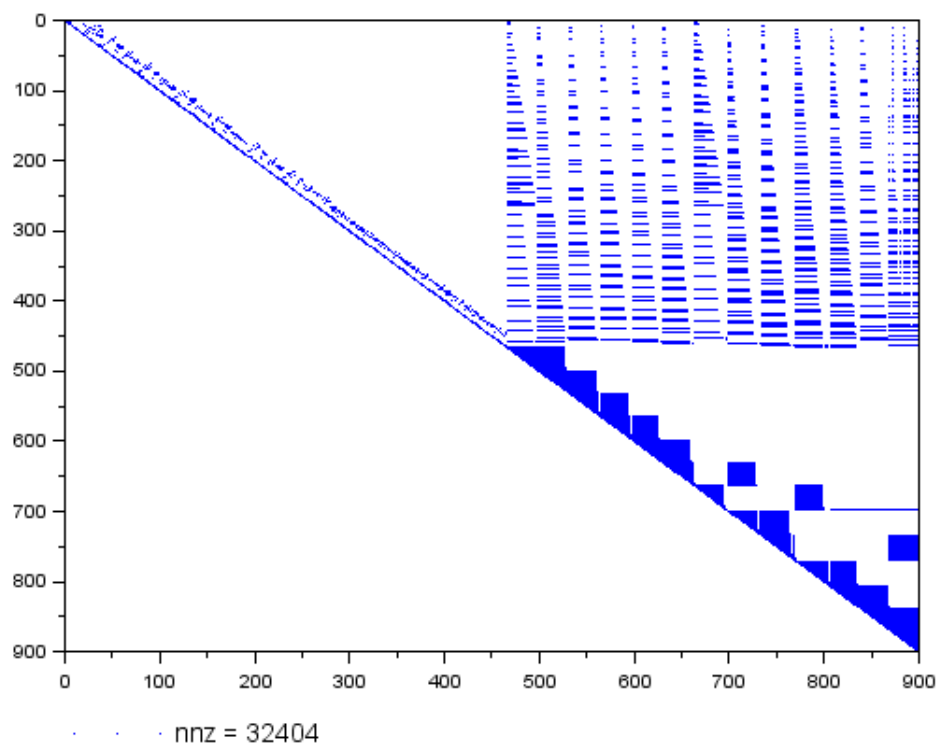
Obrázek 2.6: Řídká matice U - s permutací - parametry [0,0.1]

## Řídká matice L - s permutací - parametry [0,0.5]



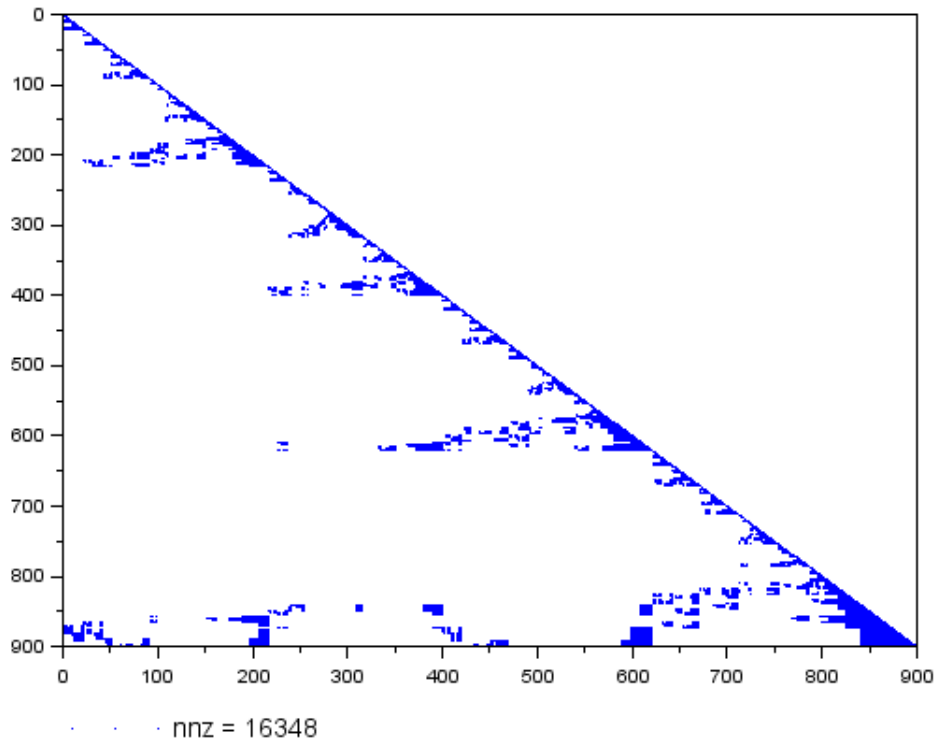
Obrázek 2.7: Řídká matice L - s permutací - parametry [0,0.5]

## Řídká matice U - s permutací - parametry [0,0.5]



Obrázek 2.8: Řídká matice U - s permutací - parametry [0,0.5]

## Řídká matice L - UMFPACK



Obrázek 2.9: Řídká matice L - UMFPACK

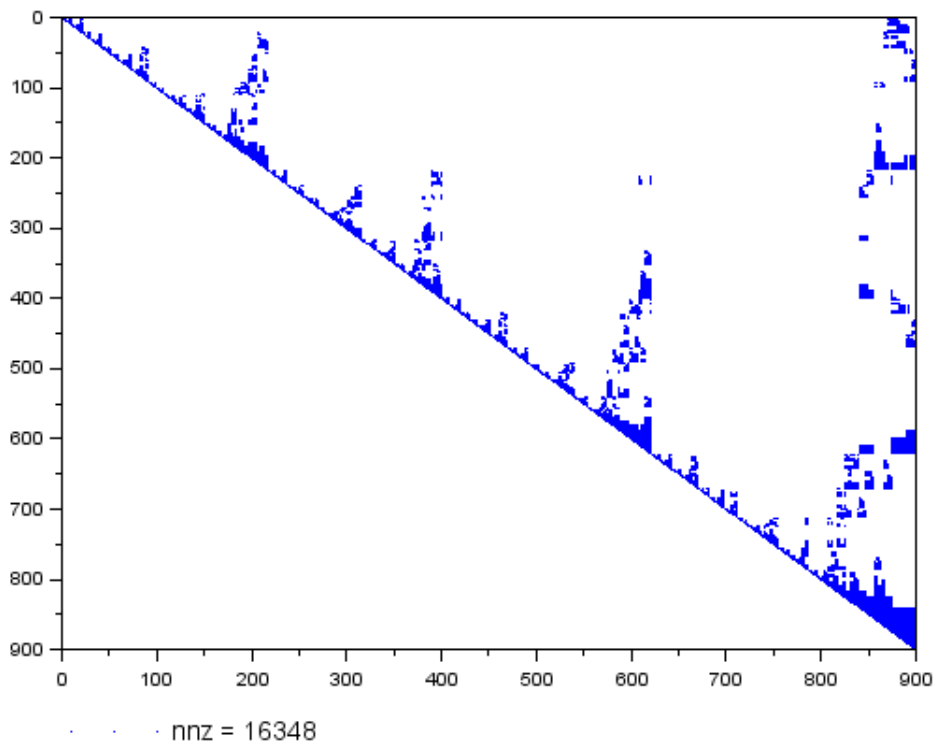
Ve čtvrtém způsobu využijeme komerčně vytvořený balíček *UMFPACK*, který je v dnešní době na velmi dobré úrovni, jelikož jeho vývinem se odborníci zabývají už alespoň dvacet let. Zadáme příkaz  $LU\_ptr = umf\_lu\,fact(As)$  a poté  $[L, U, p, q, Rd] = umf\_lu\,get(LU\_ptr)$ . Pro grafové zobrazení nenulových prvků matic  $L$  a  $U$  použijeme  $PlotSparse(L)$  a  $PlotSparse(U)$ .

Nenulové prvky matice  $L$  jsou zobrazeny na obrázku 2.9. Nenulové prvky matice  $U$  jsou zobrazeny na obrázku 2.10. Ze všech pokusů tento příkaz dává nejnižší počet nenulových prvků.

Každým výše popsaným způsobem dojdeme ke správnému řešení (vážený problém se stabilitou v tomto příkladu není), a proto nás zajímá ten nejefektivnější, při kterém dochází ke vzniku nejméně nenulových prvků. Porovnání počtů nenulových prvků najdeme v tabulce 2.3.

Ze všech čtyřech způsobů, které program Scilab nabízí, je poslední způsob s

## Řídká matice U - UMFPACK



Obrázek 2.10: Řídká matice U - UMFPACK

Tabulka 2.3: Porovnání počtu nenulových prvků

	As	L	U
Bez permutací	7 744	27 870	27 870
S permutací [0, 0.1]	7 744	18 389	18 389
S permutací [0, 0.5]	7 744	32 404	32 404
UMFPACK	7 744	16 348	16 348

---

použitím příkazu  $LU\_ptr = umf\_lu\,fact(A_s)$  ten nejefektivnější.



## Kapitola 3

# Krylovovské iterační metody

Jako v předchozí kapitole řešíme i tady systém lineárních rovnic tvaru  $Ax = b$ , kde  $A \in \mathbb{R}^{n \times n}$  je regulární řídká matice a  $b \in \mathbb{R}^n$ . Na rozdíl od řešení pomocí  $LU$  rozkladu nás nemusí zajímat přesné řešení  $x$  (to znamená přesné na úrovni strojové přesnosti). Cílem iteračních metod je získání dobré aproximace  $x_k$  přesného řešení  $x$  v co nejmenším počtu iterací, a proto v co nejmenším počtu aritmetických operací. Informace pro tuto kapitolu čerpáme především z literatury [4].

### 3.1 Krylovovské podprostory a jejich báze

V odstavci 2.5 jsme popsali výhody řídkých matic. Je-li zadána řídká matice  $A$  a vektor  $b$ , pak můžeme levné násobení matice s vektorem využít k sestavení tzv. posloupnosti Krylovovských podprostorů. Výhody těchto podprostorů spočívají mezi jinými v opětovném násobení vektoru  $b$  s řídkou maticí  $A$ , jak je vidět v následující definici.

**Definice 3.1.1** *Pro  $k \leq n$  je  $k$ -tý Krylovovský podprostor značený  $\mathcal{K}_k$  definován jako*

$$\mathcal{K}_k(A, b) \equiv \text{span}\{b, Ab, A^2b, A^3b, \dots, A^{k-1}b\}.$$

#### 3.1.1 Gram-Schmidtův ortogonalizační proces

Krylovovské metody pracují s bázemi pro Krylovovské podprostory.

Vektory  $\{b, Ab, A^2b, A^3b, \dots, A^{k-1}b\}$  jsou velmi často numericky lineárně závislé již

pro malé  $k$ , i když prostor samotný má v přesné aritmetice plnou dimenzi. Kdybychom je používali jako báze, vedlo by to ke značné ztrátě numerické stability výpočtu. Tento problém vyřešíme pomocí Gram-Schmidtovy ortogonalizace, kdy v průběhu vznikají ortogonální (kolmé) báze pro  $\mathcal{K}_k(A, b)$ , dokonce jsou ortonormální, protože jsou normalizované (velikost vektorů je jedna). Ortogonalizační proces uvedeme na obecném příkladu (ne nutně Krylovovském příkladu) pro znázornění pro podprostor tvořený třemi vektory.

Mějme podprostor prostoru  $\mathbb{R}^n$  dimenze 3,

$$\text{span}\{v_1, v_2, v_3\}, v_i \in \mathbb{R}^n.$$

Pokud vektory tvořící podprostor nejsou na sebe kolmé, pak pomocí Gram-Schmidtovy ortogonalizace můžeme vytvořit ortogonální bázi  $q_1, q_2, q_3$  splňující

$$\text{span}\{q_1, q_2, q_3\} = \text{span}\{v_1, v_2, v_3\}.$$

V prvním kroku ortogonalizačního procesu hledáme vektor  $q_1$ , za podmínky  $\|q_1\| = 1$ ,  $\text{span}\{q_1\} = \text{span}\{v_1\}$ . Vektor  $q_1$  snadno získáme normalizací vektoru  $v_1$ :

$$q_1 = \frac{v_1}{\|v_1\|}.$$

Ve druhém kroku hledáme vektor  $q_2$  za podmínek  $\text{span}\{q_1, q_2\} = \text{span}\{v_1, v_2\}$ ,  $\|q_2\| = 1$  a zároveň  $q_1 \perp q_2$ . Pro splnění podmínek stačí odečíst projekci vektoru  $v_2$  ve směru  $q_1$  od  $v_2$ , čímž získáme přechodný vektor  $z_1$ . Normalizací vektoru  $z_1$  získáme pak vektor  $q_2$ :

$$z_1 = v_2 - (q_1^T v_2)q_1,$$

$$q_2 = \frac{z_1}{\|z_1\|}.$$

Že jsou vektory  $q_1$  a  $q_2$  navzájem opravdu kolmé, můžeme snadno dokázat pomocí skalárního součinu  $q_1^T z_1$ :

$$\begin{aligned} q_1^T z_1 &= \frac{v_1^T}{\|v_1\|} \cdot (v_2 - (q_1^T v_2)q_1) = \frac{v_1^T v_2}{\|v_1\|} - (q_1^T v_2) \cdot \frac{v_1^T q_1}{\|v_1\|} \\ &= \frac{v_1^T v_2}{\|v_1\|} - \frac{v_1^T v_2}{\|v_1\|} \cdot \frac{v_1^T v_1}{\|v_1\|^2} = 0. \end{aligned}$$

---

Je-li  $q_1^T z_1 = 0$ , pak je i  $q_1^T q_2 = \frac{q_1^T z_1}{\|z_1\|} = 0$ .

Ve třetím kroku ortogonalizačního procesu hledáme vektor  $q_3$  za podmínek  $\text{span}\{q_1, q_2, q_3\} = \text{span}\{v_1, v_2, v_3\}$ ,  $\|q_3\| = 1$  a  $q_1 \perp q_2$  a zároveň  $q_1 \perp q_3$ . Pro splnění podmínek opět stačí odečíst projekci vektoru  $v_3$  ve směru  $q_1$  a  $q_2$  od  $v_3$ , tím získáme vektor  $z_2$ , který poté normalizujeme a získáme vektor  $q_3$ :

$$z_2 = v_3 - (q_1^T v_3)q_1 - (q_2^T v_3)q_2,$$
$$q_3 = \frac{z_2}{\|z_2\|}.$$

Uvažovali bychom příklad

$$\text{span}\{v_1, v_2, \dots, v_k\},$$

pak by  $k$ -tý krok ortogonalizačního procesu vypadal následovně:

$$z_{k-1} = v_k - \sum_{i=1}^{k-1} (q_i^T v_k) q_i,$$
$$q_k = \frac{z_{k-1}}{\|z_{k-1}\|}.$$

V průběhu ortogonalizačního procesu dochází ke vzniku vztahů, které jsou podstatné pro maticový zápis, ve kterém lze matice se sloupci  $[v_1, v_2, v_3]$  rozložit na matici  $Q \in \mathbb{R}^{n \times k}$ , která je ortonormální a matici  $R \in \mathbb{R}^{k \times k}$ , která má horní trojúhelníkový tvar a jejíž nenulové prvky vyplývají ze skalárních součinů (a norem) spočítaných při ortogonalizačním procesu.

**Definice 3.1.2** *QR rozklad matice  $A \in \mathbb{R}^{n \times k}$  je rozklad tvaru  $A = QR$ , kde matice  $Q \in \mathbb{R}^{n \times k}$  je ortonormální a matice  $R \in \mathbb{R}^{k \times k}$  je horní trojúhelníková matice.*

Pro matici  $A = (v_1 \ v_2 \ v_3)$  se sloupci  $v_1, v_2, v_3$  má  $QR$  rozklad tvar

$$A = (v_1 \ v_2 \ v_3) = (q_1 \ q_2 \ q_3) \cdot \begin{pmatrix} \|v_1\| & q_1^T v_2 & q_1^T v_3 \\ 0 & \|z_1\| & q_2^T v_3 \\ 0 & 0 & \|z_2\| \end{pmatrix} = QR.$$

### 3.1.2 Arnoldiho algoritmus

Pro ortogonalizaci vektorů Krylovovského podprostoru

$$\mathcal{K}_k(A, b) \equiv \text{span}\{b, Ab, A^2b, A^3b, \dots, A^{k-1}b\}$$

použijeme Gram-Schmidtův ortogonalizační proces určitého tvaru, který se nazývá *Arnoldiho algoritmus*. Postup si ukážeme na příkladu

$$\mathcal{K}_3(A, b) \equiv \text{span}\{b, Ab, A^2b\}.$$

V prvním kroku ortogonalizačního procesu hledáme vektor  $q_1$ , za podmínky  $\|q_1\| = 1$   $\text{span}\{q_1\} = \text{span}\{b\}$ . Vektor  $q_1$  získáme normalizací vektoru  $b$ :

$$q_1 = \frac{b}{\|b\|}.$$

Ve druhém kroku hledáme vektor  $q_2$  za podmínek  $\text{span}\{q_1, q_2\} = \text{span}\{b, Ab\}$ ,  $\|q_2\| = 1$  a zároveň  $q_1 \perp q_2$ . Pro splnění podmínek odečteme projekci vektoru  $Aq_1$  ve směru  $q_1$  od  $Aq_1$ , čímž získáme vektor  $z_1$ . Na rozdíl od Gram-Schmidtova procesu se zde násobí první bázevský vektor s maticí  $A$ . Normalizací vektoru  $z_1$  získáme pak vektor  $q_2$  :

$$z_1 = Aq_1 - (q_1^T Aq_1)q_1,$$

$$q_2 = \frac{z_1}{\|z_1\|}.$$

Že jsou vektory  $q_1$  a  $q_2$  navzájem opravdu kolmé, můžeme snadno dokázat pomocí skalárního součinu  $q_1^T z_1$ :

$$\begin{aligned} q_1^T z_2 &= \frac{b^T}{\|b\|} \cdot (Aq_1 - (q_1^T Aq_1)q_1) = \frac{b^T Aq_1}{\|b\|} - (q_1^T Aq_1) \cdot \frac{b^T q_1}{\|b\|} \\ &= \frac{b^T Aq_1}{\|b\|} - \left( \frac{b^T Aq_1}{\|b\|} \right) \cdot \frac{b^T b}{\|b\|^2} = 0. \end{aligned}$$

Je-li  $q_1^T z_1 = 0$ , pak je i  $q_1^T q_2 = \frac{q_1^T z_1}{\|z_1\|} = 0$ .

Ve třetím kroku ortogonalizačního procesu hledáme vektor  $q_3$  za podmínek  $\text{span}\{q_1, q_2, q_3\} = \text{span}\{b, Ab, A^2b\}$ ,  $\|q_3\| = 1$  a  $q_1 \perp q_2$  a zároveň  $q_1 \perp q_3$ . Pro splnění podmínek odečteme projekci vektoru  $Aq_2$  ve směru  $q_1$  a  $q_2$  od  $Aq_2$ , tím získáme vektor  $z_2$ . Na rozdíl od Gram-Schmidtova procesu se zde násobí druhý bázevský vektor s maticí  $A$ . Lze snadno dokázat, že  $\text{span}\{b, Ab, A^2b\} = \text{span}\{q_1, Aq_1, Aq_2\}$ , viz [4, cvičení 3.18]. Jeho normalizací získáme vektor  $q_3$ :

$$z_2 = Aq_2 - (q_1^T Aq_2)q_1 - (q_2^T Aq_2)q_2,$$

$$q_3 = \frac{z_2}{\|z_2\|}.$$

Uvažovali bychom příklad

$$\text{span}\{b, Ab, \dots, A^{k-1}b\},$$

pak by  $k$ -tý krok ortogonalizačního procesu Arnoldiho algoritmu vypadal následovně:

$$z_{k-1} = Aq_{k-1} - \sum_{i=1}^{k-1} (q_i^T Aq_{k-1})q_i,$$

$$q_k = \frac{z_{k-1}}{\|z_{k-1}\|}.$$

Báze složené z vektorů  $q_1, q_2, \dots, q_k$  se nazývá *Arnoldiho báze*. Proces spočte  $QR$  rozklad

$$(3.2) \quad [b, AQ_k] = Q_{k+1}R_{k+1}.$$

Pro konkrétní případ

$$\text{span}\{b, Ab, A^2b\},$$

vypadá  $QR$  rozklad jako

$$[b, AQ_2] = (b \quad Aq_1 \quad Aq_2) = (q_1 \quad q_2 \quad q_3) \cdot \begin{pmatrix} \|b\| & q_1^T Aq_1 & q_1^T A^2q_2 \\ 0 & \|z_1\| & q_2^T A^2q_2 \\ 0 & 0 & \|z_2\| \end{pmatrix} = Q_3R_3.$$

Jestliže vynecháme první sloupec a poslední řádek matice  $R_{k+1}$ , dostaneme čtvercovou matici  $H_k$  zvanou *Hessenbergovu matici*. Schematicky Hessenbergova matice  $H_k \in \mathbb{R}^{k \times k}$  vypadá následovně

$$H_k = \begin{pmatrix} * & * & * & \cdots & * \\ * & * & * & \cdots & * \\ 0 & * & * & \cdots & * \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & * & * \end{pmatrix}.$$

Vynecháním prvního sloupce ze (3.2) dostaneme takzvaný *Arnoldiho rozklad*

$$(3.3) \quad AQ_k = Q_{k+1}H_{k+1,k},$$

kde  $H_{k+1,k} \in \mathbb{R}^{(k+1) \times k}$ , je matice  $H_{k+1}$  bez posledního sloupce.

## 3.2 Konkrétní Krylovovské metody

Krylovovské metody zvolí v  $k$ -té iteraci  $k$ -tou aproximaci  $x_k$  řešení  $x$  v podprostoru  $\mathcal{K}_k(A, b)$ , to je

$$x_k \in \mathcal{K}_k(A, b).$$

Zde předpokládáme, že počáteční aproximace  $x_0$  je nulový vektor.

Jednotlivé metody se liší v podmínkách pro jednoznačné určení  $x_k \in \mathcal{K}_k(A, b)$ . Všechny metody mají podmínky nastavené takovým způsobem, že v  $n$ -té iteraci je zaručeno, že  $x_n = x$ . Konkrétní příklad je prezentován pro metodu GMRES. Přesný popis jednotlivých metod je nad rámec této práce. Uvedeme ale pár vlastností nejpulárnějších metod.

První metoda, metoda konjugovaných gradientů [16], je určena pro symetrické pozitivní definitní matice  $A$  (to je  $A = A^T$  a  $x^T Ax > 0$  pro  $x \neq 0$ ). V takovém případě je ve vztahu (3.3) Hessenbergova matice  $H_{k+1,k}$  symetrická, a tudíž tridiagonální. To znamená, že v Arnoldiho algoritmu je velká většina koeficientů nulových, pro vytvoření ortogonální báze stačí spočítat skalární součiny pouze se dvěma předchozími bázeovými vektory.

Druhá metoda, GMRES (Generalized Minimal Residual Method) [21], je určena pro obecné regulární matice. V tomto případě je Arnoldiho algoritmus poměrně drahý. V  $k$ -té iteraci je třeba spočítat  $k$  skalárních součinů (s vektory  $q_1, \dots, q_{k-1}$  + normalizace). Aproximace  $x_k$  v  $k$ -té iteraci je charakterizovaná podmínkou

$$(3.4) \quad x_k = \arg \min_{x \in \mathcal{K}_k(A, b)} \|b - Ax\|.$$

V  $k$ -té iteraci se minimalizuje norma  $k$ -tého rezidua  $r_k = b - Ax_k$ . Jelikož  $\mathcal{K}_n(A, b) = \mathbb{R}^n$ , je jasné, že v  $n$ -té iteraci je reziduální norma nulová, a tudíž  $x_n = x$ . Navíc je jasné, že s rostoucími podprostory klesají reziduální normy. Přesněji, během iteračního procesu jsou reziduální normy nerostoucí. Metoda GMRES použije pro řešení minimalizačního problému Arnoldiho rozklad (3.3) následovně:

$$(3.5) \quad \begin{aligned} \|r_k\| &= \|b - Ax_k\| = \left\| Q_{k+1} \|b\| e_1 - A Q_k \begin{pmatrix} y_1 \\ \vdots \\ y_k \end{pmatrix} \right\| = \left\| Q_{k+1} \|b\| e_1 - Q_{k+1} H_{k+1,k} y_k \right\| \\ &= \left\| Q_{k+1} (\|b\| e_1 - H_{k+1,k} y_k) \right\| = \left\| \|b\| e_1 - H_{k+1,k} y_k \right\|. \end{aligned}$$

---

Poslední rovnost platí, protože  $Q_{k+1}$  je ortonormální, viz (1.1).

Poslední výraz představuje malou soustavu dimenze  $(k+1) \times k$  (problém nejmenších čtverců), kde hledaný vektor je  $y_k$ . Celý postup metody GMRES včetně Arnoldiho algoritmu můžeme shrnout v podobě následujícího pseudoalgoritmu. Použili jsme příklad z knihy [20].

Vstup:  $A, b$ , výstup:  $k$ -tá aproximace  $x_k$ .

1. Normalizace:  $\beta = \|b\|$ ,  $q_1 = \frac{b}{\beta}$
2. Definujeme matici  $H_k \in \mathbb{R}^{(k+1) \times k}$  s prvky  $h_{ij}$ . Položíme  $H_k = 0$ .
3. Pro  $j = 1, 2, \dots, k$  počítáme:

$$z_j := Aq_j$$

Pro  $i = 1, \dots, j$  počítáme:

$$h_{ij} := z_j q_i$$

$$z_j = z_j - h_{ij} q_i$$

$$h_{j+1,j} = \|z_j\|_2. \text{ Jestliže } h_{j+1,j} = 0, \text{ pak } k := j$$

$$q_{j+1} = \frac{z_j}{h_{j+1,j}}$$

4. Najdeme  $y_k$  minimalizující  $\|\beta e_1 - H_k y\|_2$  a definujeme  $x_k = Q_k y_k$ .

Třetí metoda, QMR (quasi-minimal residual)[13], je založená na procesu Bi-Lanczos, kdy je vektor pouze ortogonalizován vůči dvěma předchozím vektorům a výsledná báze je "bi-ortogonální". Používá postup (3.5) jako GMRES, i když poslední rovnost tím pádem neplatí. Neminimalizují se reziduální normy, a proto křivky postupných reziduálních norem nejsou nerostoucí. Čtvrtá metoda, Bi-CG [12, 18], nepoužívá postup (3.5), ale řeší  $H_k y_k = \|b\| e_1$  ( $H_k$  je čtvercová).

Pátá metoda, Bi-CGStab [23], je nejslavnější z metod pro obecné matice na základě procesu Bi-Lanczos. Postupuje obdobně jako čtvrtá metoda, ale dochází k použití stabilizací, což má často za následek hladší křivky.

### 3.3 Výpočetní náklady

Počet aritmetických operací Krylovovských iteračních metod závisí zřejmě na počtu iterací. Víme, že každá iterace vyžaduje násobení s maticí. Pokud má  $nnz$  nenulových

---

prvků, pak jedno násobení odpovídá  $2nnz$  operací. To znamená, že pro  $k$  iterací odpovídá násobení matice s vektory pro vytvoření báze Krylovovských podprostorů celkem  $2k \cdot nnz$  operací.

V případě metody GMRES (a jiných, které zde neuvedeme) roste s číslem iterací navíc počet operací nutných pro ortogonalizaci báze vektorů a pro vyřešení systému s Hessenbergovou maticí. Ostatní probrané metody mají výpočetní náklady, které nejsou závislé na čísle iterace, ale jsou obvykle méně stabilní.

### 3.4 Krylovovské podprostory pomocí programu Scilab

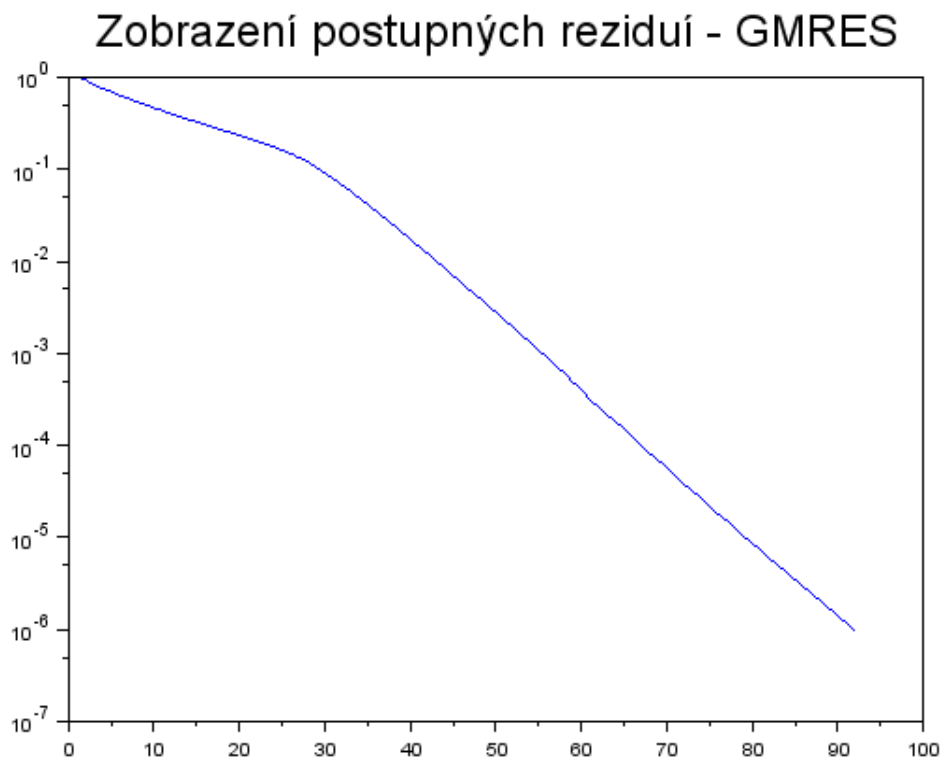
Jako v předchozí kapitole  $LU$  rozkladu i tady používáme program Scilab.

Chceme vyřešit soustavu prezentovanou na konci druhé kapitoly spojenou s konvexním-difúzním problémem s maticí  $A \in \mathbb{R}^{900 \times 900}$  a pravou stranou  $b \in \mathbb{R}^{900 \times 1}$ . Matice  $A$  není symetrická a pozitivně definitní. Matici  $A$  uložíme jako  $As$ , což značí uložení matice jako řídké (uložíme pouze nenulové prvky). Připomeneme, že umístění nenulových prvků matice  $As$  je vidět na obrázku 2.1.

Vyzkoušíme čtyři Krylovovské metody (GMRES, QMR, Bi-CG, Bi-CGStab). Pro výpočet GMRES existuje příkaz  $[x, flag, err, iter, res] = gmres(As, b, [maxi])$ , kde  $x$  je aproximace řešení systému,  $flag$  označuje, zda se podařilo najít aproximaci s danou tolerancí reziduální normy  $10^{-6}$ ,  $err$  označuje dosaženou reziduální normu,  $iter$  označuje počet iterací,  $res$  je vektor obsahující postupné reziduální normy,  $maxi$  určuje maximální počet iterací. V našem případě  $maxi = 900$ . Zadáme příkaz  $[x, flag, err, iter, res] = gmres(As, b, [900])$ , získáme pro nás důležité hodnoty  $flag = 0$ , z toho vyplývá, že se podařilo najít aproximaci,  $iter = 91$ . Pro grafové zobrazení průběhu iterací zadáme příkaz  $Plot([1 : 92], res)$ . Graf je zobrazen na obrázku 3.1. Vidíme, že křivka není rostoucí, což bylo dokázáno v předchozím odstavci.

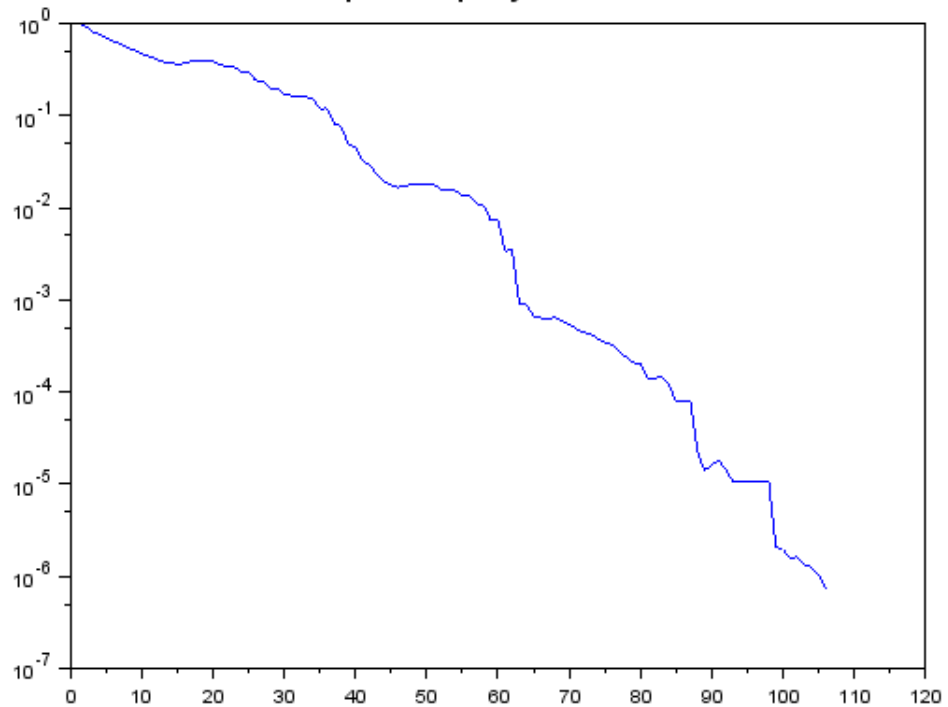
Pro výpočet pomocí QMR zadáme příkaz  $[x, flag, err, iter, res] = qmr(As, b, x0, M, M, maxi, tol)$ , kde  $x0$  označuje počáteční aproximaci, ta je vždy nulová,  $M, M$  jsou pomocné matice, které jsme zvolili jako jednotkové a  $tol$  je tolerance, kterou jsme zvolili  $10^{-6}$ . Získali jsme pro nás důležité hodnoty  $flag = 0$ , z toho vyplývá, že se podařilo najít aproximaci,  $iter = 105$ . Můžeme vidět, že počet iterací je zde vyšší než při využití GMRES. Pro grafové zobrazení použijeme příkaz  $Plot([1 : 106], res)$ .





Obrázek 3.1: Zobrazení postupných reziduí - GMRES

## Zobrazení postupných reziduí - QMR

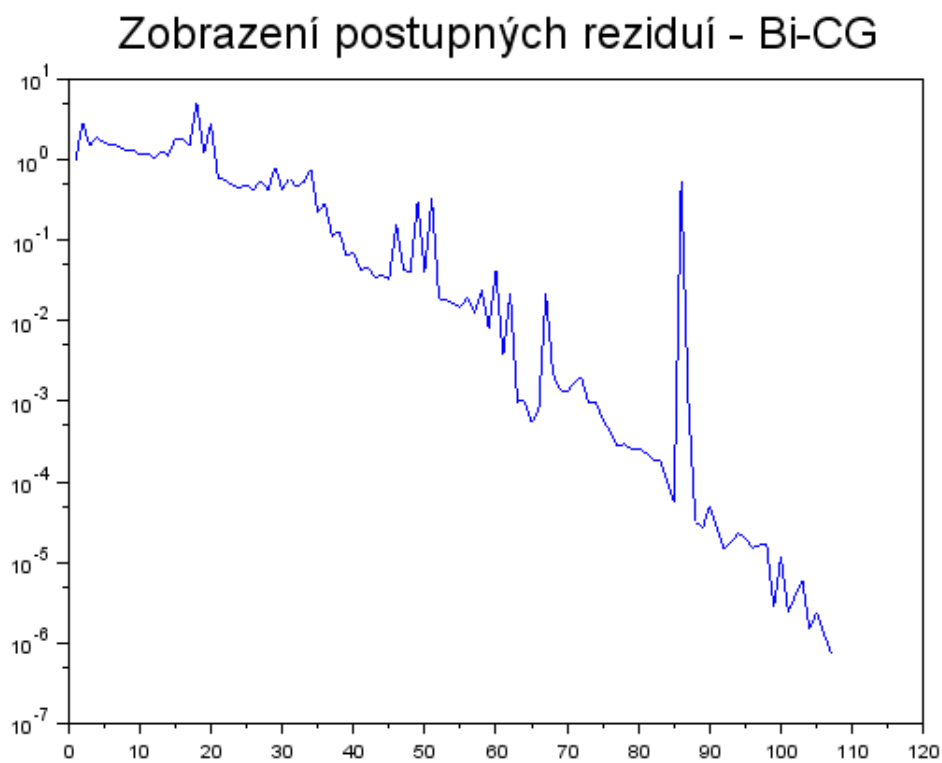


Obrázek 3.2: Zobrazení postupných reziduí - QMR

Graf je zobrazen na obrázku 3.2. Vidíme, že je křivka poměrně hladká, ale občas se vyskytují rostoucí normy.

Pro výpočet pomocí metody Bi-CG zadáme příkaz  $[x, flag, err, iter, res] = conjgrad(As, b, "bicg", tol, maxi)$ . Tolerance zůstává stejná  $tol = 10^{-6}$  a parametr  $maxi = 900$ . Získáme opět důležité hodnoty  $flag = 0$ , z toho vyplývá, že se podařilo najít aproximaci,  $iter = 106$ . Pro grafové zobrazení použijeme příkaz  $Plot([1 : 107], res)$ . Graf je zobrazen na obrázku 3.3. Křivka jednoznačně není nerostoucí.

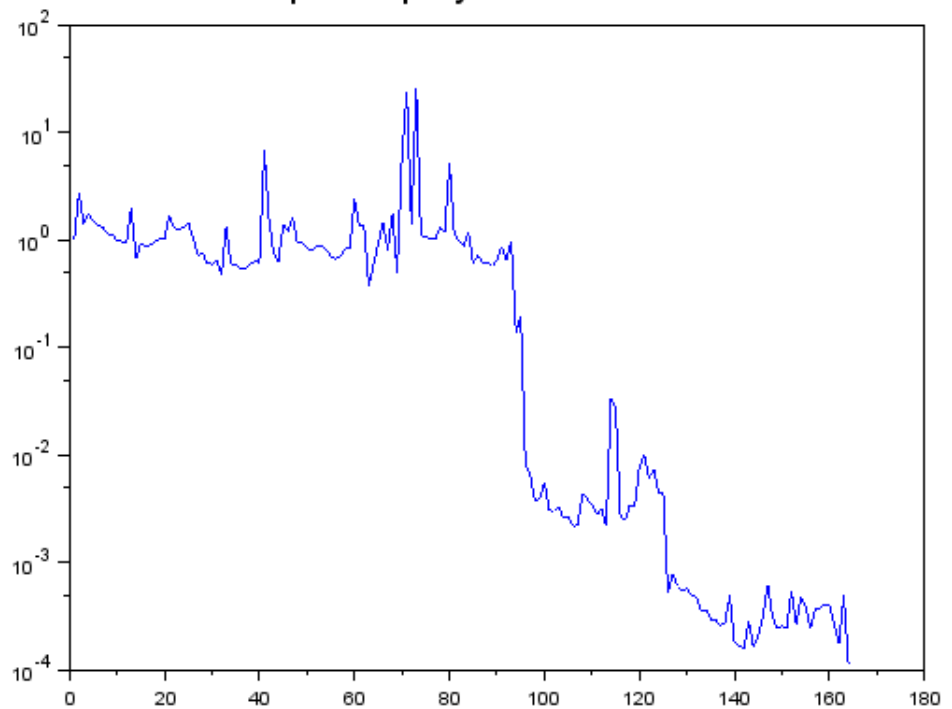
Pro výpočet pomocí metody Bi-CGStab zadáme příkaz  $[x, flag, err, iter, res] = conjgrad(As, b, "bicgstab", tol, maxi)$ . Tolerance zůstává stejná  $tol = 10^{-6}$  a  $maxi = 900$ . Po zadání příkazu získáme důležité hodnoty  $flag = 0$  z toho vyplývá, že se podařilo najít aproximaci,  $iter = 163$ . Pro grafové zobrazení použijeme příkaz  $Plot([1 : 164], res)$ . Graf je zobrazen na obrázku 3.4. Křivka absolutně není hladká,



Obrázek 3.3: Zobrazení postupných reziduí - Bi-CG

---

## Zobrazení postupných reziduí - Bi-CGStab



Obrázek 3.4: Zobrazení postupných reziduí - Bi-CGStab

obsahuje mnoho rostoucích norem.

Ve všech způsobech výpočtu si můžeme zvolit toleranci. Čím větší si zvolíme v dané metodě toleranci, tím bude počet iterací větší. My jsme zvolili toleranci ve všech případech stejnou  $tol = 10^{-6}$  a počet iterací se s každou metodou měnil.

## Kapitola 4

# Porovnání efektivity a závěr

V této závěrečné kapitole velmi stručně porovnáme efektivitu  $LU$  rozkladu s efektivitou Krylovovských metod pro řešení soustav lineárních rovnic s řídkými maticemi. Efektivita metod v numerické lineární algebře se často charakterizuje na základě tří vlastností:

1. Počet (řádivý) aritmetických operací potřebných k nalezení řešení
2. Paměťové nároky
3. Numerická stabilita vzhledem k zaokrouhlovacím chybám

### 4.1 $LU$ rozklad

Výpočetní náklady  $LU$  rozkladu plné matice jsou řádově  $n^3$ , ale pro řídké matice mohou být výrazně nižší. Například pro tridiagonální jsou řádově  $\mathcal{O}(n)$  operace. Záleží velice na volbě vhodné permutační matice. Pokud se například povede permutovat tak, aby výsledná matice byla pásová (měla nějaký nízký počet nenulových diagonál okolo hlavní diagonály), pak příslušné faktory  $L$  a  $U$  jsou též pásové. Je těžké říct dopředu, jaké budou výpočetní náklady  $LU$  rozkladu dané řídké matice s nestrukturovaným umístěním nenulových prvků. Ve velkém počtu případů se ale povede snížit náklady z  $n^3$  pro plné matice na  $n^2$  pro řídké matice.

Paměťové náklady jsou podobným způsobem vázány na otázku, zda je možné najít vhodnou permutační matici. To nelze říct předem a jednou důležitou nevýhodou  $LU$  rozkladu je riziko, že výsledné faktory  $L$  a  $U$  nelze uložit v paměti počítači.

---

Stabilita je, jak jsme viděli, potenciálně velmi ohrožena, pokud nemůžeme pro danou matici prvky uložit s dostatečnou přesností. Existují reálné příklady, kde prvky ve faktoru  $U$  rostou exponenciálně v porovnání s prvky původní matice, a příslušný zpětný chod je pak velmi nestabilní. I tady je zásadní permutace kvůli stabilitě, ale častým problémem permutací bývá, že stabilní permutace nemusí dávat nízké zaplnění faktorů  $L$  a  $U$  a naopak minimalizace zaplnění pomocí permutací může vést k nestabilnímu rozkladu.

## 4.2 Krylovovské metody

Jak jsme popsali v odstavci 3.3, výpočetní náklady Krylovovských metod závisí na počtu iterací a hlavní náklady za iteraci jsou v násobení vektoru s maticí a pro určité metody též ve vytvoření báze pro Krylovovské podprostory. Počet iterací závisí na požadované redukci reziduální normy a ta silně závisí na dané aplikaci. V aplikacích jako řešení diferenciálních rovnic vznikají různé aproximační chyby, například z důvodu diskretizace. Nemá smysl řešit příslušnou lineární soustavu o mnoho přesněji než je dosažitelná přesnost vzhledem k diskretizaci. Tady je zásadní výhoda oproti řešení pomocí  $LU$  rozkladu:  $LU$  rozklad není možné zastavit před koncem, žádné aproximace řešení nejsou průběžně k dispozici. Pokud se Krylovovské metodě povede najít rozumnou aproximaci v malém počtu  $k$  kroků, pak celkové náklady jsou řádově  $\mathcal{O}(k \cdot nnz)$  operací, což může být výrazně méně než běžné  $\mathcal{O}(n^2)$  náklady pro  $LU$  rozklad. Otázku, kolik je potřeba dělat iterací pro dosažení dané reziduální normy, řeší takzvaná teorie konvergence. Pro symetrické matice je teorie konvergence založena na takzvaných vlastních číslech matice. Pro nesymetrické matice to tak obecné není a často se dobře neví, jak konvergenci popsat [5]. Pro nízký počet potřebných iterací se často používají takzvané předpokmiňovače, to je matice  $M$  taková, že řešení modifikované soustavy  $MAx = Mb$  je pro Krylovovské metody jednodušší.

Paměťové náklady jsou pro metody se třemi bázovými vektory nízké, protože v zásadě stačí uložit matici  $A$  a několik málo vektorů. Existují aplikace, které jsou schopny zjistit, jaký je výsledek násobení matice  $A$  s vektory bez znalostí prvků matice  $A$ , které tím pádem nemusíme uložit. V takovém případě se mluví o matrix-free

---

prostředí. Pro metodu GMRES, která vyžaduje navíc uložení celé báze aktuálního Krylovovského podprostoru, je možné zrestartovat proces po několika málo krocích a pokračovat od získané aproximace. Tím omezíme maximální velikost uložené báze. Nejstabilnější Krylovovské metody jsou ty, které používají ortonormální báze, jako metoda GMERS. Používají Arnoldiho algoritmus, který se zastaví po  $n$  krocích anebo pokud  $h_{j+i,j} = 0$ . Pokud  $h_{j+i,j} = 0$ , došlo k takzvanému lucky breakdown a lze dokázat, že tím pádem známe přesné řešení lineární soustavy. [4, cvičení 9.2]. U metod nepoužívajících Arnoldiho algoritmus existuje takzvaný lucky a serious breakdown. V posledním případě musíme proces ukončit; v praxi však serious breakdown často nevzniká. [4, Kapitola 9.4.1]

Oproti  $LU$  rozkladu mají Krylovovské metody výhodu, že mají reziduální normu  $\|b - Ax_k\|$  pod kontrolou, často jsou k dispozici jako vedlejší veličina výpočetního procesu a to pro každou iteraci  $k$ . Nemůže dojít k tomu, že se provedl celý proces až do konce a konečný výsledek je špatnou aproximací. U  $LU$  rozkladu se toto stát může, jak bylo znázorněno na konkrétním příkladu v odstavci 2.4. U Krylovovských metod na druhou stranu může konvergence být pomalá a celkové výpočetní náklady mohou přesáhnout  $\mathcal{O}(n^2)$  pro  $LU$  rozklad s řídkou maticí a vhodnou permutací.





# Literatura

- [1] AMESTOY, P., R., DUFF, I., S., L'EXCELLENT, J.-Y.: *Multifrontal parallel distributed symmetric and unsymmetric solvers*, Comput. Methods Appl. Mech. Engrg., (2000), st. 501-520.
- [2] DAVIS, A., T., DUFF, I., S.: *An unsymmetric-pattern multifrontal method for sparse LU factorization*, SIAM J. Matrix Anal. Appl., 18 (1997), st. 140-158.
- [3] DEMMEL, J., EISENSTAT, S., C., GILBERT, J., R., LI, X., S., LIU, J., W., H.: *A supernodal approach to sparse partial pivoting*, SIAM J. Matrix Anal. Appl., 20 (1999), st. 720-755.
- [4] DUINTJER TEBBENS, J., HNĚTYNKOVÁ, I., PLEŠINGER, M., STRAKOŠ, Z., TICHÝ, P.: *Analýza metod pro maticové výpočty - základní metody*, Matfyzpress, 1 (2012), 312 st. ISBN 978-80-7378-201-6.
- [5] DUINTJER TEBBENS, J., MEURANT, G.: *Any Ritz Value Behavior Is Possible for Arnoldi and for GMRES*, SIAM. J. Matrix Anal. and Appl., 33 (2012), st. 958–978 (elektronické).
- [6] ELMAN, H., C., RAMAGE, A.: *A characterisation of oscillations in the discrete two-dimensional convection-diffusion equation*, Math. Comp., 72 (2003), st. 263-288 (elektronické).
- [7] ELMAN, H., C., RAMAGE, A.: *An analysis of smoothing effects of upwinding strategies for the convection-diffusion equation*, SIAM J. Numer. Anal. 40 (2002), st. 254-281 (elektronické).
- [8] ELMAN, H., C., SILVESTER, D., J., WATHEN, A., J.: *Finite Elements and Fast Iterative Solvers : with Applications in Incompressible Fluid Dynamics: with Applications in Incompressible Fluid Dynamics*, OUP Oxford, 2005, st. 416, ISBN 019152378X, 9780191523786.
- [9] ERNST, O., G.: *Residual - minimizing Krylov subspace methods for stabilized discretizations of convection-diffusion equation*, SIAM J. Matrix Anal. Appl., 21 (2000), st. 1079-1101 (elektronické).

- 
- [10] FICHTNER, W., GÄRTNER, K., SCHENK, O.: *Efficient sparse LU factorization with left-right looking strategy on shared memory multiprocessors*, BIT, 40 (2000), st. 158-176.
- [11] FISHER, B., RAMAGE, A., SILVESTER, D., J., WATHEN, A., J.: *On parameter choice and iterative convergence for stabilised discretisations of advection-diffusion problems*, Comput. Methods Appl. Mech. Engrg., 179 (1999), st. 179-195.
- [12] FLETCHER, R.: *Conjugate gradient methods for indefinite systems*, in Numerical analysis (Proc 6th Biennial Dundee Conf., Univ. Dundee, Dundee, 1975), Springer, Berlin, 1976, st. 73-89. Lecture Notes in Math., Vol. 506.
- [13] FREUND, R., W., NACHTIGAL, N., M.: *QMR: a quasi-minimal residual method for non-Hermitian linear systems*, Numer. Math., 60 (1991), st. 315-339.
- [14] GILBERT, J., R., PEIERLS, T.: *Sparse partial pivoting in time proportional to arithmetic operations*, SIAM J. Sci. Statist. Comput., 9 (1988), st. 862-874.
- [15] GUPTA, A.: *Highly scalable parallel algorithms for sparse matrix factorization*, IEEE Trans. Parallel Distrib. System, (1997), st. 502-520.
- [16] HESTENES, M., R., STIEFEL, E.: *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards, 49 (1952), st. 409-436.
- [17] KLEMERA, P.: *Aplikovaná matematika - vybrané kapitoly pro studující farmacie*, Karolinum, 2 (2001), 91 st. ISBN 80-246-0303-9.
- [18] LANCZOS, C.: *Solution of systems of linear equations by minimized iterations*, J. Research Nat. Bur. Standards, 49 (1952), st. 33-53.
- [19] LIESEN, J., STRAKOŠ, Z.: *GMRES convergence analysis for a convection-diffusion model problem*, SIAM J. Sci. Comput., 26 (2005), st. 1989-2009 (elektronické).
- [20] SAAD, Y.: *Iterative methods for sparse linear systems*. PWS Publishing Company, 1996.
- [21] SAAD, Y., SCHULTZ, M., H.: *GMRES: a generalised minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), st. 856-869.
- [22] TURING, A., M.: *Rounding/off errors in matrix processes*, Q. J. Mech. Appl. Math., 1 (1948), st. 287-308.
- [23] VAN DER VORST, H., A.: *Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for solution of nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 13 (1992), st. 631-644.

- 
- [24] Microsoft, Excel, 2016,  
[https : //products.office.com/cs – cz/excel](https://products.office.com/cs-cz/excel)
- [25] Science and Technology, HSL, MA48 2.2.0, 2013,  
[http : //www.hsl.rl.ac.uk/catalogue/ma48.html](http://www.hsl.rl.ac.uk/catalogue/ma48.html)
- [26] Mathworks, Inc., MATLAB 8.5 , 2015,  
[http : //www.mathworks.com/products/matlab/](http://www.mathworks.com/products/matlab/)
- [27] John W. Eaton, OCTAVE 4.0.1, 2016,  
[https : //www.gnu.org/software/octave/](https://www.gnu.org/software/octave/)
- [28] Scilab Enterprises S.A.S, SCILAB 5.5.2, 2015,  
[http : //www.scilab.org/](http://www.scilab.org/)