MATEMATICKO-FYZIKÁLNÍ FAKULTA
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Benjamin de Silva

Nulový prostor matice a jeho použití v aplikacích

Katedra numerické analýzy

Vedoucí bakalářské práce: prof. Ing. Miroslav Tůma, CSc.
Studijní program: Matematika
Studijní obor: Obecná matematika

Praha 2024

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ....

Podpis autora

Děkuji panu profesorovi Tůmovi za ochotu, trpělivost a za jeho čas a své rodině za podporu.

Název práce: Nulový prostor matice a jeho použití v aplikacích

Autor: Benjamin de Silva

Katedra: Katedra numerické analýzy

Vedoucí bakalářské práce: prof. Ing. Miroslav Tůma, CSc., Katedra numerické matematiky

Abstrakt: Práce se zabývá metodami pro výpočet nulového prostoru matic. Nejprve odvodíme základní faktorizační metody a způsoby, jak z těchto faktorizací vypočítat nulový prostor matic. Pak popisujeme obecné numerické vlastnosti zavedených metod pro výpočet nulového prostoru matic. Na závěr tyto metody porovnáváme pomocí měření vzhledem k výpočetnímu času a k Frobeniově normě pro různé typy matic. Měření a vykreslení výsledků do grafů je realizováno v prostředí MATLAB.

Klíčová slova: matice, nulový prostor, faktorizace matic, metody pro výpočet nulového prostoru

Title: Null space of matrix and its use in applications

Author: Benjamin de Silva

Department: The department of numerical analysis

Supervisor: prof. Ing. Miroslav Tůma, CSc., Department of Numerical Mathematics

Abstract: The focus of this thesis are the methods for computing the null space of matrices. Firstly we introduce the basic matrix factorization methods and how we can calculate the matrix null space from these factorizations. Then we describe the numerical properties of each of these methods for computing the matrix null space. In the final part we will compare these methods with respect to computation time and Frobenius norm for various types of matrices. The measurements and processing of the results into graphs is realised in MATLAB.

Keywords: matrices, null space, matrix factorizations, methods for computing matrix null space

Obsah

Úvod **2**

1 Metody pro výpočet nulového prostoru **4**
1.1 Singulární rozklad 5
1.1.1 Spektrální rozklad 5
1.1.2 Výpočet nulového prostoru matice pomocí SVD 7
1.2 LU rozklad 8
1.2.1 Výpočet nulového prostoru pomocí LU rozkladu 9
1.3 Sloupcová eliminace 9
1.3.1 Výpočet nulového prostoru pomocí sloupcové eliminace . . 10
1.4 Gauss-Jordanova eliminace 11
1.4.1 Výpočet nulového prostoru pomocí Gauss-Jordan- ovy eliminace 11
1.5 QR rozklad 12
1.5.1 Výpočet nulového prostoru pomocí QR rozkladu 13
1.6 LQ rozklad 13
1.6.1 Výpočet nulového prostoru pomocí LQ rozkladu 14

2 Numerické vlastnosti metod **15**
2.1 LU rozklad a Gauss-Jordanova eliminace 15
2.2 SVD 15
2.3 QR rozklad 16

3 Experimenty **17**
3.1 Výsledky 18
3.1.1 Matice typu 100 x n 18
3.1.2 Matice typu 100n x 150n 21

Závěr **27**

Seznam použité literatury **28**

Seznam obrázků **29**

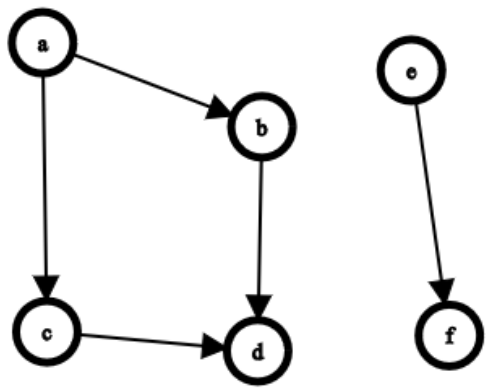
Úvod

Jedním ze základních problémů při studiu matematiky, se kterým se setkáme, je řešení soustav lineárních rovnic. S tímto se spojuje pojem nulového prostoru matice A, což je množina nenulových vektorů x takových, že $Ax = 0$. Tuto množinu značíme $\mathcal{N}(A)$. Klíčovou vlastností nulového prostoru je, že pokud řešíme úlohu $Ax = b$ pro libovolnou pravou stranu b a známe partikulární řešení x, tak množinu všech řešení této soustavy lze vyjádřit množinou

$$\{x + y : y \in \mathcal{N}(A)\}.$$

Díky této vlastnosti se nulový prostor matice využívá v metodách pro řešení mnoha úloh, které mají aplikace v různých oborech.

Například pro úlohy s grafy můžeme znázornit pojem nulového prostoru pomocí matice incidence tak, že dimenze nulového prostoru těchto matic odpovídá počtu komponent grafu a nulové vektory reprezentují jednotlivé komponenty. Uvažujme následující orientovaný graf.



Obrázek 1: Orientovaný graf G s šesti vrcholy

Pro orientovaný graf G z 1 matici incidence A definujeme následovně:

- Sloupce matice reprezentují jednotlivé vrcholy
- Řádky matice reprezentují jednotlivé hrany grafu
- Pro prvky řádku příslušné hrany platí, že prvek je roven -1, pokud v něm hrana začíná, nebo je roven 1, pokud do něj hrana vede. Jinak je prvek roven 0.

Dostáváme následující matici

$$A = \begin{array}{c} \\ ab \\ ac \\ bd \\ cd \\ ef \end{array} \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} \end{array}.$$

Výpočtem nulového prostoru matice A dostáváme

$$\mathcal{N}(A) = \{t_1 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} + t_2 \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} : t_1, t_2 \in \mathbb{R}\},\ dim(\mathcal{N}(A)) = 2.$$

Vidíme tedy, že dimenze nulového prostoru skutečně odpovídá počtu komponent grafu a jednotlivé vektory nulového prostoru nám dávají informaci o tom, které vrcholy do komponenty patří nebo nepatří podle toho, jestli jsou jejich příslušné prvky nenulové. Toto se dá ukázat i formálně. Samozřejmě, pro náš jednoduchý příklad toto vidíme hned, ale v případě, kdybychom měli komplikovanější graf, tak uvedený postup s maticí incidence by mohl být užitečný způsob pro zjištění těchto informací. Poznamenejme ale, že i nulové prostory této jednoduché matice se vyskytují v praktických úlohách.

Znalost nulového prostoru matice má aplikaci v různých metodách pro řešení aplikačních úloh, například

- Metody pro řešení úlohy kvadratického programování v optimalizaci (Gómez, 2005)
- Silová metoda ve statice pro výpočet staticky neurčitých konstrukcí (Heath, Plemmons a Ward, 1984)
- Metoda duální proměnné pro řešení Navier-Stokesových rovnic pro nestlačitelné tekutiny (Hall, Peterson, Porsching a Sledge, 1985)

V této práci nejprve popíšeme různé metody pro rozklad matic a základní myšlenky, na kterých jsou tyto rozklady založeny. Uvidíme, že z těchto rozkladů lze snadno vypočítat nulový prostor původní matice A.

Poté shrneme numerické vlastnostni zavedených metod, na základě kterých se pokusíme určit, jak se jednotlivé metody budou při výpočtu nulového prostoru matice chovat.

V poslední kapitole provedeme numerické experimenty, ve kterých budeme metody porovnávat pro různé typy matic. Hlavním cílem našeho měření bude srovnání výpočetního času a přesnosti metod.

1. Metody pro výpočet nulového prostoru

Máme-li zadanou matici A$\in \mathbb{C}^{m\times n}$ hodnosti m, pak hledáme nějakou matici B $\in \mathbb{C}^{n\times(n-m)}$ řádu $n-m$ takovou, že platí $AB=0$. Tedy matici B takovou, že sloupce matice B tvoří bázi nulového prostoru matice A.

Nulový prostor matice A můžeme vypočítat pomocí známých faktorizačních metod. U každé z těchto metod si ukážeme základní myšlenky, na kterých jsou založeny, a postupy, jak z nich dopočítat nulový prostor matice. V této kapitole si ukážeme různé metody, jak pro matici A vypočítat hledanou matici nulového prostoru B.

Odvození faktorizací vychází z (Tebbens, Hnětynková, Plešinger, Strakoš a Tichý, 2012),(Golub a Van Loan, 1996) a (Watkins, 2002) a metody pro výpočet nulového prostoru vychází z (Heath a kol., 1984).

Základní pojmy

Nejprve si zavedeme základní definice, se kterými budeme pracovat.

Definice 1 (Nulový prostor matice)**.** *Pro matici $A \in \mathbb{C}^{m\times n}$ definujeme* nulový prostor matice *jako množinu*

$$\mathcal{N}(A) = \{x \in \mathbb{C}^n : Ax = 0\}.$$

Definice 2 (Normální matice)**.** *Čtvercová matice A je* normální*, pokud platí*

$$AA^* = A^*A$$

Definice 3 (Hustota matice, řídká matice, hustá matice)**.** *Nechť A je matice typu $m \times n$.*

- *Nechť k je počet nenulových prvků matice A,* hustotou matice *rozumíme číslo*
$$\rho = \frac{k}{m \cdot n}$$
- *A je* řídká*, pokud je většina prvků A nulových*
- *A je* hustá*, pokud není řídká*

Definice 4 (Frobeniova norma matice)**.** *Nechť $A \in \mathbb{C}^{m\times n}$.* Frobeniovou normou *matice A rozumíme číslo*

$$\|A\|_F = (\sum_{i=1}^{m}\sum_{i=1}^{n} a_{ij}^2)^{\frac{1}{2}}$$

Někdy je vhodné nahlížet na matici jako na posloupnost sloupcových vektorů, tedy matici $A = (a_{ij})_{m\times n}$ můžeme zapsat jako $A = (\mathbf{a_1}|\mathbf{a_2}|...|\mathbf{a_n})$, kde $\mathbf{a_i} = (a_{1i}, a_{2i}, \ldots, a_{mi})^T, i \in \{1, \ldots, n\}$ je i-tý m-složkový sloupcový aritmetický vektor matice A.

1.1 Singulární rozklad

Singulární rozklad (zkráceně SVD) je metoda pro rozklad matic, která nám umožňuje jednoduše určit jejich hodnost či normu, ortogonální bázi oboru hodnot a nulového prostoru. Před zavedením singulárního rozkladu je nejprve nutné zavést spektrální rozklad. Odvození spektrálního rozkladu a SVD vychází z (Tebbens a kol., 2012).

1.1.1 Spektrální rozklad

Pro odvození singulárního rozkladu budou důležité spektrální rozklady matic A^*A a AA^*. Matice A^*A a AA^* jsou čtvercové hermitovské pozitivně semidefinitní a můžeme uvažovat jejich spektrální rozklad.

Lemma 1 (Vztah nulových prostorů a oborů hodnot matic 1.)**.** *Nechť $A \in \mathbb{C}^{m\times n}$, potom platí*

$$\mathcal{N}(A) \bigoplus \mathcal{R}(A^*) = \mathbb{C}^n,\ \mathcal{N}(A) \perp \mathcal{R}(A^*),$$

$$\mathcal{N}(A^*) \bigoplus \mathcal{R}(A) = \mathbb{C}^m,\ \mathcal{N}(A^*) \perp \mathcal{R}(A).$$

Důkaz. viz Tebbens a kol. (2012, str. 19).

□

Lemma 2 (Vztah nulových prostorů a oborů hodnot matic 2.)**.** *Nechť $A \in \mathbb{C}^{m\times n}$, pak platí*

$$\mathcal{N}(A^*A) = \mathcal{N}(A), \quad \mathcal{R}(A^*A) = \mathcal{R}(A^*)$$

$$\mathcal{N}(AA^*) = \mathcal{N}(A^*), \quad \mathcal{R}(AA^*) = \mathcal{R}(A)$$

Důkaz. viz Tebbens a kol. (2012, str. 123).

□

Dvě předchozí lemmata nám dávají vztahy pro obory hodnot a nulové prostory matic AA^*,A^*A a matic A, $A*$. Je-li matice A hodnosti r, pak z toho, že platí $dim(\mathcal{R}(A)) = r = dim(\mathcal{R}(A^*))$. Z předchozích lemmat pak dostáváme

$$dim(\mathcal{R}(AA^*)) = dim(\mathcal{R}(A)) = r = dim(\mathcal{R}(A^*)) = dim(\mathcal{R}(A^*A)).$$

Věta 3 (Spektrální rozklad pro normální matice)**.** *Matice A je normální právě tehdy, když existuje unitární matice U a diagonální matice D tak, že*

$$U^*AU = D,\ tj.\ A = UDU^*$$

Důkaz. viz Tebbens a kol. (2012, str. 40).

□

Všimněme si, že matice AA^*, A^*A jsou normální, a tedy pro ně bude platit předchozí věta.

Předpokládejme, že A je hermitovská pozitivně semidefinitní řádu n s hodností r s nezápornými vlastními čísly $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r > 0$, $\lambda_{r+1} = \ldots = \lambda_n = 0$.

Podle věty o spektrálním rozkladu pro normální matice existuje rozklad matice A ve tvaru

$$A = UDU^*,\ U^*U = I,\ D = diag(\lambda_1, \ldots, \lambda_n).$$

Sloupce unitární matice $U = (u_1, \ldots, u_n)$ jsou vlastní vektory matice A a platí

$$Au_j = \lambda_j u_j,\ j = 1, \ldots, n$$

Vlastní vektory u_1,$u_2, \ldots, u_n$ tvoří ortonormální bázi prostoru $\mathbb{C}^n$.

Ze spektrálního rozkladu matice A vidíme bázi nulového prostoru a oboru hodnot matice A,

$$\mathcal{R}(A) = span\{u_1, \ldots, u_r\},\ \mathcal{N}(A) = span\{u_{r+1}, \ldots, u_n\}.$$

Buďte λ_j vlastní čísla a v_j ortonormální vlastní vektory matice A^*A, $j = 1, \ldots, n$. Uvažujme dále bez újmy na obecnosti

$$\lambda_1 \geq \ldots \geq \lambda_r > 0,\ \lambda_{r+1} = \lambda_n = 0.$$

Pak pro $V = (v_1, \ldots, v_n)$ platí

$$V^*A^*AV = diag(\lambda_1, \ldots, \lambda_r, 0, \ldots, 0)$$

Z lemmatu o vztahu nulových prostorů a oborů hodnot matic 2. plyne, že $v_1, \ldots, v_r$ resp. $v_{r+1}, \ldots, v_n$ tvoří ortonormální bázi

$$\mathcal{R}(A^*A) = \mathcal{R}(A^*) = span\{v_1, \ldots, v_r\},$$

resp.

$$\mathcal{N}(A) = \mathcal{N}(A^*A) = span\{v_{r+1}, \ldots, v_n\}.$$

Lemma 4 (Vztah mezi ortonormálními vektory matic A^*A a AA^*.)**.** *Uvažujme spektrální rozklad matice A^*A s vlastními čísly λ_j a v_j, $j = 1, \ldots, r$. Potom jsou vektory*

$$u_j \equiv Av_j/\sqrt{\lambda_j},\ j = 1, \ldots, r$$

ortonormální vlastní vektory AA^* a platí

$$AA^*v_j = \lambda_j u_j,\ \ \|u_j\| = 1,\ j = 1, \ldots, r.$$

Důkaz. viz Tebbens a kol. (2012, str. 124).

□

Z uvedeného lemmatu plyne, že vektory u_j, $j = 1, \ldots, r$ jsou ortonormální vlastní vektory matice AA^*, které odpovídají právě m nenulovým vlastním číslům λ_j a platí

$$Av_j = \sqrt{\lambda_j} u_j,\ j = 1, \ldots, r.$$

Analogie těchto vztahů platí i pro matici AA^*, tj. pokud $u_1, \ldots, u_r$ tvoří ortonormální bázi oboru hodnot a $u_{r+1}, \ldots, u_m$ tvoří ortonormální bázi doplňku oboru hodnot matice AA^*, pak platí

$$span\{u_1, \ldots, u_r\} = \mathcal{R}(AA^*) = \mathcal{R}(A),$$

$$span\{u_{r+1}, \ldots, u_m\} = \mathcal{N}(AA^*) = \mathcal{N}(A^*).$$

Označíme-li $U = (u_1, \ldots, u_m)$, pak platí

$$U^* AA^* U.$$

Tedy jsme získali spektrální rozklad matice AA^* pomocí spektrálního rozkladu A^*A. Z konstrukce plyne, že nenulová vlastní čísla matic A^*A a AA^* se rovnají včetně násobnosti.

Z lemmatu o vztahu nulových prostorů a oborů hodnot matic 2. máme

$$\mathcal{R}(A^*) = span\{v_1, \ldots, v_r\} \ ,\mathcal{N}(A) = span\{v_{r+1} \ldots, v_n\}$$

$$\mathcal{R}(A) = span\{u_1, \ldots, u_r\} \ ,\mathcal{N}(A) = span\{u_{r+1} \ldots, u_m\}$$

Nyní můžeme popsat singulární rozklad.

Definice 5 (Singulární čísla)**.** *Nechť $A \in \mathbb{C}^{m\times n}$. Odmocniny vlastních čísel matice A^*A nazveme singulárními čísly matice A,*

$$\sigma_j = \sqrt{\lambda_j}, \ \ j = 1, \ldots, r, \ \ r = rank(A).$$

$$\sigma_1 \geq \ldots \geq \sigma_r > 0.$$

Tedy podle předchozích poznatků platí $Av_j = \sigma_j u_j$, $j = 1, \ldots, r$ a můžeme psát

$$AV = U\Sigma,$$

kde

$$\Sigma = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{C}^{m\times n}, \ \Sigma_r = diag(\sigma_1, \ldots, \sigma_r) \in \mathbb{C}^{r\times r}, \ ,$$

$$V = (v_1, \ldots, v_n).U = (u_1, \ldots, u_m).$$

Ekvivalentně

$$A = U\Sigma V^*, \ \ A^* = V\Sigma^T U^*.$$

Věta 5 (Singulární rozklad)**.** *Nechť $A \in \mathbb{C}^{m\times n}$ je matice hodnosti r, pak A lze vyjádřit jako*

$$A = U\Sigma V^*,$$

kde $U \in \mathbb{C}^{m\times m}, V \in \mathbb{C}^{n\times n}$ jsou unitární a $\Sigma \in \mathbb{C}^{m\times n}, \Sigma = diag(\sigma_1, \ldots, \sigma_r, 0, \ldots, 0)$.

Vidíme tedy, že pokud bychom byli schopni spočítat singulární rozklad matice A, tak nulový prostor matice A je roven množině $span\{v_{r+1}, \ldots, v_n.\}$

1.1.2 Výpočet nulového prostoru matice pomocí SVD

Pomocí SVD můžeme vyřešit úlohu hledání matice B splňující $AB = 0$ následovně

- Transformujeme matici A na bidiagonální tvar pomocí unitárních transformací nebo Golub-Kahanovy iterační bidiagonalizace. Transformovaná matice A je ve tvaru

  $$C = PAQ,$$

  kde P,Q jsou unitární a C je bidiagonální.

- Nalezneme singulární rozklad bidiagonální matice C pomocí iteračního procesu, který nuluje její naddiagonální prvky, například pomocí implicitního QR algoritmu, potom dostáváme rovnost

$$\Sigma = U^* A V$$

- Nulový prostor matice A odpovídá sloupcovým vektorům $v_{m+1}, \ldots, v_n$ matice V, tedy

$$B = [v_{m+1}, \ldots, v_n].$$

Výše uvedený postup je zapsán bez násobení matice A permutačními maticemi P_1 zleva a P_2 zprava. V případě, kdy uvažujeme $P_1 A P_2$, tak můžeme opět aplikovat algoritmus a dostáváme rozklad

$$\Sigma = U^* P_1 A P_2 V,$$

a potom platí, že hledaná matice nulového prostoru je

$$B = P_2 V_2.$$

Obecně pokud známe singulární rozklad matice A, tak matici B získáme z následujících vztahů

$$P_1 A P_2 = U \Sigma V^*,$$
$$[V_1, V_2] = V,$$
$$B = P_2 V_2.$$

1.2 LU rozklad

Jednou ze základních metod pro řešení soustav lineárních rovnic je LU rozklad, který nám původní matici A rozloží na součin dolní trojuhelníkové matice a horní trapezoidální matice. LU rozklad matice A hodnosti m je založen na Gaussově eliminaci následovně

$$A \longrightarrow A^{(1)} \equiv \begin{bmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,n} \\ 0 & a^1_{2,2} & \ldots & a^1_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a^1_{m,2} & \ldots & a^1_{m,n} \end{bmatrix} = M^{-1} A,$$

kde

$$M_1^{-1} = I - m_1 e_1^T = \begin{bmatrix} 1 & & & & \\ -\frac{a_{2,1}}{a_{1,1}} & 1 & & & \\ -\frac{a_{3,1}}{a_{1,1}} & & 1 & & \\ \vdots & & & \ddots & \\ -\frac{a_{m,1}}{a_{1,1}} & & & & 1 \end{bmatrix}, \quad m_1 = \begin{bmatrix} \frac{a_{2,1}}{a_{1,1}} \\ \frac{a_{3,1}}{a_{1,1}} \\ \vdots \frac{a_{m,1}}{a_{1,1}} \end{bmatrix}.$$

Jelikož A má hodnost m, tak permutováním zaručíme, že nikdy nedělíme nulou, tedy LU rozklad lze provést. V i-tém kroku bude

$$A^{(i)} \equiv \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & & & a_{1,n} \\ 0 & a_{2,2}^{(1)} & \dots & & & a_{2,n}^{(1)} \\ \vdots & 0 & \ddots & & & \\ & \vdots & & a_{i,i}^{(i-1)} & \dots & a_{i,n}^{(i-1)} \\ 0 & 0 & & a_{m,i}^{(i-1)} & \dots & a_{m,n}^{(i-1)} \end{bmatrix}, \; m_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \frac{a_{i+1,i}}{a_{i,i}} \\ \frac{a_{i+2,i}}{a_{i,i}} \\ \vdots \\ \frac{a_{m,i}}{a_{i,i}} \end{bmatrix}.$$

Po m-1 krocích dostaneme

$$U \equiv A^{(m-1)} = M_{m-1}^{-1} \dots M_2^{-1} M_1^{-1} A, \;\; L \equiv M_1 M_2 \dots M_{m-1},$$

kde $U \in \mathbb{C}^{m \times n}$ je horní trapezoidální a L je dolní trojuhelníková řádu m s jednotkami na diagonále a upravením rovnosti dostáváme

$$A = LU = L[U_1 \; U_2], \;\; U_1 = [u_1, \dots, u_m], \;\; U_2 = [u_{m+1}, \dots, u_n],$$

kde U_1 je horní trojuhelníková.

Nyní můžeme LU rozklad použít k řešení naší úlohy $AB = 0$.

1.2.1 Výpočet nulového prostoru pomocí LU rozkladu

Mějme matici $A \in \mathbb{C}^{m \times n}$ hodnosti m.

- Provedeme potřebné permutace matice A na matici $P_1 A P_2$ tak, abychom měli rozklad matice v hledaném tvaru.
- Vypočítáme LU rozklad, dostáváme $P_1 A P_2 = LU = L[U_1 \; U_2]$.
- Položíme $J = U_1^{-1} U_2$, pak hledané řešení úlohy je

$$B = P_2 \begin{bmatrix} -J \\ I_{n-m} \end{bmatrix}$$

1.3 Sloupcová eliminace

Metoda sloupcové eliminace je založena na stejné myšlence jako LU rozklad, nicméně místo poddiagonálních prvků budeme nulovat naddiagonální prvky. Neuvažujeme-li permutace, pak

$$M_1^{-1} = \begin{bmatrix} 1 & -\frac{a_{1,2}}{a_{1,1}} & -\frac{a_{1,3}}{a_{1,1}} & \dots & -\frac{a_{1,n}}{a_{1,1}} \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & & \ddots & & \\ 0 & & & & 1 \end{bmatrix},$$

pak

$$A^{(1)} = AM_1^{-1} \begin{bmatrix} a_{1,1} & 0 & 0 & \dots & 0 \\ a_{2,1} & a_{2,2}^{(1)} & a_{2,3}^{(1)} & \dots & a_{2,n}^{(1)} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{m,1} & a_{m,2}^{(1)} & \dots & & a_{m,n}^{(1)} \end{bmatrix}.$$

V obecném i-tém kroku ($i < m$) budeme mít

$$M_i^{-1} = \begin{bmatrix} 1 & 0 & \dots & & & & 0 \\ 0 & 1 & & & & & 0 \\ \vdots & 0 & \ddots & & & & \\ & & & 1 & -\frac{a_{i,i+1}}{a_{i,i}} & \dots & -\frac{a_{m,n}}{a_{i,i}} \\ & & & & \ddots & & 0 \\ & & & & & & \vdots \\ 0 & \dots & & & & 0 & 1 \end{bmatrix}$$

a

$$A^{(i)} = AM_1^{-1}M_2^{-1} \dots M_i^{-1} =$$

$$= \begin{bmatrix} a_{1,1} & 0 & \dots & & & & & 0 \\ a_{2,1} & a_{2,2}^{(1)} & 0 & \dots & & & & \vdots \\ \vdots & a_{3,2}^{(1)} & a_{3,3}^{(2)} & \ddots & & & & \\ & & \ddots & \ddots & & & & \\ & & & a_{i,i-1}^{(i-1)} & a_{i,i}^{(i)} & 0 & \dots & \\ & & & & \ddots & \ddots & & \\ a_{m,1} & a_{m,2}^{(1)} & \dots & & & a_{m,m}^{(i)} & \dots & a_{m,n}^{(i)} \end{bmatrix}.$$

Po (m-1)-tém kroku, kde při použití sloupcových permutací jsou tyto kroky dobře definovány, dostáváme rozklad

$$P_1AP_2 = [L\ 0]U,\ U^{-1} = [N_1\ N_2],$$

kde $N_1 \in \mathbb{C}^{n\times m}$ a $N_2 \in \mathbb{C}^{n\times(n-m)}$, $L \in \mathbb{C}^{m\times m}$ je dolní trojuhelníková a U je horní trojuhelníková řádu n.

1.3.1 Výpočet nulového prostoru pomocí sloupcové eliminace

Shrneme si tedy naznačený postup:

- Permutováním převedeme A na P_1AP_2.
- Podobně jako u LU rozkladu aplikujeme Gaussovu eliminaci na sloupce matice A a získáme rozklad $P_1AP_2 = [L\ 0]U$.
- Položíme $U^{-1} = [N_1\ N_2]$, potom hledaná matice nulového prostoru je

$$B = P_2N_2.$$

Sloupcovou eliminaci můžeme chápat jako transpozici LU rozkladu matice A^T.

1.4 Gauss-Jordanova eliminace

Tato metoda odpovídá Gaussově eliminaci se zpětným chodem, tedy nejprve eliminujeme poddiagonální prvky, poté naddiagonální prvky a nakonec přenásobíme diagonální prvky tak, abychom měli jednotkovou diagonálu. Předpokládejme, že už jsme provedli Gaussovu eliminaci a máme matici ve tvaru LU rozkladu, tj. $P_1AP_2 = LU$. Potom ve zpětném chodu je postup v i-tém kroku následující

$$M_i^{-1} = \begin{bmatrix} 1 & 0 & \dots & 0 & -a_{1,i} & 0 & \dots & 0 \\ 0 & 1 & \ddots & & \vdots & & & \vdots \\ \vdots & 0 & \ddots & & -a_{i-1,i} & & & \\ & & \vdots & & 1 & 0 & & \\ & & & & & \ddots & \ddots & \\ & & & & & & 1 & 0 \\ 0 & \dots & & & & & 0 & 1 \end{bmatrix}, \; A^{(i)} = \frac{1}{a_{i,i}^{(i-1)}} M_i^{(-1)} A^{(i-1)}$$

Po (m-1). kroku dostáváme

$$A^{(m-1)} = M_{m-1}^{-1} \dots M_2^{-1} M_1^{-1} L[U_1 \; U_2] =$$

$$= \begin{bmatrix} 1 & 0 & \dots & 0 & a_{1,m+1}^{(m-1)} & \dots & a_{1,n}^{(m-1)} \\ 0 & 1 & & \vdots & a_{2,m+1}^{(m-1)} & \dots & a_{2,n}^{(m-1)} \\ \vdots & 0 & \ddots & & \vdots & & \vdots \\ 0 & \dots & 0 & 1 & a_{m+1,m+1}^{(m-1)} & \vdots & a_{m,n}^{(m-1)} \end{bmatrix} = [I_m \; J]$$

kde

$$J = \begin{bmatrix} a_{1,m+1}^{(m-1)} & a_{1,m+2}^{(m-1)} & \dots & a_{1,n}^{(m-1)} \\ a_{2,m+1}^{(m-1)} & a_{2,m+2}^{(m-1)} & \dots & a_{2,n}^{(m-1)} \\ \vdots & & \ddots & \\ a_{m+1,m+1}^{(m-1)} & \dots & & a_{m,n}^{(m-1)} \end{bmatrix}$$

Získali jsme tedy následující rozklad

$$P_1AP_2 = M[I_m \; J],$$

kde $M \in \mathbb{C}^{m \times m}$ a $J \in \mathbb{C}^{m \times (n-m)}$.

1.4.1 Výpočet nulového prostoru pomocí Gauss-Jordanovy eliminace

Postup pro výpočet nulového prostoru pomocí Gauss-Jordanovy eliminace je tedy následující

- Provedeme potřebné permutace a převedeme matici A na matici P_1AP_2.
- Vypočítáme LU rozklad matice $P_1AP_2 = LU$.
- Postupným přenásobováním a sloupcovou eliminací získáme požadovaný rozklad $P_1AP_2 = M[I_m \; J]$

- Hledaný nulový prostor matice A je roven

$$B = P_2 \begin{bmatrix} -J \\ I_{n-m} \end{bmatrix}.$$

1.5 QR rozklad

Tento rozklad nám rozloží matici A na součin v obecném případě unitární, jinak ortogonální matice $Q \in \mathbb{C}^{m\times m}$ a horní trapezoidální matice $R \in \mathbb{C}^{m\times n}$, tedy $A = QR$. Máme více algoritmů pro výpočet QR rozkladu matice A, hlavní metody jsou založené na Givensových rotacích, Householderových reflexích a varianty Gram-Schmidtova ortogonalizačního procesu. Klasická Gram-Schmidtova ortogonalizace (CGS) nám z matice A iteračně vygeneruje ortonormální bázi $Q = (q_1, \ldots, q_m)$ a matici koeficientů R.

Věta 6 (QR rozklad)**.** *Nechť $A \in \mathbb{C}^{m\times n}$, pak existuje ortogonální matice $Q \in \mathbb{C}^{m\times m}$ a horní trapezoidální matice $R \in \mathbb{C}^{m\times n}$ tak, že*

$$A = QR.$$

Důkaz. viz Golub a Van Loan (1996, str. 246).

□

Ukážeme, jak spočítat QR rozklad pomocí CGS. Označme $A = [a_1, a_2, \ldots, a_n]$, kde $a_1, \ldots, a_n$ jsou sloupcové vektory matice A, budeme navíc v každém kroku iterace chtít, aby platilo $span\{a_1, \ldots, a_k\} = span\{q_1, \ldots, q_k\}$. Postup je pak následující

- V prvním kroku normováním dostaneme $r_{1,1} = \|a_1\|$ a $q_1 = \frac{a_1}{r_{1,1}}$.
- V druhém kroku musí platit $span\{a_1, a_2\} = span\{q_1, q_2\}$ a $q_1 \perp q_2$, což splníme tím, že odečteme projekci vektoru a_2 do podprostoru $span\{q_1\}$ a normalizujeme. Tedy

$$z = (I - q_1 q_1^*)a_2 = a_2 - (q_1^* a_2)q_1, \quad r_{1,2} = q_1^* a_2, \quad r_{2,2} = \|z\|,$$

$$q_2 = \frac{z}{r_{2,2}}.$$

- V k-tém kroku stejným postupem odečteme od od vektoru a_k jeho projekci do prostoru $span(q_1, \ldots, q_{k-1})$ a normalizujeme,

$$z = (I - Q_{k-1}Q_{k-1}^*)a_k = a_k - \sum_{i=1}^{k-1}(q_i^* a_k)q_i, \quad r_{i,k} = q_i^* a_k, \; k = 1, \ldots, k-1$$

$$r_{k,k} = \|z\|,$$

$$q_k = \frac{z}{r_{k,k}}.$$

Ve výsledku dostáváme unitární matici $Q = [q_1, \dots, q_m]$ a horní trapeziodální matici $R \in \mathbb{C}^{m \times n}$

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,m} & r_{1,m+1} & \dots & r_{1,n} \\ 0 & r_{2,2} & & r_{2,n} & r_{2,m+1} & & r_{2,n} \\ \vdots & \ddots & \ddots & & & \vdots & \\ 0 & \vdots & 0 & r_{m,m} & r_{m,m+1} & & r_{m,n} \end{bmatrix}$$

a bude platit rovnost

$$A = QR.$$

Ve výpočtu výše jsme předpokládali, že prvních m sloupcových vektorů matice A jsou lineárně nezávislé, což lze, neboť matice A je hodnosti m.

1.5.1 Výpočet nulového prostoru pomocí QR rozkladu

Shrnutím předcházejícího dostaneme následující postup:

- Permutováním převedeme A na P_1AP_2.
- Pomocí vhodné metody vypočítáme QR rozklad $P_1AP_2 = QR = Q[R_1\ R_2]$.
- Položíme $J = R_1^{-1}R_2$. Potom hledaná matice nulového prostoru je

$$B = P_2 \begin{bmatrix} -J \\ I_{n-m} \end{bmatrix}.$$

kde pomocí $R = [r_1, \dots, r_m, r_{m+1}, \dots, r_n]$ značíme $R_1 = [r_1, \dots, r_m]$, $R_2 = [r_{m+1}, \dots, r_n]$.

1.6 LQ rozklad

Tento rozklad rozloží matici A na součin

$$A = [L\ 0]Q = [L\ 0] \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix},$$

kde L je dolní trojuhelníková typu $m \times m$ a Q je unitární typu $n \times n$, Q_1 je typu $m \times n$ a Q_2 je typu $(n-m) \times n$. LQ rozklad počítáme podobně jako QR rozklad za pomocí unitárních transformací aplikovaných zprava, jako příklad si uvedeme analogii Gram-Schmidtova algoritmu pro LQ rozklad.

Označme $A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}$. Potom v prvním kroku máme

$$l_{11} = \|a_1\|,\ q_1 = \frac{a_1}{l_{11}}.$$

Dále uvažujme obecný k-tý krok

$$l_{ik} = q_i^* a_k,\ j = 1, \dots, k-1,$$

$$w_k = \sum_{i=1}^{k-1} l_{ik} q_i,$$

$$l_{kk} = \|a_k - w_k\|,$$

$$q_k = l_{kk}(a_k - w_k).$$

Na konci iteračního procesu dostáváme

$$A = LQ = \begin{bmatrix} l_{11} & 0 & 0 & \dots & 0 & \dots & 0 \\ l_{21} & l_{22} & & & \vdots & & \\ \vdots & & \ddots & & & & \\ l_{m1} & \dots & & l_{mm} & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix}$$

1.6.1 Výpočet nulového prostoru pomocí LQ rozkladu

Nulový prostor matice můžeme hledat pomocí LQ rozkladu následovně:

- Provedeme potřebné permutace a převedeme matici A na matici $P_1 A P_2$.
- Vypočítáme pomocí vhodně zvolené metody rozklad $P_1 A P_2 = [L\ 0]Q = [L\ 0] \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$
- Hledaná matice nulového prostoru je

$$B = P_2 Q_2.$$

Vidíme, že na LQ rozklad lze nahlížet jako na transpozici QR rozkladu matice A^T

2. Numerické vlastnosti metod

V této části si shrneme numerické vlastnosti uvedených metod a případné komplikace, které by při jejich použití mohly nastat. Při popisu numerických vlastností metod vycházíme z (Tebbens a kol., 2012),(Golub a Van Loan, 1996) a (Watkins, 2002).

V praktických úlohách se některé z metod LQ a QR, resp. LU a sloupcové eliminace, upřednostňují a rozlišují, ovšem v našem čistě matematickém kontextu si jsou metody velmi podobné a můžeme jednu implementovat za pomocí druhé, tedy dále budeme z těchto čtyř zmíněných metod uvažovat jen QR a LU rozklad.

2.1 LU rozklad a Gauss-Jordanova eliminace

Pokud budeme uvažovat řídkou matici A, tak v procesu eliminace u obou faktorizačních metod bude docházet k zaplnění. Pro LU rozklad ale existují efektivní metody pro zachování řídkosti a Gauss-Jordanova metoda bude pamětově náročnější. Pro rozklad pomocí Gauss-Jordanovy eliminace $P_1AP_2 = M[I_m\ J]$ je matice J obvykle hustá a může zabírat velké množství paměti pro velké $n - m$. Jelikož na konci této metody chceme dostat blok jednotkové matice na hlavní diagonále, tak tato metoda bude potřebovat velký počet eliminací a tedy bude výpočetně náročná. Obě metody jsou pouze podmíněně zpětně stabilní.

Proto může dojít k numerickým nestabilitám kvůli růstovému faktoru. Například pro Gaussovu eliminaci s řádkovou permutací pro matici řádu n v následujícím případě dostáváme

$$\begin{bmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ \vdots & \ddots & \ddots & & \vdots \\ -1 & \dots & -1 & 1 & 1 \\ -1 & \dots & -1 & -1 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & & & & 1 \\ 0 & 1 & & & 2 \\ \vdots & \ddots & \ddots & & \vdots \\ 0 & \dots & 0 & 1 & 2^{n-2} \\ 0 & \dots & 0 & 0 & 2^{n-1} \end{bmatrix}$$

a tedy je růst exponenciální.

Pro $m = n$ je přibližný počet operací pro výpočet LU rozkladu $\frac{2}{3}n^3$ a pro Gauss-Jordanovu eliminaci $\frac{4}{3}n^3$.

Navíc pro výpočet nulového prostoru je v případě LU rozkladu nutné navíc spočítat inverzní matici, která může být špatně podmíněna, což zvyšuje výpočetní dobu a nestabilitu výpočtu.

2.2 SVD

SVD faktorizaci můžeme spočítat pomocí spektrálních rozkladů matic AA^* a A^*A, ovšem podmíněnost obou matic je rovna $\kappa(A)^2$, tedy pokud by byla matice A špatně podmíněna, tak by podle **Věty 5** mohl být výpočet singulárních čísel nepřesný. Abychom se vyhnuli problémům se špatnou podmíněností matice, tak matici nejprve unitárními transformacemi převedeme na bidiagonální tvar a až poté vypočítáme singulární rozklad matice.

SVD je numericky stabilní metoda, ale jelikož potřebujeme uchovávat matice U, Σ, V, tak pro velké matice bude metoda SVD výpočetně i pamětově náročná. Dále při aplikaci unitárních transformací zleva i zprava bude docházet v případě řídkých matic k zaplnění, tedy opět bude tato metoda pamětově náročná. Složitost výpočtu faktorizace SVD závisí na iteračním procesu, kterým počítáme singulární rozklad bidiagonální matice. Kdybychom předpokládali, že A je čtvercová matice řádu m, iterační proces rychle konverguje a nezvyšuje počet operací, pak dolní odhad na počet operací je $\frac{16}{3}m^3$.

Na rozdíl od metod LU a QR, z SVD hned vidíme vektory nulového prostoru a není třeba provádět dodatečné výpočty.

2.3 QR rozklad

QR rozklad je typicky implementován pomocí Givensových rotací, Householderových reflexí nebo různými modifikacemi Gram-Schmidta algoritmu, u každé z těchto variant z podmínky ortogonality bude pro velké řídké matice docházet k zaplnění a k větší výpočetní a pamětové náročnosti oproti jednodušším metodám.

V porovnání s LU rozkladem bude QR rozklad numericky stabilnější než LU rozklad, ovšem kvůli procesu ortogonalizace bude QR rozklad výpočetně náročnější než LU rozklad. Tedy bychom čekali, že QR rozklad nám bude dávat přesnější výsledky, ale bude pomalejší. Pro $m = n$ je přibližný počet operací pro Householderův QR rozklad $\frac{4}{3}n^3$ a Givensův QR rozklad $2n^3$, u ICGS obecně nelze počet operací odhadnout.

Podobně jako u LU rozkladu, i zde v posledním kroku počítáme inverzní matici, která může být špatně podmíněna, toto se bude projevovat větší výpočetní dobou a většími nestabilitami.

3. Experimenty

Měření budeme provádět v prostředí MATLABu verze R2023b, který je přímo určený na numerické výpočty s maticemi a ve kterém jsou implementované maticové operace a algoritmy pro faktorizace, které budeme používat. Měření rozdělíme na dvě části a v obou částech budeme měřit vše zvlášť pro náhodně vygenerované husté a řídké reálné matice s plnou řádkovou hodností, kde řídké matice budeme počítat s přibližnou hustotou $\rho = 0.1$. Všechna měření provedeme několikrát a výsledná data zprůměrujeme.

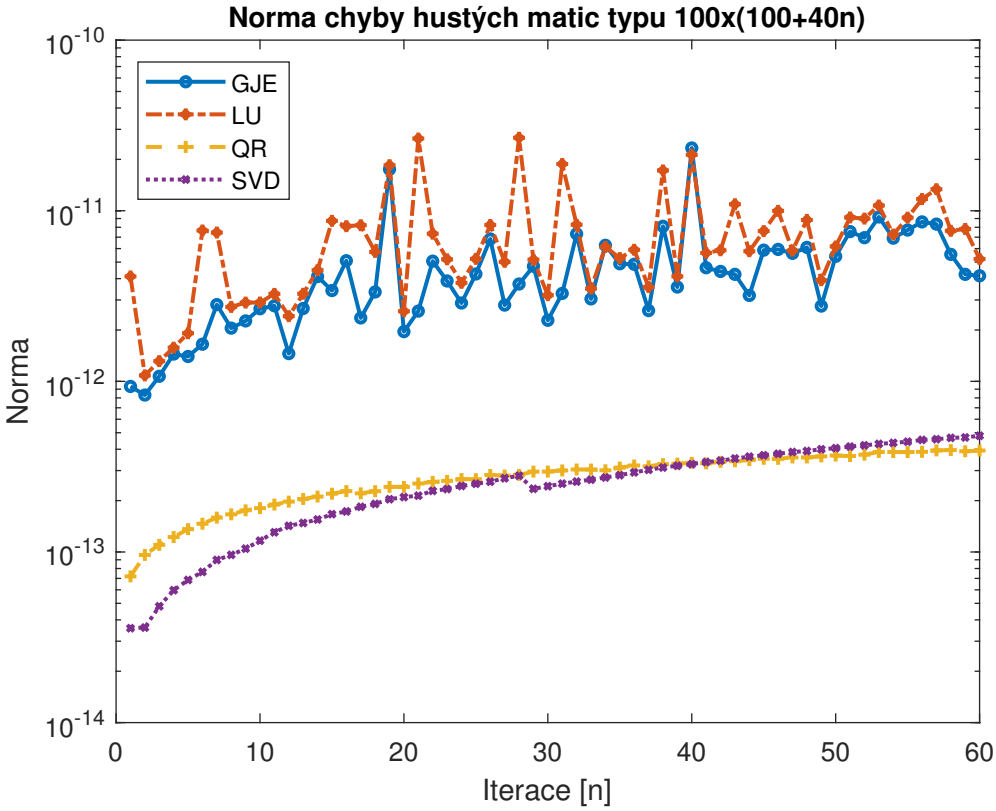
V první části budeme mít matice typu $100 \times (100 + 40n)$, tedy pevný počet řádků a budeme přidávat sloupce, a v druhé části budeme mít matice typu $100n \times 150n$, tedy budeme mít řádky a sloupce v poměru $2 : 3$, pro $n = 1,2, \ldots, k$, kde k bude dostatečně velké tak, abychom mohli pozorovat rozdíly mezi jednotlivými metodami. Tedy budeme v první části zkoumat chování metod pro relativně malé úlohy a v druhé části pro velké úlohy. Cílem našeho měření bude zkoumání chyby každé metody $\|AB\|_F \approx 0$ a čas výpočtu v závislosti na velikosti úlohy.

3.1 Výsledky

3.1.1 Matice typu 100 x n

Husté matice

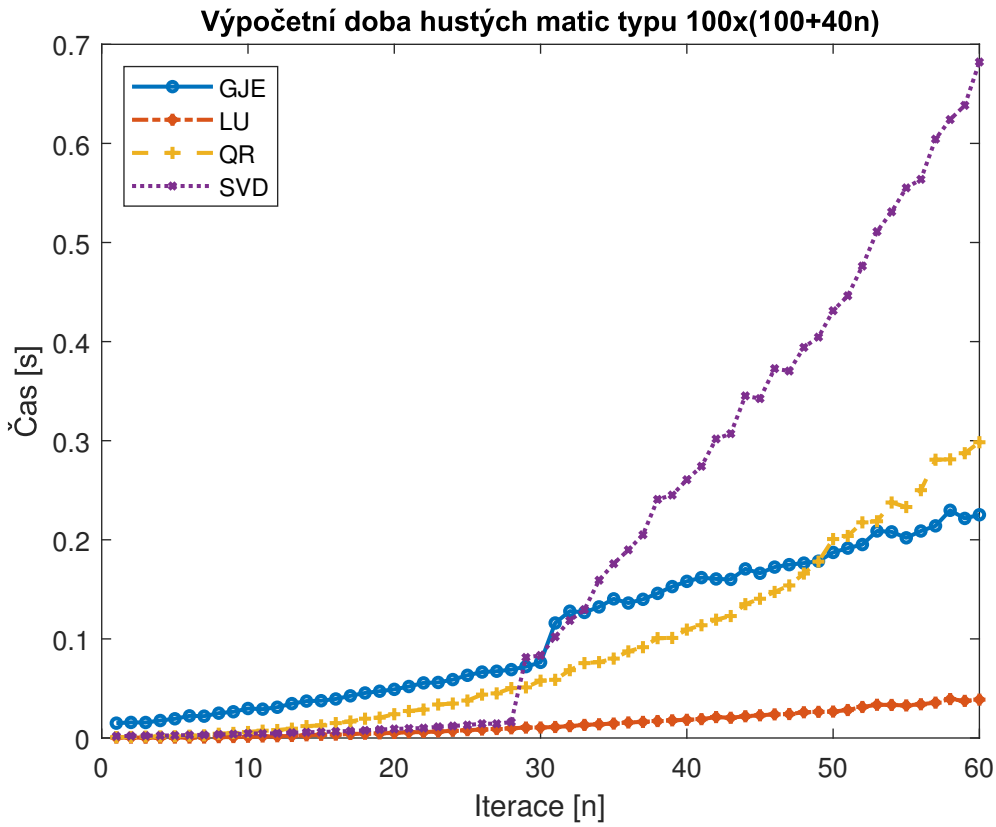
Z hlediska výpočetní přesnosti pro husté matice z grafu 3.1 jsou metody SVD a QR srovnatelné s nejmenší normou chyby $< 4.5 \cdot 10^{-13}$. Zvětšujeme-li počet sloupců husté matice tak vidíme, že metoda SVD ztrácí přesnost rychleji. Metody GJE a LU 3.1 mají přibližně stejnou normu chyby $< 7 \cdot 10^{-11}$, ovšem v některých bodech grafu je vidět, že obě metody mají pro nějaké matice několikanásobně vyšší normu. Toto je nejspíš způsobené špatnou podmíněností maticového bloku prvních 100 sloupců, který invertujeme při výpočtu matice B.



Obrázek 3.1: Norma chyby pro husté matice typu $100 \times (100 + 40n)$

Výpočetní čas metod pro husté matice z grafu 3.2 je nejmenší v případě LU rozkladu s hodnotami $< 0.05s$ s pomalým růstem pro rostoucí počet sloupců. QR rozklad je z počátku rychlejší než GJE , obě metody s výpočetním časem < 0.3, ovšem s rostoucím počtem sloupců má ke konci GJE menší výpočetní dobu. Toto je způsobené tím, že pro metody LU i GJE eliminujeme pro každou matici stejný počet řádků, zatímco z podmínky ortogonality je výpočet QR rozkladu v každém kroku složitejší, obvzláště pro řídké matice. Pro menší počet sloupců má i malou výpočetní dobu SVD, ovšem při dosažení poměru řádků ku sloupcům přibližně $1 : 10$ výpočetní doba SVD výrazně roste s konečnými hodnotami $< 0.7s$. Jelikož jsou tyto matice moc malé na to, aby způsobovaly problém s nedostatkem paměti, tak je toto chování nejspíš způsobeno tím, že v implementaci SVD se od určitého

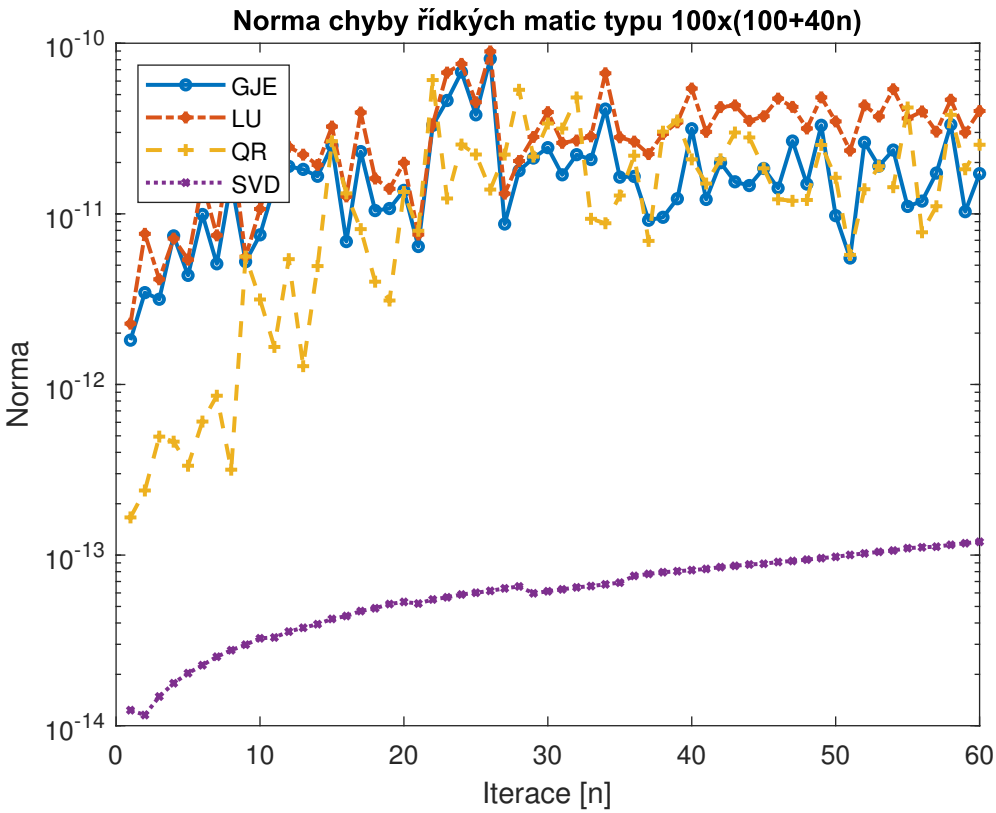
rozměru matice používá jiný algoritmus pro výpočet SVD. Jak uvidíme, tak stejný jev nastane i v případě řídkých matic.



Obrázek 3.2: Čas výpočtu pro husté matice typu $100 \times (100 + 40n)$

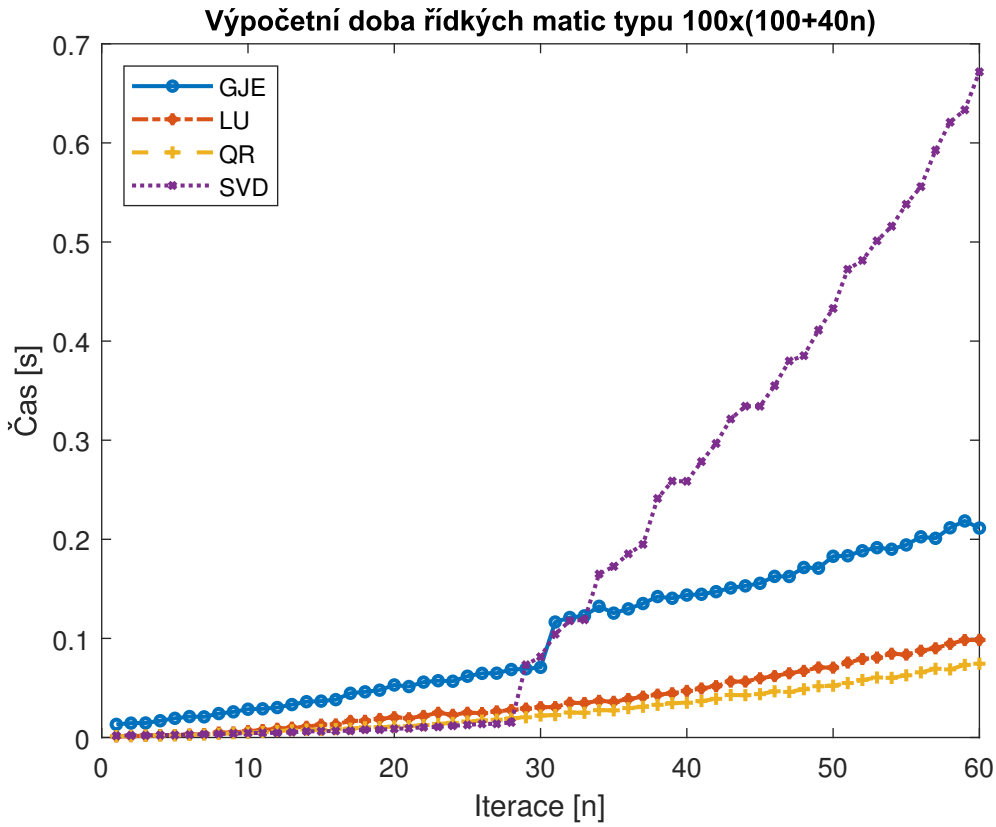
Řídké matice

Pro řídké matice mají metody GJE a LU podle grafu 3.3 stejně jako v případě hustých matic normu chyby s hodnotami $< 9 \cdot 10^{-11}$ s tím, že GJE je přesnější. Na rozdíl od hustých matic jsou zde obě metody méně stabilní s častějším kolísáním chyby ve všech krocích. K podobným nestabilitám ze stejných důvodů špatné podmíněnosti invertované matice dochází i u QR z grafu 3.3, který má normu chyby s hodnotami $< 9 \cdot 10^{-11}$. Metoda SVD je nejpřesnější se stabilními hodnotami normy $< 10^{-13}$ a pomalu rostoucí chybou.



Obrázek 3.3: Norma chyby pro řídké matice typu $100 \times (100 + 40n)$

Na rozdíl od hustých matic má výrazně menší výpočetní dobu metoda QR podle grafu 3.4 s výpočetním časem $< 0.05s$. Metody SVD a GJE se nijak od hustých matic neliší a metoda LU je nepatrně pomalejší oproti hustým maticím. Kratší výpočetní doba QR bude nejspíš způsobena jinou implementací pro řídké matice, která způsobuje menší výpočetní dobu ale i méně přesné výsledky. Větší výpočetní doba LU rozkladu je způsobena tím, že narozdíl od hustých matic je potřeba intenzivněji permutovat matici tak, abychom zaručili regularitu bloku matice U, kterou budeme invertovat.



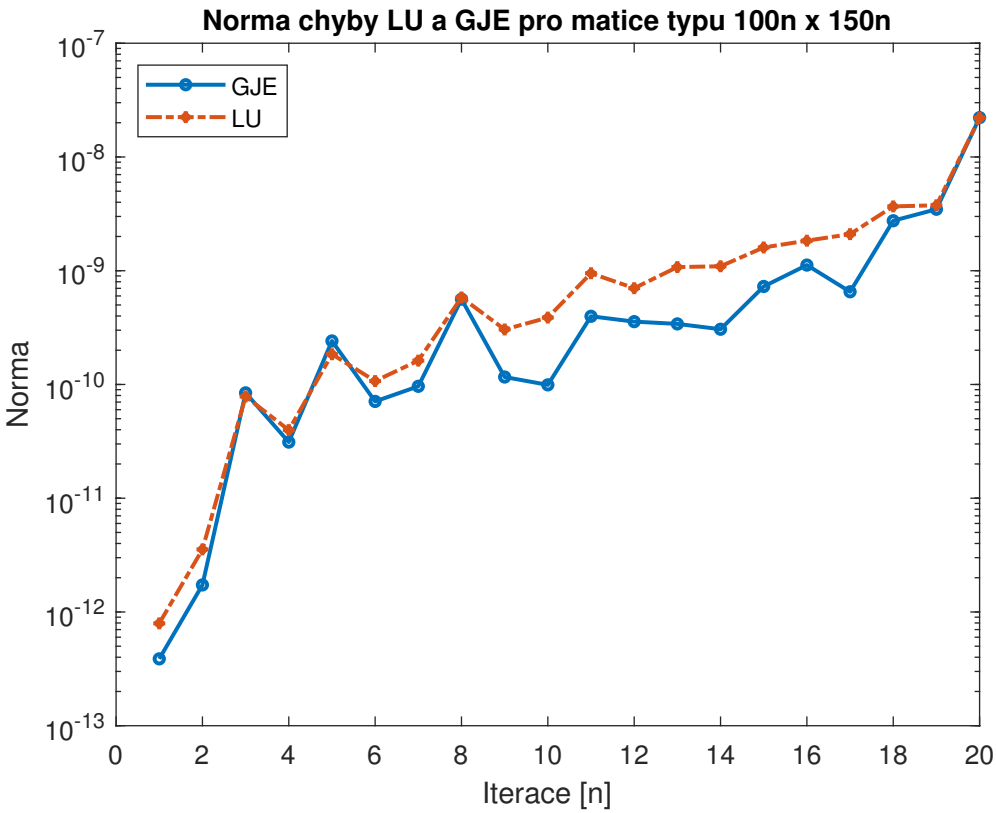
Obrázek 3.4: Čas výpočtu pro řídké matice typu $100 \times (100 + 40n)$

Zmíněné nestability metod LU a GJE jsou způsobeny špatnou podmíněností matic, které při výpočtu matice B invertujeme. Podíváme-li se na grafy normy chyby metod LU a GJE pro husté a řídké matice, tak vidíme, že pro řídké matice jsou je kolísání chyby u obou metod častější. Tedy pro řídké matice jsou obě metody méně stabilní. Z vypozorovaných výsledků bychom čekali, že v druhé části měření pro velké matice budou obě metody GJE a SVD nepraktické kvůli výpočetní náročnosti. Vidíme, že pro matice s rostoucím počtem sloupců bude nejvhodnější volba metody LU v případě hustých matic a QR v případě řídkých matic.

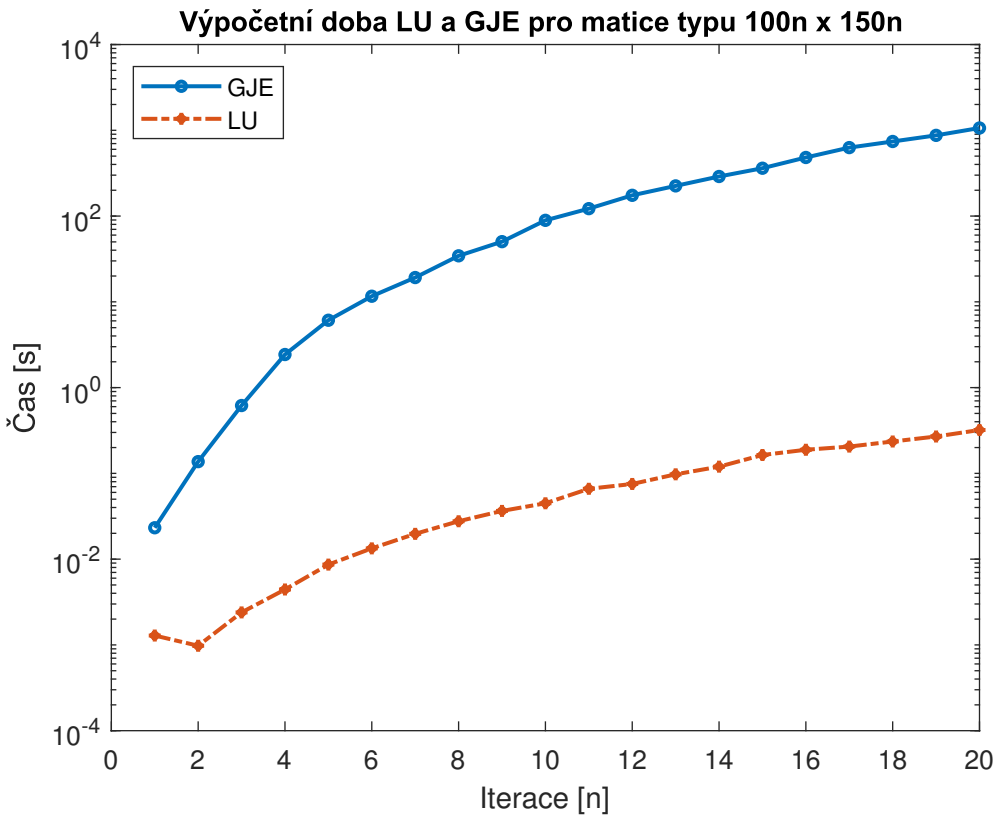
3.1.2 Matice typu 100n x 150n

Husté matice

Oproti prvnímu měření z grafů 3.6 a 3.8 už můžeme pozorovat velké rozdíly ve výpočetním času. V případě hustých matic vidíme, že výpočetní doba metody GJE je mnohonásobně větší oproti LU s výpočetní dobou $< 10^3 s$ pro $n = 1, \ldots, 20$ a se srovnatelnou normou chyby. Z tohoto důvodu vyřadíme GJE z dalších měření a budeme zkoumat pouze metody LU, QR a SVD.



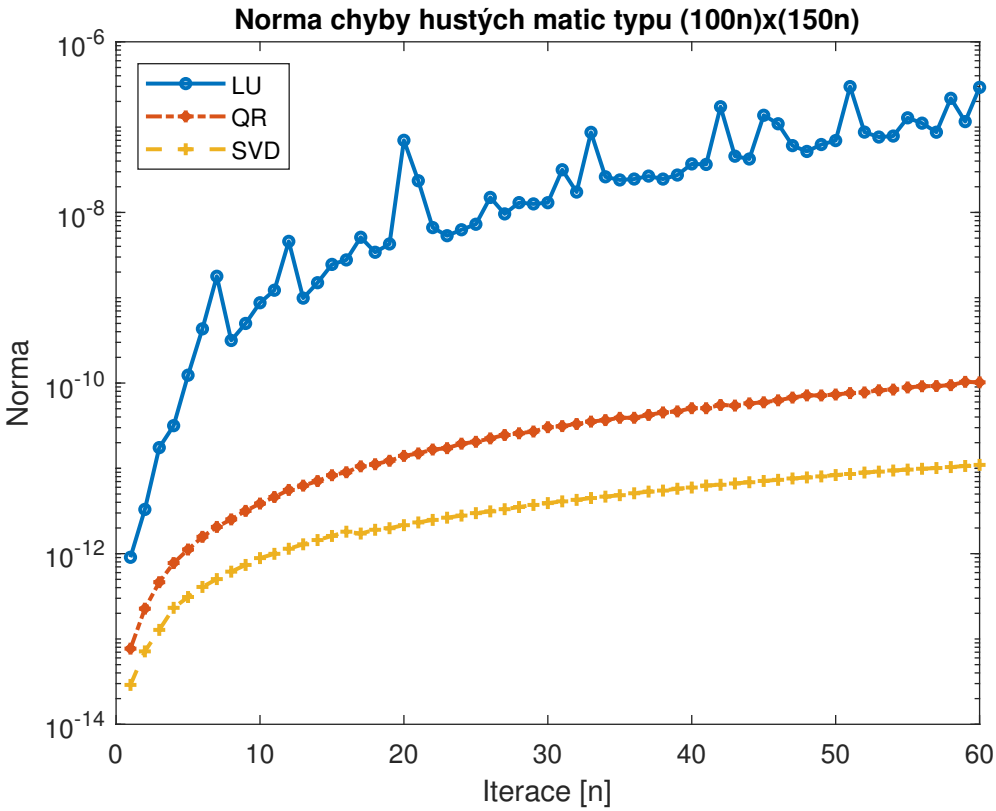
Obrázek 3.5: Norma chyby GJE a LU pro husté matice typu $100n \times 150n$



Obrázek 3.6: Čas výpočtu GJE a LU pro husté matice typu $100n \times 150n$

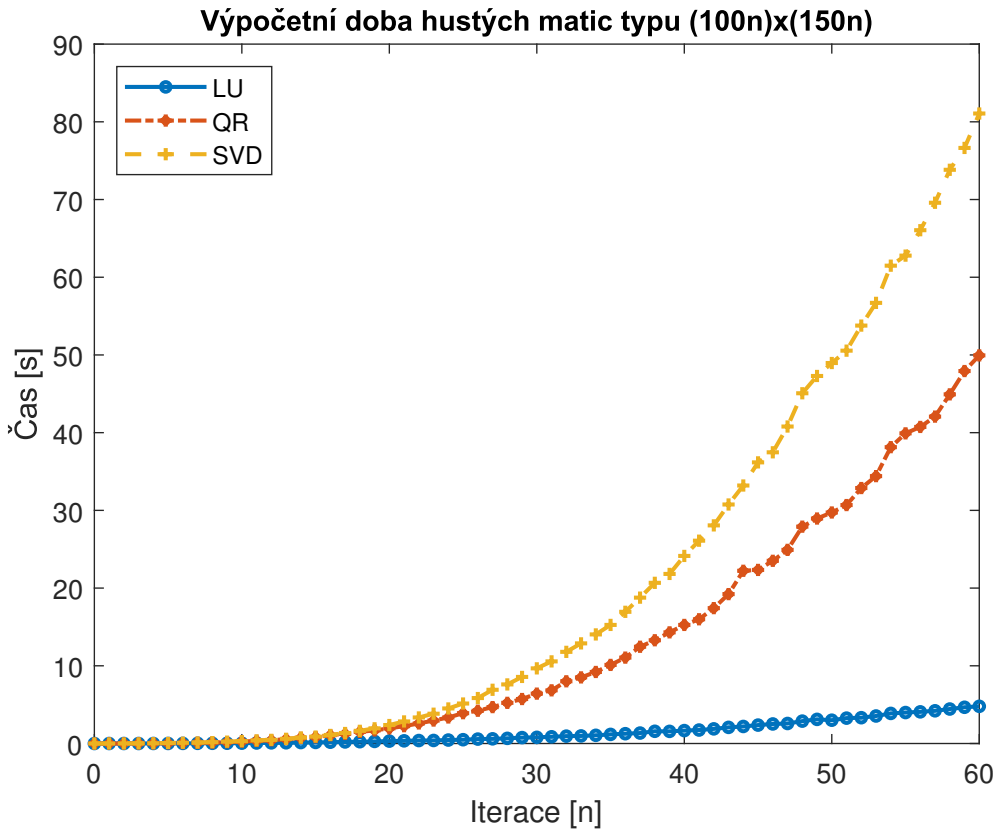
Podobně jako u prvního měření má i zde metoda LU nejmenší výpočetní dobu

$< 10s$. Výrazně vyšší výpočetní čas má metoda QR s časem $< 55s$ a metoda SVD s časem $< 90s$. Vidíme, že výpočetní doba LU roste s rostoucí dimenzí úlohy velmi pomalu a je to tedy velmi efektivní metoda pro řešení velkých úloh z hlediska výpočetní doby.



Obrázek 3.7: Norma chyby pro husté matice typu $100n \times 150n$

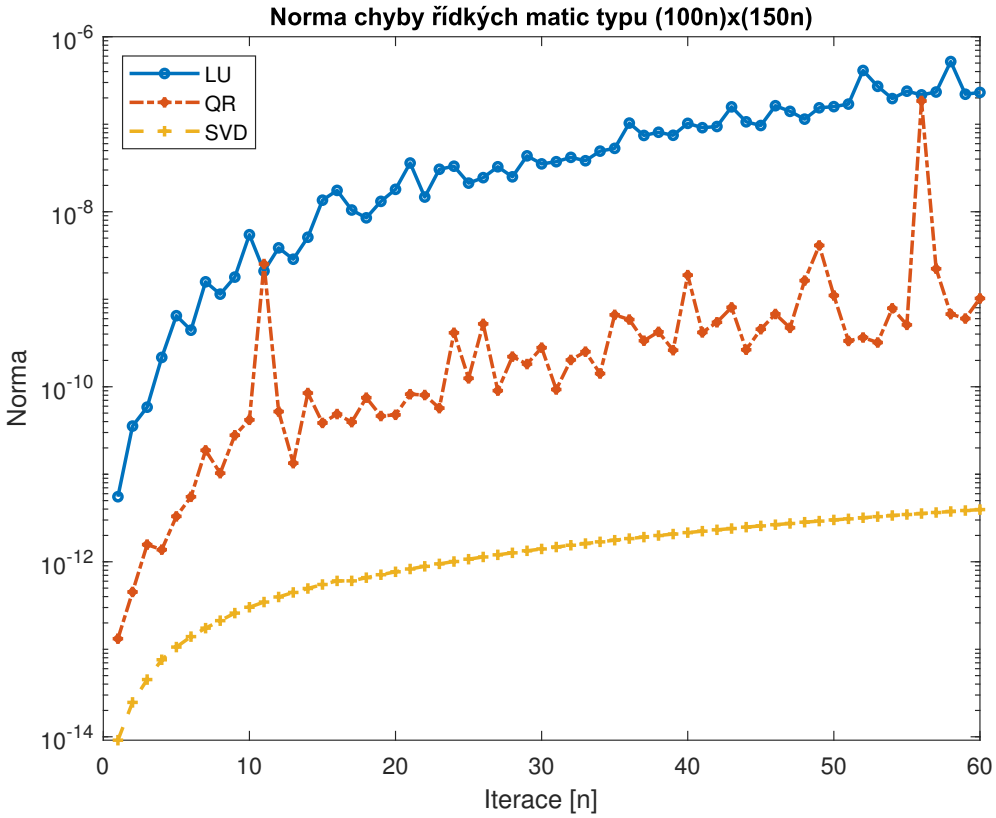
Na druhou stranu metoda LU podle grafu 3.7 má výrazně rychleji rostoucí normu chyby s hodnotami $< 6 \cdot 10^{-7}$. Podobná ztráta přesnosti je i u metody GJE 3.5 s hodnotami $< 4.5 \cdot 10^{-9}$ v prvních 20 krocích, ovšem bychom čekali, že i v dalších krocích by nabývala přibližně stejných hodnot jako LU, podobně jako v předchozím případě širokých matic. Přesnější jsou opět podle 3.7 metody QR s hodnotami $< 1.2 \cdot 10^{-10}$ a SVD s hodnotami $< 2 \cdot 10^{-11}$. Oproti malým úlohám jsou všechny metody méně přesné, ovšem GJE a LU ztrácejí přesnost rychleji a SVD je naopak stabilní s menší ztrátou přesnosti s rostoucím rozměrem úlohy.



Obrázek 3.8: Čas výpočtu pro husté matice typu $100n \times 150n$

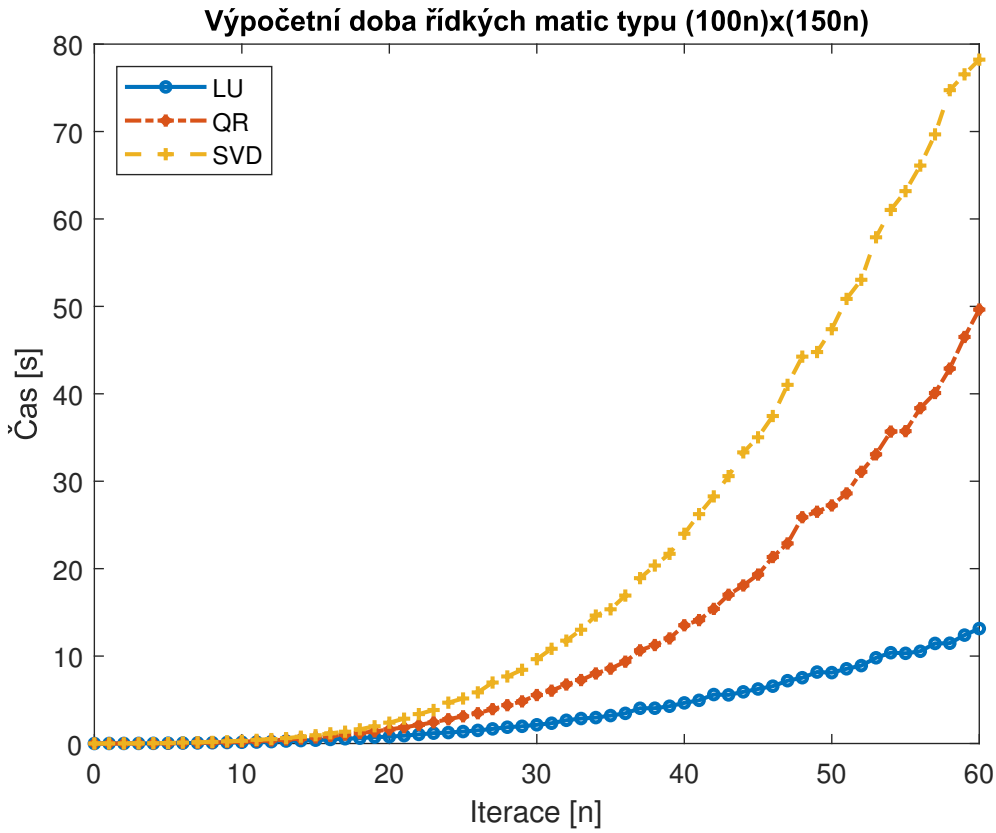
Řídké matice

Stejně jako pro všechna ostatní měření i zde podle grafu 3.9 je SVD stabilní a nejpřesnější s hodnotami normy $< 4 \cdot 10^{-12}$. QR je podobně jako u malých úloh i zde pro řídké matice nestabilní s větší normou chyby oproti hustým maticím s hodnotami $< 2.5 \cdot 10^{-9}$ až na vyjímky. Podobně i LU rozklad pro řídké matice ztrácí přesnost oproti hustým maticím s hodnotami $< 6 \cdot 10^{-7}$.



Obrázek 3.9: Norma chyby pro řídké matice typu $100n \times 150n$

Z hlediska výpočetní doby podle grafu 3.10 má i zde metoda LU nejkratší výpočetní dobu s časem $< 15s$, ovšem je téměř dvakrát pomalejší oproti hustým maticím. Toto je způsobené tím, že je potřeba zajistit dobrou podmíněnost horního trojuhelníkového bloku matice U, aby byl dobře invertovatelný. Výrazně delší výpočetní čas má metoda QR s hodnotami $< 55s$, zatímco výpočetní čas SVD roste nejrychleji s hodnotami $< 85s$.



Obrázek 3.10: Čas výpočtu pro řídké matice typu $100n \times 150n$

Závěr

V rámci této práce jsme si na začátku vysvětlili základní myšlenky a teorii pro různé rozklady matic používáné prohledání nulového prostoru matic. Díky těmto rozkladům jsme schopni přímo vzorcem vyjádřit hledanou matici nulového prostoru.

Tyto metody mají různé teoretické vlastnosti výpočetní náročnosti a numerické stability, které jsme v práci popsali.

V poslední části práce jsme metody implementovali v programovacím prostředí MATLAB a měřili jsme jejich výpočetní přesnost a dobu výpočtu na náhodně generovaných reálných maticích s plnou řádkovou hodností..

- V prvním případě jsme použili metody na malé husté a řídké matice. V obou případech byla výpočetní doba všech metod kromě GJE v řádu pár milisekund, GJE má téměř desetinásobnou výpočetní dobu oproti ostatním metodám a téměř dvojnásobnou výpočetní dobu pro řídké matice oproti hustým maticím. V obou případech je SVD nejpřesnější, a zatímco pro husté matice je přesnost SVD srovnatelná s QR, tak pro řídké matice je QR zatížena nestabilitami, které jsou v případě hustých a řídkých matic přítomné i u LU a GJE. Tyto nestability jsou způsobeny špatnou podmíněností invertované matice.

- V druhém případě jsme měření prováděli pro velké husté a řidké matice. Jelikož už u malých úloh měla GJE mnohonásobně větší výpočetní dobu než ostatní metody, které pro velké matice jsou v řádu několika minut, tak pro velké úlohy je ze stejného důvodu prakticky nepoužitelná. Pro husté matice je z hlediska výpočetní doby zdaleka nejefektivnější LU s výpočetní dobou pro největší úlohu $< 5s$. Pro husté i řídké matice je nejpřesnější SVD. V případě LU pro řídké matice nepřesnosti ve výpočtech mohou způsobit, že invertovaná matice je singulární, tedy je nutné použít přesnější varianty LU rozkladu. Podobně jako pro řídké matice pro malé úlohy, i zde je QR méně přesné a nestabilní oproti hustým maticím.

V obou částech měření výsledky odpovídají našim teoretickým předpokladům.

Do budoucna by bylo zajímavé přejít k úplně řídkým implementacím, které umožňují vyhnout se většině nestabilit. Soudobé algoritmy pro LU i QR rozklady umožňují efektivně určovat hodnost matice a zároveň jsou použitelné i pro matice velkých dimenzí. Experimenty by bylo také vhodné roztřídit podle různých aplikací a ukázat začlenění algoritmů do širšího výpočetního kontextu, kde je báze nulového prostoru potřeba.

Seznam použité literatury

Golub, G. H. a Van Loan, C. F. (1996). *Matrix Computations*. The Johns Hopkins University Press, Baltimore and London, fourth edition.

Gómez, M. A. (2005). A null-space method for computing the search direction in the general inertia-controlling method for dense quadratic programming. *European Journal of Operational Research*, **161**(3), 655–662. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2003.04.002. URL `https://www.sciencedirect.com/science/article/pii/S0377221703006775`.

Hall, C. A., Peterson, J. S., Porsching, T. A. a Sledge, F. R. (1985). The dual variable method for finite element discretizations of Navier/Stokes equations. *International Journal for Numerical Methods in Engineering*, **21**, 883–898. doi: 10.1002/nme.1620210508.

Heath, M. T., Plemmons, R. J. a Ward, R. C. (1984). Sparse orthogonal schemes for structural optimization using the force method. *Siam Journal on Scientific and Statistical Computing*, **5**, 514–532. URL `https://api.semanticscholar.org/CorpusID:121080472`.

Tebbens, J., Hnětynková, I., Plešinger, M., Strakoš, Z. a Tichý, P. (2012). *Analýza metod pro maticové výpočty*. První vydání. Matfyzpress, Praha. ISBN 978-80-7378-201-6.

Watkins, D. S. (2002). *Fundamentals of Matrix Computations*. Pure and Applied Mathematics. Wiley-Interscience, New York, second edition. ISBN 0-471-21394-2. doi: 10.1002/0471249718. URL `https://doi.org/âĂę718`.

Seznam obrázků

1 Orientovaný graf G s šesti vrcholy 2

3.1 Norma chyby pro husté matice typu $100 \times (100 + 40n)$ 18
3.2 Čas výpočtu pro husté matice typu $100 \times (100 + 40n)$ 19
3.3 Norma chyby pro řídké matice typu $100 \times (100 + 40n)$ 20
3.4 Čas výpočtu pro řídké matice typu $100 \times (100 + 40n)$ 21
3.5 Norma chyby GJE a LU pro husté matice typu $100n \times 150n$ . . . 22
3.6 Čas výpočtu GJE a LU pro husté matice typu $100n \times 150n$ 22
3.7 Norma chyby pro husté matice typu $100n \times 150n$ 23
3.8 Čas výpočtu pro husté matice typu $100n \times 150n$ 24
3.9 Norma chyby pro řídké matice typu $100n \times 150n$ 25
3.10 Čas výpočtu pro řídké matice typu $100n \times 150n$ 26