

**MATEMATICKO-FYZIKÁLNÍ  
FAKULTA**  
Univerzita Karlova

**BAKALÁŘSKÁ PRÁCE**

Andrea Kučerová

**Regularizační vlastnosti Krylovovských  
metod**

Katedra numerické matematiky

Vedoucí bakalářské práce: RNDr. Iveta Hnětynková, Ph.D.

Studijní program: Matematika

Studijní obor: Obecná matematika

Praha 2019

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

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

Velké díky patří vedoucí mé bakalářské práce RNDr. Ivetě Hnětynkové, Ph.D. za spoustu času, který mi v průběhu psaní práce věnovala, a za její velikou trpělivost a cenné rady.

Název práce: Regularizační vlastnosti Krylovovských metod

Autor: Andrea Kučerová

Katedra: Katedra numerické matematiky

Vedoucí bakalářské práce: RNDr. Iveta Hnětynková, Ph.D., Katedra numerické matematiky

Abstrakt: Cílem této práce je studovat a popsat regularizační vlastnosti iteračních Krylovovských metod pro řešení lineárních algebraických *ill-posed* problémů zatížených bílým šumem. Nejprve popíšeme vlastnosti těchto problémů, především vysokou citlivost na změny v datech. Ukážeme, že klasické metody pro řešení aproximačních úloh (jako například metoda nejmenších čtverců) zde selhávají. Proto se budeme věnovat objasnění regularizačních vlastností projekcí na Krylovův prostor. Uvedeme základní Krylovovské regularizační metody, konkrétně RRGMRRES, CGLS a LSQR, a ilustrujeme jejich chování na modelových příkladech z Regularizačního toolboxu v prostředí MATLAB.

Klíčová slova: Krylovův prostor, iterační proces, ill-posed úlohy, regularizace

Title: Regularization properties of Krylov subspace methods

Author: Andrea Kučerová

Department: Department of Numerical Mathematics

Supervisor: RNDr. Iveta Hnětynková, Ph.D., Department of Numerical Mathematics

Abstract: The aim of this thesis is to study and describe regularizing properties of iterative Krylov subspace methods for finding a solution of linear algebraic *ill-posed* problems contaminated by white noise. First we explain properties of this kind of problems, especially their sensitivity to small perturbations in data. It is shown that classical methods for solving approximation problems (such as the least squares method) fail here. Thus we turn to explanation of regularizing properties of projections onto Krylov subspaces. Basic Krylov regularizing methods are considered, namely RRGMRRES, CGLS, and LSQR. The results are illustrated on model problems from Regularization toolbox in MATLAB.

Keywords: Krylov subspace, iterative process, ill-posed problems, regularization

# Obsah

<b>Značení</b>	<b>2</b>
<b>Úvod</b>	<b>3</b>
<b>1 Úvod do inverzních problémů</b>	<b>4</b>
1.1 Základní definice a věty . . . . .	4
1.2 Vlastnosti problému . . . . .	6
1.3 Naivní řešení . . . . .	9
<b>2 Krylovovy prostory</b>	<b>13</b>
2.1 Zavedení Krylovova prostoru a související pojmy . . . . .	13
2.2 Výpočet báze Krylovova prostoru . . . . .	14
<b>3 Regularizace pomocí projekce</b>	<b>19</b>
3.1 Regularizační vlastnosti Krylovových prostorů . . . . .	19
3.2 Vybrané Krylovovské metody . . . . .	21
3.3 Numerické srovnání . . . . .	26
<b>Závěr</b>	<b>29</b>
<b>Seznam použité literatury</b>	<b>30</b>

# Značení

$\mathbb{N}$	množina přirozených čísel
$\mathbb{R}$	množina reálných čísel
$\mathbb{R}^n$	množina reálných vektorů dimenze $n$
$\mathbb{R}^{m \times n}$	množina reálných matic typu $m \times n$
$\dim(\mathcal{V})$	dimenze vektorového prostoru $\mathcal{V}$
$\text{span}\{v_1, \dots, v_n\}$	lineární obal vektorů $v_1, \dots, v_n$
$\langle x; y \rangle$	skalární součin vektorů $x$ a $y$
$\mathcal{K}_k(A, b)$	$k$ -tý Krylovův podprostor pro matici $A$ a vektor $b$
$\overrightarrow{\mathcal{K}_k(M, v)}$	$k$ -tý shiftovaný Krylovův prostor
$\text{rank}(A)$	hodnota matice $A$
$\text{Im}(A)$	obraz matice $A$ , neboli sloupcový prostor
$\text{Ker}(A)$	jádro matice $A$ , neboli nulový prostor
$\det(A)$	determinant matice $A$
$A^T$	transpozice matice $A$
$I$	jednotková matice
$\text{diag}(a_1, \dots, a_n)$	diagonální matice s prvky $a_1, \dots, a_n$ na diagonále
$\ x\ $	euklidovská norma vektoru $x$
$\ x\ _A$	$A$ -norma vektoru $x$
$\ A\ $	spektrální norma matice $A$
$\kappa(A)$	číslo podmíněnosti matice $A$

# Úvod

Při řešení reálných úloh z nejrůznějších oblastí, jako jsou geologie, lékařství, zpracování obrazu a mnohá další, se setkáváme s potřebou řešit lineárně algebraické problémy

$$Ax \approx b, A \in \mathbb{R}^{m \times n}, x \in \mathbb{R}^n, b \in \mathbb{R}^m,$$

kde  $A$  a  $b$  jsou známé (nenulové) a  $x$  je hledaný vektor. Takovéto úlohy vznikají většinou jako diskretizace spojité úlohy a mají často vlastnosti, které komplikují jejich řešení. Pochází-li z diskretizace spojitého inverzního problému, bývají citlivé na malé změny vstupních dat  $A$  a  $b$ , obsahují chyby (šum) vzniklé při diskretizaci, měření a ukládání dat a podobně, viz. [1, Kap. 2-3]. Navíc matice úlohy  $A$  (reprezentující diskretizovaný model) je často špatně podmíněná, její singulární čísla klesají plynule k nule bez výrazných mezer a tedy je i numerická hodnota  $A$  špatně definovaná, viz. [1, s. 7-9]. Pro problémy s takovými vlastnostmi používáme označení *ill-posed* problémy.

Klasické metody pro řešení lineárních algebraických aproximačních problémů, jako je metoda nejmenších čtverců, viz. [2, Kap. 6], v případě ill-posed problémů selhávají, protože si neumí poradit s citlivostí aproximace řešení na chyby v datech. Je tedy nutné použít jiné metody, které počítají řešení méně citlivé. Takovým metodám říkáme *regularizační*, viz. [3, Kap. 4]. Základním příkladem regularizační metody je TSVD [1, Kap. 4.2], založená na aproximaci matice  $A$  za využití singulárního rozkladu [2, Kap. 5.2]. Pro reálné úlohy se však lépe uplatní iterační metody [1, Kap. 6], jako například RRGMRRES [4], CGLS [5, Kap. 7.4.1] a LSQR [6], které využívají regularizačních vlastností Krylovových prostorů [2, Kap. 9].

Cílem této práce je studium regularizačních vlastností iteračních Krylovových metod. Zaměříme se nejprve na vlastnosti vektorů generujících Krylovův prostor. Pro tento účel uvedeme známé postupy pro konstrukci jeho ortonormální báze – Arnoldiho [7, Kap. 6.3] a Lanczosův proces [7, Kap. 6.6]. Popíšeme chování projekcí na Krylovův prostor při růstu jeho dimenze. Následně shrneme metody RRGMRRES, CGLS a LSQR. Pro zjednodušení zde uvažujeme, že šum je v úloze obsažen pouze ve vektoru  $b$  (vektor pozorování). Navíc předpokládáme pouze tak zvaný *bílý* šum, jehož složky pochází z normálního rozdělení se střední hodnotou v nule, viz. [1, s. 39-47]. Výsledky ilustrujeme na dvou jednoduchých modelových úlohách z Regularizačního toolboxu [8] v prostředí MATLAB.

V Kapitole 1 zavádíme potřebné pojmy a popisujeme vlastnosti ill-posed problémů. Vysvětlíme, proč je v případě řešení ill-posed úloh nevhodné použít metodu nejmenších čtverců. Popisujeme základní regularizační metodu TSVD.

Kapitola 2 má za cíl zavést pojmy související s Krylovovými prostory a popsat problematiku hledání vhodné báze Krylovova prostoru v obecném a symetrickém případě.

Konečně v Kapitole 3 přichází na řadu rozbor regularizačních vlastností Krylovových prostorů. Popisujeme zde základní iterační Krylovovské metody vhodné pro řešení ill-posed úloh a ilustrujeme jejich chování numerickými experimenty.

# 1. Úvod do inverzních problémů

V první kapitole si připomeneme základní definice a věty nutné pro pochopení celé problematiky a obecné vlastnosti inverzních problémů, jako jsou zesilování chyb v průběhu jejich řešení a nevhodnost použití tzv. naivního řešení.

## 1.1 Základní definice a věty

Následující pojmy lze definovat nad komplexními čísly, pro naše potřeby však bude stačit zavést je nad tělesem reálných čísel.

Základní úvahy se týkají spektra matice, zavedeme tedy pojem vlastních čísel. Podrobnější rozbor spektrálních vlastností matic popisuje [2, s. 8], případně [9, s. 15], [10, s. 320-328] a [11].

**Definice 1** (Vlastní číslo). *Mějme čtvercovou matici  $A \in \mathbb{R}^{n \times n}$ , kde  $n \in \mathbb{N}$ . Reálné číslo  $\lambda$  nazveme vlastním číslem matice  $A$ , pokud existuje nenulový vektor  $x \in \mathbb{R}^n$  takový, že platí*

$$Ax = \lambda x.$$

Dalším základním pojmem týkajícím se matic a vektorů je norma. Obecnou definici lze najít v [2, s. 8]. My zde uvedeme pouze konkrétní příklady vektorových norem, se kterými budeme dále pracovat. Prvním příkladem je euklidovská norma na  $\mathbb{R}^n$ :

**Definice 2** (Euklidovská norma na  $\mathbb{R}^n$ ). *Mějme vektor  $x = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ . Euklidovskou normou vektoru  $x$  rozumíme reálné číslo*

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

Budeme potřebovat také pojem  $A$ -normy:

**Definice 3** ( $A$ -norma na  $\mathbb{R}^n$ ). *Mějme čtvercovou pozitivně definitní symetrickou matici  $A \in \mathbb{R}^{n \times n}$  a vektor  $x \in \mathbb{R}^n$ .  $A$ -normou vektoru  $x$  rozumíme reálné číslo*

$$\|x\|_A = \sqrt{x^T A x},$$

kde  $x^T$  je vektor transponovaný k  $x$ .

Fakt, že takto definovaná zobrazení jsou opravdu normy, zde nebudeme dokazovat, viz např. [2, s. 8-13]. Dále si připomeneme pojem skalárního součinu, který je s normou úzce spjatý. Opět neuvеdeme obecnou definici skalárního součinu, ale pouze definice skalárních součinů příslušných výše definovaným normám, tedy standardního skalárního součinu a součinu definovaného pomocí konkrétní pozitivně definitní hermitovské matice  $A \in \mathbb{R}^{n \times n}$ . Ukážeme si vzájemný vztah normy a skalárního součinu.

**Definice 4** (Standardní skalární součin v  $\mathbb{R}^n$ ). *Mějme dva reálné vektory  $x = (x_1, x_2, \dots, x_n)^T$  a  $y = (y_1, y_2, \dots, y_n)^T$ . Standardním skalárním součinem vektorů  $x$  a  $y$  rozumíme reálné číslo*

$$\langle x; y \rangle = y^T x = \sum_{i=1}^n x_i y_i.$$

Podobně vypadá součin definovaný pomocí matice  $A$ .

**Definice 5** (Skalární součin pomocí matice). *Mějme čtvercovou pozitivně definitní symetrickou matici  $A \in \mathbb{R}^{n \times n}$  a vektory  $x, y \in \mathbb{R}^n$ . Skalárním součinem pomocí matice  $A$  vektorů  $x$  a  $y$  rozumíme reálné číslo*

$$\langle x; y \rangle_A = y^T A x.$$

Z předchozích definic lze nahlédnout vztahy těchto norem a skalárních součinů. Pro vektor  $x \in \mathbb{R}^n$  tedy platí následující:

$$\begin{aligned} \|x\| &= \sqrt{\langle x; x \rangle}; \\ \|x\|_A &= \sqrt{\langle x; x \rangle_A}. \end{aligned}$$

Pro zavedení a vysvětlení vlastností naivního řešení v následující kapitole a odvození dalších pojmů je třeba singulárního rozkladu. K tomu je nutné definovat hodnotu a singulární číslo matice.

**Definice 6** (Hodnota matice). *Mějme matici  $A \in \mathbb{R}^{m \times n}$ , kde  $m, n \in \mathbb{N}$ . Její hodnotou rozumíme číslo  $\text{rank}(A) = \dim(\text{Im}(A))$ .*

**Definice 7** (Singulární číslo matice). *Mějme matici  $A \in \mathbb{R}^{m \times n}$ , kde  $m, n \in \mathbb{N}$ . Je-li  $\lambda$  nenulové vlastní číslo matice  $A^T A$ , pak číslo  $\sigma = \sqrt{\lambda}$  nazveme singulárním číslem matice  $A$ .*

Konečně se dostáváme k vyslovení věty o singulárním rozkladu. Uvedeme zde maticovou verzi této věty, opět pouze v reálném případě. Odvození a podrobnější rozbor lze nalézt v [2, s. 125-129].

**Věta 8** (O singulárním rozkladu). *Mějme matici  $A \in \mathbb{R}^{m \times n}$ , kde  $m, n \in \mathbb{N}$ . Necht  $r$  je její hodnota. Potom existují kladná reálná čísla  $\sigma_1, \dots, \sigma_r$  a ortogonální matice  $U = (u_1, \dots, u_m) \in \mathbb{R}^{m \times m}$  a  $V = (v_1, \dots, v_n) \in \mathbb{R}^{n \times n}$  takové, že platí vztah*

$$A = U \Sigma V^T = \sum_{i=1}^r \sigma_i u_i v_i^T, \quad (1.1)$$

kde

$$\Sigma = \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{m \times n}, \Sigma_r = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r \times r}.$$

Vektory  $u_1, \dots, u_m$  potom nazýváme levými singulárními vektory a vektory  $v_1, \dots, v_n$  pravými singulárními vektory  $A$ . Čísla  $\sigma_1, \dots, \sigma_r$  jsou singulární čísla matice  $A$ . Výraz  $\sum_{i=1}^r \sigma_i u_i v_i^T$  nazveme diadickým rozvojem matice  $A$ .

*Poznámka.* Je-li  $r = \text{rank}(A)$  hodnota matice  $A$ , pak singulární čísla  $\sigma_1, \dots, \sigma_r$  uvažujeme seřazená od největšího po nejmenší. Takové uspořádání se nám bude později hodit.

V dalším také budeme potřebovat pojem *Moore-Penroseovy pseudoinverze*, který je zobecněním pojmu inverze pro obecné matice.

**Definice 9** (Moore-Penroseova pseudoinverze matice). *Mějme matici  $A \in \mathbb{R}^{m \times n}$ , kde  $m, n \in \mathbb{N}$ . Necht  $r$  je její hodnota a  $A = U\Sigma V^T$  její singulární rozklad. Potom Moore-Penroseovou pseudoinverzí matice  $A$  rozumíme matici*

$$A^\dagger = V\Sigma^\dagger U^T,$$

kde

$$\Sigma^\dagger = \begin{pmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{n \times m}.$$

Pro vlastnosti  $A^\dagger$  viz. [2, Kap. 5.3.1], nebo [12, Kap. 5.5]. Definujeme také *číslo podmíněnosti matice*.

**Definice 10** (Číslo podmíněnosti matice). *Mějme regulární čtvercovou matici  $A \in \mathbb{R}^{n \times n}$  a  $\sigma_1, \dots, \sigma_n$  její singulární čísla. Potom číslem podmíněnosti matice  $A$  rozumíme číslo*

$$\kappa(A) = \frac{\sigma_1}{\sigma_n} = \frac{\|A\|}{\|A^{-1}\|}.$$

Poznamenejme, že pro  $A \in \mathbb{R}^{m \times n}$  lze číslo podmíněnosti zobecnit na

$$\kappa(A) = \frac{\sigma_1}{\sigma_r},$$

kde  $\text{rank}(A) = r$  (přičemž obvykle se předpokládá  $\text{rank}(A) = \min(m, n)$ ).

## 1.2 Vlastnosti problému

Uvažujme lineární aproximační problém

$$Ax \approx b, A \in \mathbb{R}^{m \times n}, x \in \mathbb{R}^n, b \in \mathbb{R}^m, \quad (1.2)$$

kde  $A$  a  $b$  jsou známé (nenulové) a  $x$  je hledaný vektor. Budeme předpokládat, že  $A^T b \neq 0$ , jinak problém nelze dobře vyřešit a obvykle má smysl uvažovat pouze triviální řešení (nulový vektor). Vektor pravé strany  $b$  může obsahovat chyby (šum) vzniklé například při měření, diskretizaci, ukládání a zaokrouhlování jeho prvků atd. Cílem je získat dobrou aproximaci vektoru  $x$  odpovídajícího neznámé úloze bez chyb.

Podívejme se nyní na vlastnosti vektoru pravé strany  $b$ . Jak jsme už zmiňovali,  $b$  zpravidla obsahuje šum. Uvažujme tedy

$$b = \tilde{b} + e, \quad (1.3)$$

kde  $e$  reprezentuje šum a  $\tilde{b}$  značí (neznámou) přesnou pravou stranu, pro niž předpokládáme, že platí

$$A\tilde{x} = \tilde{b}, \quad (1.4)$$

kde  $\tilde{x}$  je hledané přesné řešení problému bez šumu. Budeme předpokládat, že šum v problému (1.2) je takzvaně bílý, neboli složky vektoru  $e$  pochází ze stejného normálního rozdělení se střední hodnotou v nule. Více o jiných typech šumu např. v [1, s. 39-47].

Definujme nyní *hladinu šumu* ve vektoru  $b$ :

**Definice 11** (Hladina šumu v  $b$ ). Mějme  $b$  vektor pravé strany z problému (1.2). Potom hladinou šumu v  $b$  rozumíme číslo

$$\mu_e = \frac{\|e\|}{\|b\|}.$$

Uvažujme problém (1.2) pocházející z diskretizace inverzního problému, například integrální rovnice s hladkým jádrem viz. [1, Kap. 2.1]. Potom je jeho řešení ovlivněno několika vlastnostmi, které nyní rozebereme. Budeme je ilustrovat na problémech *deriv2(100)* a *shaw(100)* z [8] s přidaným šumem na hladině  $\mu_e = 10^{-4}$ . Používáme zde dva různé příklady, protože ilustrují různé chování, které může nastat. Konstrukci problémů uvádíme v následujících příkladech:

*Příklad 1* (Konstrukce *deriv2(100)* s přidaným šumem na hladině  $\mu_e = 10^{-4}$ ).

$$\begin{aligned} [A, b_{exact}, x] &= \text{deriv2}(100); \\ hl &= 10^{-4}; \\ e &= \text{randn}(100, 1); \\ e &= \text{norm}(b_{exact}) * e / \text{norm}(e); \\ e &= hl * e; \\ b &= b_{exact} + e; \end{aligned}$$

V konstrukci problému matice  $A$  reprezentuje model, vektor  $x$  je přesné řešení a  $b_{exact}$  přesná pravá strana. Dále  $hl$  reprezentuje zvolenou hladinu šumu,  $e$  je náhodný vektor z normálního rozdělení a  $b$  je výsledná pravá strana problému obsahující šum.

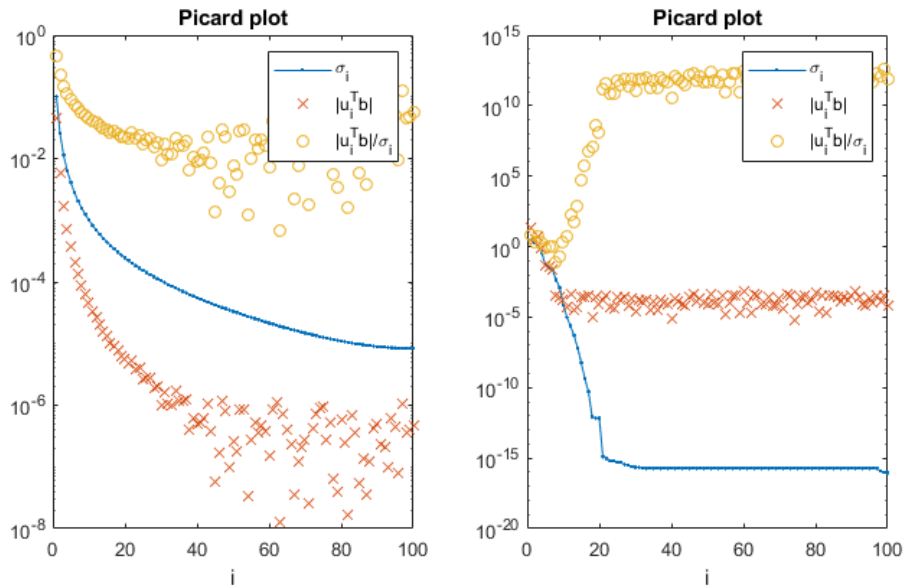
*Příklad 2* (Konstrukce *shaw(100)* s přidaným šumem na hladině  $\mu_e = 10^{-4}$ ). Konstrukce probíhá podobně jako pro problém *deriv2(100)* v Příkladu 1, pouze první řádek je nahrazen následujícím příkazem.

$$[A, b_{exact}, x] = \text{shaw}(100);$$

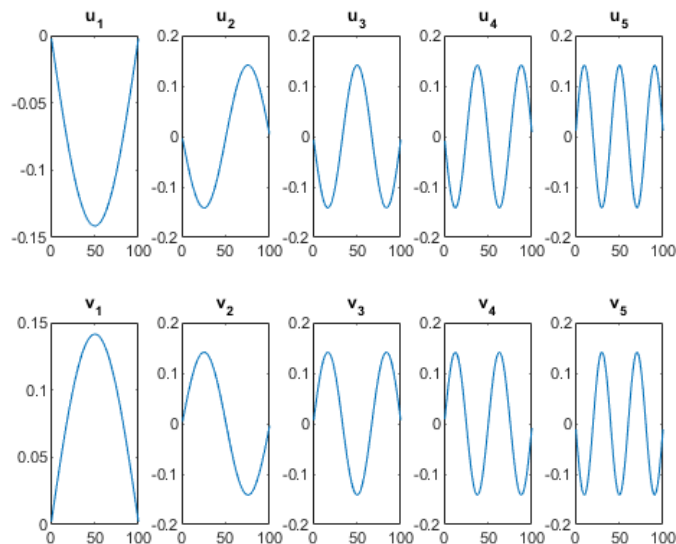
Všechny ostatní kroky jsou stejné.

Začněme tedy maticí  $A$ . Uvažujme její singulární rozklad  $A = U\Sigma V^T$ .

- $A$  má tu vlastnost, že její singulární čísla  $\sigma_1, \dots, \sigma_r$  klesají v průměru k nule, jako vidíme na Obrázku 1.1, který obsahuje tak zvaný Picardův graf používaný standardně pro numerickou analýzu aproximačních problémů. Pro vykreslení využíváme funkci *picard(U,s,b)* z [8], kde  $\Sigma = \text{diag}(s)$ . Tedy je  $A$  typicky špatně podmíněná,  $\kappa(A)$  je velké.
- V levých i pravých singulárních vektorech  $u_1, \dots, u_m$  a  $v_1, \dots, v_n$  matice  $A$  se pro velké indexy častěji střídají znaménka složek. Při vykreslení vektorů se potom zdají být vektory s většími indexy „méně hladké“, a proto říkáme, že mají „větší frekvence“, jako na Obrázku 1.2. Tento fakt souvisí s tím, že matice  $A$  má *vlastnost zhlazování*, protože vychází typicky z nějakého přirozeného zhlazujícího procesu. Například při fotografování dochází k rozmazávání hranic jednotlivých barev. Singulární vektory příslušné velkým singulárním číslům mají tedy menší frekvence než singulární vektory příslušné malým singulárním číslům.



Obrázek 1.1: Picardovy grafy. Vlevo pro problém z Příkladu 1. Singulární čísla klesají. Velikosti projekcí  $|u_i^T b|$  nejprve klesá a poté se ustálí poblíž hladiny související s  $\mu_e$ . Koeficienty  $|u_i^T b|/\sigma_i$  také nejprve klesají, poté začnou zvolna růst. Vpravo problém z Příkladu 2. Zde je působení šumu mnohem výraznější, hladina šumu je o deset řádů výš než hladina singulárních čísel. Je zde vidět rozdíl v rozdělení singulárních čísel matic z obou problémů. Zatímco vlevo vidíme plynulý pokles a singulární čísla jsou všechna výrazně vyšší než strojová přesnost, vpravo je vidět výrazný pokles a všechna singulární čísla počínaje dvacátým jsou na úrovni strojové přesnosti.



Obrázek 1.2: Prvních pět levých a pravých singulárních vektorů matice  $A$  (vykreslených po složkách) pro problém z Příkladu 1. Vidíme, že s rostoucím indexem se ve vektorech častěji střídají znaménka složek.

Podrobnější vysvětlení uvedených vlastností je založené na Riemann–Lebesgueově lemmatu a lze jej nalézt v [1, s. 7-9].

Podívejme se nyní na vektor pravé strany  $b$  a jeho části  $\tilde{b}$  a  $e$ . Připomeňme, že vektory  $\tilde{b}$  a  $e$  (ani jejich normy) v reálných úlohách neznáme, máme k dispozici pouze jejich součet  $b$ . Hladina šumu není v praxi také známa.

- Jelikož má  $A$  vlastnost zhlazování, lze předpokládat, že vektor  $\tilde{b}$  je hladký. Když tedy budeme  $\tilde{b}$  projektovat do báze  $u_1, \dots, u_m$  levých singulárních vektorů, potom očekáváme, že velikost projekcí  $u_i^T \tilde{b}$  bude se vzrůstajícím indexem klesat. Obvykle dokonce platí, že tyto projekce klesají rychleji než singulární čísla  $A$ , tedy koeficienty  $\frac{|u_i^T \tilde{b}|}{\sigma_i}$  pro rostoucí  $i$  v průměru klesají. Říkáme, že  $\tilde{b}$  splňuje *diskrétní Picardovu podmínku*, viz. [1, s. 33-37].
- Naproti tomu vektor  $e$  vznikl náhodně, tedy jeho projekce  $u_i^T e$  do vektorů báze budou přes všechny indexy srovnatelně velké a se vzrůstajícím indexem nebudou nijak významně klesat, viz [1, s. 41-42]. Tedy vektor  $e$  *diskrétní Picardovu podmínku nespĺňuje*.

V Picardových grafech na Obrázku 1.1 vidíme, že koeficienty  $\frac{|u_i^T b|}{\sigma_i}$  nejprve klesají a potom v průměru rostou. Ve vztahu

$$\frac{|u_i^T b|}{\sigma_i} = \left| \frac{u_i^T \tilde{b}}{\sigma_i} + \frac{u_i^T e}{\sigma_i} \right|$$

velikost prvního členu součtu dle *diskrétní Picardovy podmínky* pro  $\tilde{b}$  klesá, z čehož vyplývá, že chování koeficientů  $\frac{|u_i^T b|}{\sigma_i}$  je pro vysoké indexy výrazně deformováno složkou  $e$ .

Problém (1.2) s takovými vlastnostmi, jako jsme právě popsali, nazýváme *ill-posed* problém a jeho řešení je citlivé na změny vstupních dat, tedy malé výchyly v  $b$  mohou způsobit velké rozdíly ve výsledném  $x$ , jak ukážeme v následující kapitole. Řešení *ill-posed* problému může být navíc *nejednoznačné*.

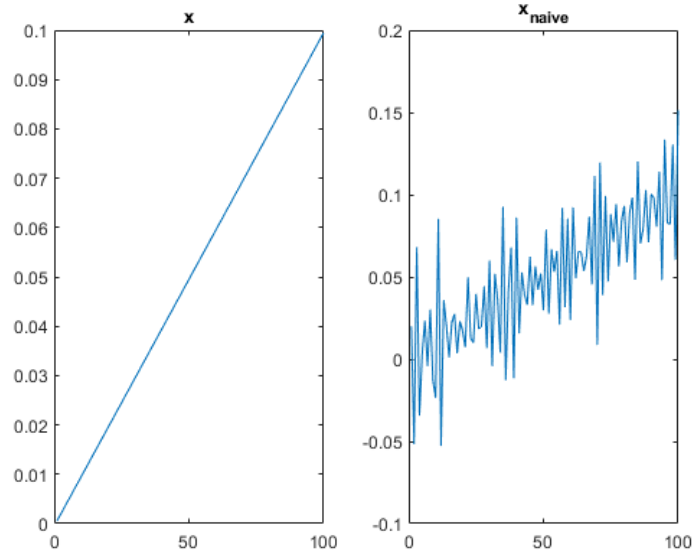
### 1.3 Naivní řešení

Uvažujme nyní pro jednoduchost, že  $A$  je čtvercová regulární matice typu  $n \times n$ . Mějme její singulární rozklad (1.1), problém (1.2) pak má přesné řešení, které lze vyjádřit pomocí inverze matice  $A$

$$x_{naive} = A^{-1}b = (U\Sigma V^T)^{-1}b = V\Sigma^{-1}U^T b = \sum_{i=1}^n \frac{u_i^T b}{\sigma_i} v_i. \quad (1.5)$$

Pokud porovnáme přesný vektor  $\tilde{x}$  z problému v Příkladu 1 (kde  $A$  je čtvercová a regulární) s vektorem (1.5), uvidíme, že  $x_{naive}$  hledanému řešení vůbec neodpovídá, viz Obrázek 1.3. To je také důvod, proč se takto spočtená aproximace označuje jako *naivní řešení*.

Podívejme se nyní na to, čím je chyba způsobena. Z (1.5) vidíme, že  $x_{naive}$  lze vyjádřit v bázi pravých singulárních vektorů  $A$ , přičemž  $\frac{u_i^T b}{\sigma_i}$  jsou koeficienty



Obrázek 1.3: Srovnání přesného řešení  $\tilde{x}$  (vlevo) a naivního řešení (vpravo) problému z Příkladu 1. Vidíme, že naivní řešení  $x_{naive}$  neodpovídá hledanému vektoru  $\tilde{x}$ , navíc lze ve vektoru  $x_{naive}$  pozorovat velké rozdíly mezi jeho prvky (oscilace).

příslušné lineární kombinace. Víme, že vektor  $b$  se skládá ze dvou složek,  $b = \tilde{b} + e$ , rozepišme tedy podrobněji vztah (1.5):

$$x_{naive} = A^{-1}(\tilde{b} + e) = \sum_{i=1}^n \frac{u_i^T \tilde{b}}{\sigma_i} v_i + \sum_{i=1}^n \frac{u_i^T e}{\sigma_i} v_i.$$

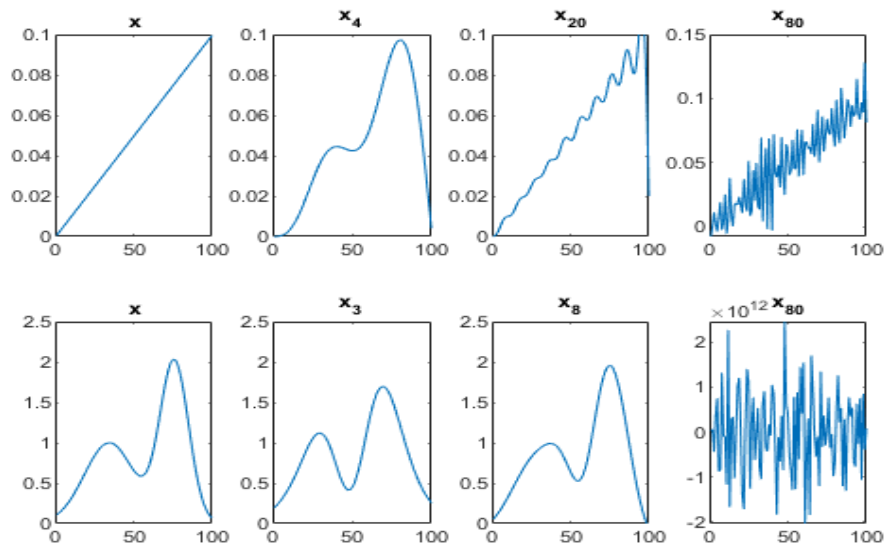
První člen reprezentuje přesný vektor  $\tilde{x}$  a druhý chybu v aproximaci (1.5). Všimněme si koeficientů v obou součtech. Víme z Kapitoly 1.2, že  $\tilde{b}$  splňuje diskrétní Picardovu podmínku, tedy koeficienty  $\frac{u_i^T \tilde{b}}{\sigma_i}$  z prvního součtu klesají velice rychle k nule. Naopak o druhém členu víme, že výrazy  $u_i^T e$  jsou přes všechny indexy řádově srovnatelně velké, tudíž pokud je dělíme singulárními čísly, budou velikosti příslušných koeficientů  $\frac{u_i^T e}{\sigma_i}$  velmi rychle stoupat. Členům naivního řešení bude tedy pro větší indexy zcela dominovat druhá, chybová část součtu. Obdobně tomu bude i pro obecnou matici  $A$ , místo inverze  $A^{-1}$  se využije *Moore-Penroseova pseudoinvertze*  $A^\dagger$  matice  $A$ , viz. Definice 9. V takovém případě bude vektor  $x_{naive}$  představovat řešení ve smyslu nejmenších čtverců problému (1.2). Můžeme tedy zavést obecnou definici naivního řešení.

**Definice 12** (Naivní řešení). *Mějme  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ . Nechť  $r = \text{rank}(A)$ . Pak vektor*

$$x_{naive} = A^\dagger b = \sum_{i=1}^r \frac{u_i^T b}{\sigma_i} v_i$$

*nazveme naivním řešením ill-posed problému  $Ax \approx b$ .*

Základní metoda řešení problému (1.2), kterou teď uvedeme, je založena na následující větě, která nám dává aproximaci matice maticí nižší hodnoti ve smyslu minimalizace normy rozdílu. Důkaz lze nalézt v [2, s. 133].



Obrázek 1.4: Srovnání přesného řešení  $\tilde{x}$  (vlevo) a řešení problémů z Příkladů 1 a 2, pomocí aproximace z Věty 13. Nahoře Problém 1 pro  $k = 4, 20, 40$ . Vidíme, že vektor  $x_{20}$  aproximuje  $\tilde{x}$  mnohem lépe než  $x_{naive}$ , viz. Obrázek 1.3, zatímco  $x_4$  obsahuje málo informace a  $x_{40}$  je ovlivněno šumem a je rozoscilované. Dole Problém 2 pro  $k = 3, 8, 80$ . V tomto případě je výsledek mnohem uspokojivější než v předchozím případě. Navíc stačilo mnohem menší  $k$ . Obecně je těžké nalézt vhodný regularizační parametr  $k$ .

**Věta 13** (Eckart–Young–Mirsky). *Mějme matici  $A \in \mathbb{R}^{m \times n}$ , kde  $m, n \in \mathbb{N}$ . Necht  $r$  je její hodnota a  $1 < k < r$ . Je-li  $A = \sum_{i=1}^r \sigma_i u_i v_i^T$  singulární rozklad  $A$ , potom pro matici*

$$A_k = \sum_{i=1}^k \sigma_i u_i v_i^T$$

platí, že

$$A_k = \underset{\substack{\text{rank}(X)=k \\ X \in \mathbb{R}^{m \times n}}}{\text{argmin}} \|X - A\|,$$

neboli  $A_k$  je nejlepší aproximace  $A$  maticí hodnosti  $k$ . Navíc máme

$$\|A_k - A\| = \sigma_{k+1}.$$

V naivním řešení dle Definice 12 jsou problematické členy s vysokými indexy, protože jsou příslušná singulární čísla příliš malá. Pokud se nám podaří odhadnout správnou míru aproximace, tedy hodnotu  $k$  z Věty 13, můžeme nahrazením (1.2) problémem

$$A_k x_k \approx b \tag{1.6}$$

dostat často lepší výsledek, jak vidíme na Obrázku 1.4, protože matice  $A_k$  má v určitém smyslu lepší numerické vlastnosti než  $A$ . Řešení

$$x_k = A_k^\dagger b = \sum_{i=1}^k \frac{u_i^T b}{\sigma_i} v_i$$

úlohy (1.6) pro vhodné  $k$  aproximuje  $\tilde{x}$  lépe než naivní řešení  $x_{naive}$ . Tato nejjednodušší regularizační metoda se nazývá *Truncated SVD* (TSVD) a parametr  $k$  nazveme *regularizačním parametrem*, viz. [3, s. 50]. Obecně je ale problém nalézt vhodné  $k$ . Pokud zvolíme  $k$  příliš malé, ochudíme se o velké množství informací z původních dat  $A, b$  a aproximace řešení nebude dostatečná. Pokud bude naopak  $k$  příliš velké, výsledek  $x_k$  může být ovlivněn šumem, stejně jako v případě naivního řešení. Oba tyto případy vidíme na Obrázku 1.4 pro problémy z Příkladů 1 a 2. Pro některé úlohy vůbec neexistuje vhodné  $k$ , jehož použití by vedlo k dobré aproximaci řešení úlohy (1.2) pomocí (1.6). Metoda TSVD je navíc výpočetně náročná, protože je v průběhu nutné počítat alespoň část SVD rozkladu. Proto se v následující kapitole zaměříme na jiný typ regularizace, která často vede na lepší výsledky.

## 2. Krylovovy prostory

Kapitola 1.3 nám poskytla základní metodu pro aproximaci řešení úlohy (1.2), ilustrovali jsme i roli regularizačního parametru a důležitost jeho vhodné volby. V této kapitole se dostaneme k zavedení pojmu *Krylovova prostoru*. Uvedeme některé jeho vlastnosti a budeme se zabývat postupy, které slouží k výpočtu jeho báze v obecném a symetrickém případě.

### 2.1 Zavedení Krylovova prostoru a související pojmy

Začněme definicí Krylovova prostoru.

**Definice 14** (Krylovův prostor). *Mějme čtvercovou matici  $M \in \mathbb{R}^{n \times n}$ , vektor  $v \in \mathbb{R}^n$  a  $k \leq n$ , kde  $k, n \in \mathbb{N}$ . Pak prostor*

$$\mathcal{K}_k(M, v) = \text{span}\{v, Mv, \dots, M^{k-1}v\}$$

*nazveme  $k$ -tým Krylovovým prostorem příslušným matici  $M$  a vektoru  $v$ .*

Zamysleme se nyní nad dimenzí prostoru  $\mathcal{K}_k(M, v)$ . Všechny vektory z definiční množiny  $\{v, Mv, \dots, M^{k-1}v\}$  leží ve sloupcovém prostoru  $\text{Im}(M)$  matice  $M$ , tedy jejich lineární obal bude mít dimenzi nejvýše rovnou hodnotě  $\text{rank}(M) < \infty$ . Díky tomu má smysl definovat následující pojem.

**Definice 15** (Stupeň vektoru  $v$  vzhledem k matici  $M$ ). *Mějme čtvercovou matici  $M \in \mathbb{R}^{n \times n}$  a vektor  $v \in \mathbb{R}^n$ . Potom číslo*

$$d(M, v) = \min\{k \in \mathbb{N} \mid \dim(\mathcal{K}_k(M, v)) = \dim(\mathcal{K}_{k+1}(M, v))\}$$

*nazveme stupněm vektoru  $v$  vzhledem k matici  $M$ .*

Díky pozorování z předchozího odstavce je Definice 15 korektní, neboť dimenze prostorů  $\mathcal{K}_k(M, v)$  nemůže s rostoucím  $k$  růst do nekonečna a definiční množina je tedy neprázdná. Vidíme, že  $\dim(\mathcal{K}_k(M, v)) \leq d(M, v)$ . V závislosti na  $k$  lze ale odvodit dokonce následující vztahy pro dimenzi Krylovova prostoru  $\mathcal{K}_k(M, v)$ , viz. též [7, Kap. 6.2] nebo [13, Kap. 12].

**Lemma 16** (Dimenze Krylovova prostoru  $\mathcal{K}_k(M, v)$ ). *Mějme čtvercovou matici  $M \in \mathbb{R}^{n \times n}$ , vektor  $v \in \mathbb{R}^n$  a  $k \leq n$ , kde  $k, n \in \mathbb{N}$ . Platí následující vztahy pro dimenzi Krylovova prostoru  $\mathcal{K}_k(M, v)$ :*

$$\dim(\mathcal{K}_k(M, v)) = \begin{cases} k & k \leq d(M, v) \\ d(M, v) & k > d(M, v). \end{cases}$$

*Důkaz.* V případě, kdy  $k \leq d(M, v)$ , plyne rovnost ihned z Definice 15. Předpokládejme tedy, že  $k > d(M, v)$ . K důkazu stačí ukázat, že pokud pro nějaké  $r \in \mathbb{N}$  platí  $\dim(\mathcal{K}_r(M, v)) = \dim(\mathcal{K}_{r+1}(M, v))$ , je potom i  $\dim(\mathcal{K}_{r+1}(M, v)) = \dim(\mathcal{K}_{r+2}(M, v))$ . Toto nyní dokážeme.

Z předpokladu

$$\dim(\mathcal{K}_r(M,v)) = \dim(\mathcal{K}_{r+1}(M,v))$$

plyne, že  $M^r v \in \mathcal{K}_r(M,v)$ . Proto

$$M^r v = \alpha_0 v + \alpha_1 Mv + \dots + \alpha_{r-1} M^{r-1} v$$

pro nějaké koeficienty  $\alpha_0, \dots, \alpha_{r-1} \in \mathbb{R}$ . Použitím tohoto vztahu dostaneme

$$\begin{aligned} M^{r+1} v &= M(M^r v) \\ &= M(\alpha_0 v + \alpha_1 Mv + \dots + \alpha_{r-1} M^{r-1} v) \\ &= \alpha_0 Mv + \alpha_1 M^2 v + \dots + \alpha_{r-1} M^r v \in \mathcal{K}_{r+1}(M,v), \end{aligned}$$

tedy platí i dokazovaná rovnost

$$\dim(\mathcal{K}_{r+1}(M,v)) = \dim(\mathcal{K}_{r+2}(M,v)).$$

Z Definice 15 víme, že  $\dim(\mathcal{K}_{d(M,v)}(M,v)) = \dim(\mathcal{K}_{d(M,v)+1}(M,v))$  a užitím právě dokázaného dostaneme následující vztahy:

$$\mathcal{K}_1(M,v) \subsetneq \dots \subsetneq \mathcal{K}_{d(M,v)}(M,v) = \mathcal{K}_{d(M,v)+1}(M,v) = \dots = \mathcal{K}_k(M,v) = \dots$$

Zřejmě

$$\dim(\mathcal{K}_k(M,v)) = \dim(\mathcal{K}_{d(M,v)}(M,v)) = d(M,v),$$

kde druhá rovnost plyne z první části tvrzení. Tím je důkaz hotov. □

Pro využití v konkrétních metodách je třeba znát bázi Krylovova prostoru  $\mathcal{K}_k(M,v)$ . Ihned je vidět, že vektory  $v, Mv, \dots, M^{d(M,v)-1} v$  tvoří bázi prostoru  $\mathcal{K}_k(M,v)$ , která se nazývá *Krylovova*. Pro použití ve většině metod je ale taková báze numericky nevhodná. Při výpočtu v konečné aritmetice pro její vektory s většími indexy platí, že jsou téměř (numericky) lineárně závislé na předchozích vektorech, viz. [14, Kap. 2.4]. Cílem další kapitoly bude popsat výpočet numericky vhodnější báze Krylovova prostoru  $\mathcal{K}_k(M,v)$ .

## 2.2 Výpočet báze Krylovova prostoru

V této kapitole se zaměříme na způsoby výpočtu ortonormální báze Krylovova prostoru  $\mathcal{K}_k(M,v)$ . Základním přístupem je *Arnoldiho algoritmus*, viz. [15]. Tento proces popíšeme a následně ukážeme, že jeho výsledkem je opravdu ortonormální báze prostoru  $\mathcal{K}_k(M,v)$ .

Vstupními parametry Arnoldiho algoritmu jsou matice  $M$ , vektor  $v$  a koeficient  $k$ , který určuje počet iterací. Budeme pro zjednodušení předpokládat, že  $1 \leq k < d(M,v)$ , pro další iteraci by proces vygeneroval už jen nulový vektor. Posledním parametrem je číslo  $reo \in \{0,1\}$ , které udává počet reortogonalizací, které lze uvažovat, chceme-li při výpočtu v konečné aritmetice zlepšit ortogonalitu spočtených bázevých vektorů.

Jako první vektor  $v_1$  nové báze vezmeme znormovaný vektor  $v$ . Dále se postupuje v každé iteraci následovně. Vektor  $v_j$  vytvoříme vynásobením vektoru  $v_{j-1}$  maticí  $M$  zleva, následnou ortogonalizací vůči všem předchozím vektorům  $v_1, \dots, v_{j-1}$  a znormováním.

---

**Algoritmus 1** Arnoldiho algoritmus (MGS)

---

**Vstup:**  $M, v, k, reo$ **Výstup:**  $H_{k+1,k}, V_{k+1}$ 

```
 $v_1 = \frac{v}{\|v\|}$ 
for  $j = 1, \dots, k$  do
   $w = Mv_j$ 
  for  $i = 1, \dots, j$  do
     $h_{i,j} = v_i^T w$ 
     $w = w - h_{i,j}v_i$ 
  end for
  for  $p = 1, \dots, reo$  do
    for  $i = 1, \dots, j$  do
       $c = v_i^T w$ 
       $h_{i,j} = h_{i,j} + c$ 
       $w = w - cv_i$ 
    end for
  end for
   $h_{j+1,j} = \|w\|$ 
   $v_{j+1} = \frac{w}{\|w\|}$ 
end for
```

---

*Poznámka.* Více informací o Arnoldiho algoritmu lze nalézt v původním článku viz. [15]. Jiné varianty implementace Arnoldiho algoritmu (například CGS) viz. [7, s. 146-157] nebo [2, s. 172].

*Poznámka.* V případě, kdyby  $k = d(M, v)$ , bude  $h_{k+1,k} = 0$  a potřebujeme, aby se algoritmus zastavil. Z toho důvodu se při výpočtu v konečné aritmetice do Algoritmu 1 přidává *zastavovací kritérium*, tedy kontrola, zda není norma posledního spočteného vektoru  $w$  menší než zvolená tolerance  $TOL$ .

Všimněme si, že počet operací vykonaných za jednu iteraci Algoritmu 1 se s rostoucím počtem vektorů zvyšuje. Roste tím i výpočetní náročnost tohoto algoritmu, říkáme, že je založen na *dlouhých rekurencích*. Výhodou těchto dlouhých rekurencí je obvykle poměrně pomalá ztráta ortogonality. Při použití reortogonalizace se přesnost ještě zvyšuje. Pro  $reo = 1$  bylo dokázáno, že ortogonalita spočtených vektorů je pro nepříliš špatně podmíněnou matici  $M$  zachována na strojové přesnosti, pro podrobnosti a předpoklady viz. [16].

Uvažujme nyní pouze přesnou aritmetiku a  $k < d(M, v)$ . Výstupem algoritmu jsou navzájem ortonormální vektory  $v_i$ ,  $i = 1, \dots, k + 1$  a koeficienty  $h_{i,j}$ ,  $j = 1, \dots, k$ ,  $i = 1, \dots, k + 1$ . Označme  $V_{k+1} = (v_1, \dots, v_k, v_{k+1})$  matici výsledných vektorů a  $H_k$  matici koeficientů následujícím způsobem:

$$H_k = \begin{pmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,k} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,k} \\ & \ddots & \ddots & \vdots \\ & & h_{k,k-1} & h_{k,k} \end{pmatrix} \in \mathbb{R}^{k \times k},$$

kde zbývající prvky v dolním trojúhelníku jsou nulové. Matice tohoto tvaru se nazývají *horní Hessenbergovy*. Všimněme si navíc, že prvky na poddiagonále jsou kladné, neboť se jedná o normy nenulových vektorů.

Označíme-li

$$H_{k+1,k} = \begin{pmatrix} H_k \\ 0 \dots 0, h_{k+1,k} \end{pmatrix} \in \mathbb{R}^{(k+1) \times k},$$

dostaneme

$$MV_k = V_{k+1}H_{k+1,k} = \tag{2.1}$$

$$= V_k H_k + h_{k+1,k} v_{k+1} e_k^T. \tag{2.2}$$

Dále lze psát

$$V_k^T MV_k = H_k, \tag{2.3}$$

což lze interpretovat tak, že matice  $H_k$  je projekcí (zúžením) matice  $M$  jako operátoru na  $\mathbb{R}^n$  na podprostor  $\mathcal{K}_k(M, v)$ .

Zbývá nám ukázat, že matice  $V_k$  má ve sloupcích ortonormální bázi prostoru  $\mathcal{K}_k(M, v)$ .

**Lemma 17.** *Mějme čtvercovou matici  $M \in \mathbb{R}^{n \times n}$ , vektor  $v \in \mathbb{R}^n$  a  $k < d(M, v)$ , kde  $k, n \in \mathbb{N}$ . Potom vektory  $v_1, \dots, v_k$  z Arnoldiho algoritmu tvoří ortonormální bázi prostoru  $\mathcal{K}_k(M, v)$ .*

*Důkaz.* Důkaz provedeme indukcí. Pro  $k \leq d(M, v)$  dokážeme, že ve vyjádření  $v_k$  vzhledem k bázi  $\{v, Mv, \dots, M^{k-1}v\}$  je koeficient u  $M^{k-1}v$  nenulový a že  $\{v_1, \dots, v_k\}$  tvoří bázi  $\mathcal{K}_k(M, v)$ .

Pro  $k = 1$  je tvrzení zřejmé. Pro  $k = 2$  je

$$Mv_1 = M \frac{v}{\|v\|} \in \mathcal{K}_2(M, v) \setminus \mathcal{K}_1(M, v).$$

Tedy  $v_1$  a  $Mv_1$  jsou lineárně nezávislé vektory a výsledná množina  $\{v_1, v_2\}$  tvoří ortonormální bázi  $\mathcal{K}_2(M, v)$ . Navíc koeficient u  $Mv$  je roven  $1/\|v\| \neq 0$ . Nechť tvrzení platí pro nějaké  $k \geq 2$ . Vyjádříme  $v_k$  vzhledem k bázi  $\{v, Mv, \dots, M^{k-1}v\}$ ,

$$v_k = \alpha_0 v + \dots + \alpha_{k-1} M^{k-1} v,$$

kde  $\alpha_{k-1} \neq 0$  z předpokladu. Potom máme

$$Mv_k = \alpha_0 Mv + \dots + \alpha_{k-1} M^k v,$$

z čehož plyne, že  $Mv_k$  není lineárně závislý na předchozích vektorech a tedy, po odečtení projekcí  $v_i^T Mv_k$  pro  $i = 1, \dots, k$  a znormování, tvoří množina

$$\{v_1, \dots, v_{k+1}\}$$

ortonormální bázi  $\mathcal{K}_{k+1}(M, v)$ . Navíc koeficient u  $M^k v$  se při odečítání projekcí nemění, protože  $M^k v$  není v prostoru  $\text{span}\{v_1, \dots, v_k\}$ , a při normalizaci se pouze přeškáluje, tedy je nenulový. □

Nalezli jsme tedy ortonormální bázi Krylovova prostoru  $\mathcal{K}_k(M, v)$ . Uvažujme nyní Arnoldiho algoritmus aplikovaný na matici  $M$  symetrickou. Z rovnosti (2.3) dostaneme

$$H_k = V_k^T MV_k = V_k^T M^T V_k = (V_k^T MV_k)^T = H_k^T,$$

tedy i  $H_k$  je symetrická matice. Díky tomu, že je navíc horní Hessenbergova, je už  $H_k$  nutně tridiagonální. Dále ji budeme značit  $T_k$ ,

$$T_k = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_k \\ & & & \beta_k & \alpha_k \end{pmatrix} \in \mathbb{R}^{k \times k}.$$

Vidíme, že v případě symetrické matice  $M$  je v Arnoldiho algoritmu mnoho koeficientů nulových, celý proces se zjednoduší na tříčlennou rekurenci. Navíc, díky symetrii  $T_k$ , je jeden z koeficientů roven normalizačnímu koeficientu z předchozí iterace, tedy ho není třeba znovu počítat. Takto modifikovaný algoritmus nazveme *Lanczosovou tridiagonalizací*, viz. též [17]. Dostáváme následující analogii vztahu (2.3)

$$V_k^T M V_k = T_k.$$

Dále také dostaneme vztahy

$$\begin{aligned} M V_k &= V_{k+1} T_{k+1,k} = \\ &= V_k T_k + \beta_{k+1} v_{k+1} e_k^T, \end{aligned}$$

analogické jako pro Arnoldiho algoritmus.

---

### Algoritmus 2 Lanczosova tridiagonalizace

---

**Vstup:**  $M, v, k, reo$

**Výstup:**  $T_{k+1,k}, V_k$

```

 $v_1 = \frac{v}{\|v\|}$ 
for  $j = 1, \dots, (k - 1)$  do
  if  $j = 1$  then
     $w = M v_j$ 
  else
     $w = M v_j - \beta_j v_{j-1}$ 
  end if
   $\alpha_j = v_j^T w$ 
   $w = w - \alpha_j v_j$ 
  for  $p = 1, \dots, reo$  do
    for  $i = 1, \dots, j$  do
       $c = v_i^T w$ 
       $w = w - c v_i$ 
    end for
  end for
   $\beta_{j+1} = \|w\|$ 
   $v_{j+1} = \frac{w}{\|w\|}$ 
end for

```

---

Předchozí úvahy implikují následující lemma.

**Lemma 18.** *Mějme čtvercovou symetrickou matici  $M \in \mathbb{R}^{n \times n}$ , vektor  $v \in \mathbb{R}^n$  a  $k \leq d(M, v)$ , kde  $k, n \in \mathbb{N}$ . Potom se Arnoldiho proces pro  $M$  a  $v$  redukuje na tříčlennou rekurenci.*

Výhoda Lanczosovy tridiagonalizace spočívá v tom, že v každé iteraci je třeba provést stejný počet operací, říkáme, že jde o algoritmus založený na *krátkých rekurencích*. Tato vlastnost je ale zároveň jeho nevýhodou, protože v konečné aritmetice (bez použití ortogonalizace vůči všem předchozím vektorům) se může stát, že vektory se vzdálenými indexy na sebe nebudou (numericky) kolmé. Ztráta ortogonality je běžná vlastnost krátkých rekurencí, viz. [14, Kap. 5.9]. Proto se do algoritmu Lanczosovy tridiagonalizace často přidává (někdy i vícenásobná) reortogonalizace,  $reo \in \{0, 1, 2\}$ .

*Poznámka.* Lze provádět různé varianty reortogonalizace. Výše uvedená je tak zvaná *plná reortogonalizace*, kde každý spočtený vektor ortogonalizujeme proti všem předchozím. Dostáváme tedy dlouhé rekurence jako v Algoritmu 1. Existují však i jiné přístupy (například parciální reortogonalizace), kde lze snížit výpočetní náklady za cenu určité ztráty ortogonality oproti plné reortogonalizaci. Pro další techniky viz. [13].

### 3. Regularizace pomocí projekce

V této kapitole se zaměříme na regularizaci problému (1.2) pomocí projekce. Nejprve na jednoduché volbě  $\mathcal{K}_k(A,b)$  vysvětlíme, proč mají projekce na Krylovovy prostory regularizační vlastnosti. Pak uvedeme metody s ještě sofistikovanější volbou Krylovova prostoru a budeme testovat jejich efektivitu.

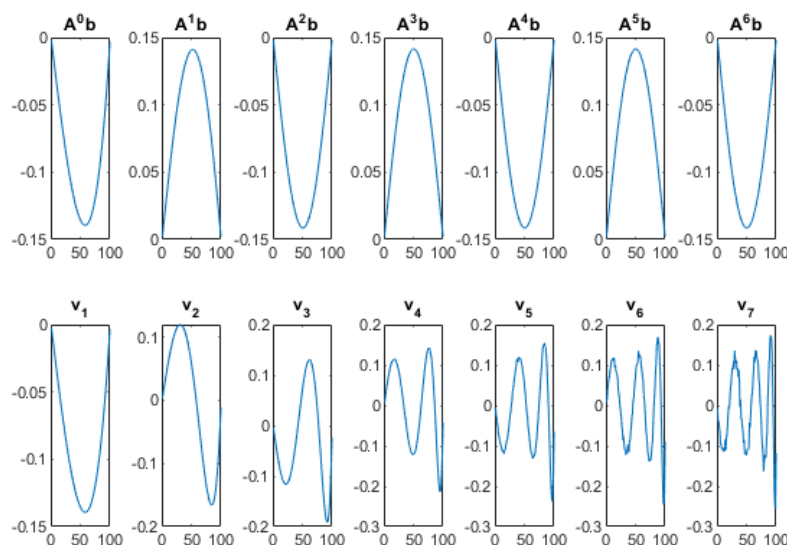
#### 3.1 Regularizační vlastnosti Krylovových prostorů

Kapitola 1.3 nám nabídla metodu TSVD, která spočívala v nahrazení matice  $A$  maticí  $A_k$  z Věty 13. Tím jsme (1.2) aproximovali problémem (1.6). Podobnou úvahu můžeme učinit s Krylovovými prostory. Namísto Věty 13 použijeme tentokrát vztah (2.3). Uvažujme, že máme k dispozici data  $A \in \mathbb{R}^{n \times n}$  a  $b \in \mathbb{R}^n$ . Nejjednodušší možností je uvažovat Krylovův prostor pro volbu  $M \equiv A$  a  $v \equiv b$ , tedy

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

pro nějaké  $1 \leq k \leq d(A,b)$ . Potom Arnoldiho procesem získáme jeho bázi ve sloupcích  $V_k$ , přičemž

$$V_k^T A V_k = H_k. \tag{3.1}$$



Obrázek 3.1: Srovnání Krylovovy (nahore) a Arnoldiho (dole) báze prostoru  $\mathcal{K}_7(A,b)$  pro problém z Příkladu 1. Pro vizualizaci jsou vektory Krylovovy báze vykresleny znormalizované. Všimněme si, že vektory Krylovovy báze jsou všechny srovnatelně hladké. Naopak pro vektory báze z Arnoldiho algoritmu platí, že hladké vektory jsou především ty s nižšími indexy, pro vyšší indexy pozorujeme ve vektorech stále větší oscilace.

Protože  $v_1 = b/\|b\|$  a  $V_k$  má ortonormální sloupce, tak vektor

$$V_k^T b = \|b\|e_1$$

představuje projekci pravé strany  $z$  (1.2) na Krylovův prostor.

Nyní chceme problém (1.2) nahradit problémem, který bude snáze řešitelný. Vynásobíme (1.2) zleva maticí  $V_k^T$ , dostaneme

$$V_k^T Ax \approx V_k^T b. \quad (3.2)$$

Uvažujme hledání aproximace vektoru  $\tilde{x}$ , viz. (1.4), na  $\mathcal{K}_k(A, b)$ , tedy chceme nalézt

$$x_k = V_k y_k \in \mathcal{K}_k(A, b),$$

kde  $y_k \in \mathbb{R}^k$  je vektor koeficientů  $x_k$  v bázi  $V_k$ . Pak dosazením do (3.2) máme

$$V_k^T A V_k y_k \approx V_k^T b.$$

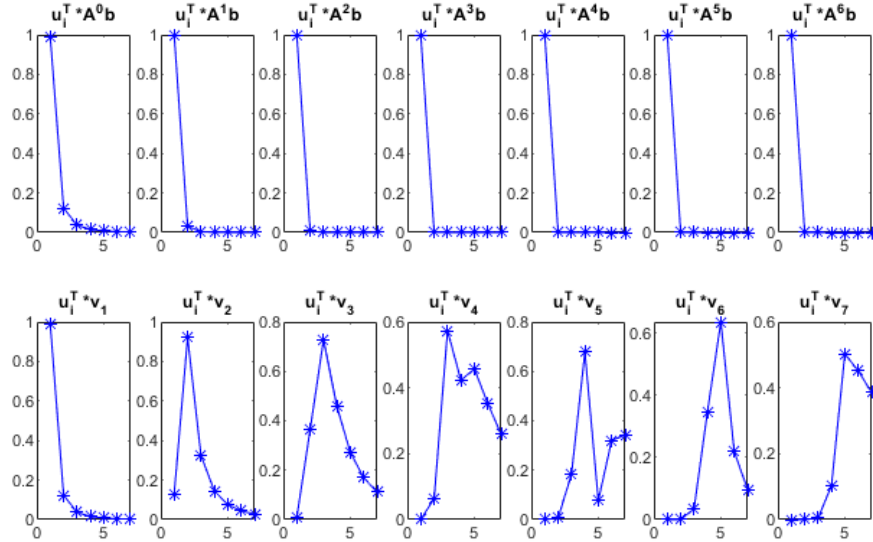
Z uvedeného dostáváme pomocí (3.1) nový problém

$$H_k y_k \approx \|b\| e_1, \quad (3.3)$$

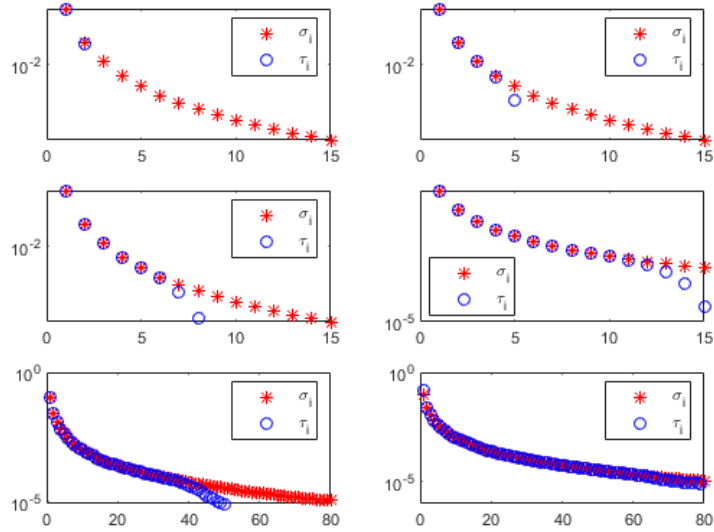
$$x_k = V_k y_k, \quad (3.4)$$

který představuje aproximaci problému (1.2) vzhledem k  $\mathcal{K}_k(A, b)$ , přičemž jeho dimenze je pouze  $k$ . Zaměříme se nyní na jeho vlastnosti.

Jak již víme, při numerickém řešení jsou problematické singulární vektory  $A$  odpovídající vyšším frekvencím, viz. Kapitola 1.3. Provádíme-li tedy regularizaci pomocí projekce, musí být nutně projekční prostor dominován hladkými bázovými vektory, viz. [1, Kap. 6.2]. Obrázek 3.1 nahoře ilustruje pomocí Krylovovy báze pro Příklad 1, že Krylovův prostor  $\mathcal{K}_k(A, b)$  je (pro malé  $k$ ) skutečně generován hladkými vektory. To je dáno násobením maticí  $A$ , která má zhlazující vlastnost. Dole pak vidíme Arnoldiho bázi spočtenou Algoritmem 1 (bez reortogonalizace).



Obrázek 3.2: Porovnání projekcí vektorů Krylovovy (nahore) a Arnoldiho (dole) báze prostoru  $\mathcal{K}_7(A, b)$  pro problém z Příkladu 1 do levých singulárních vektorů  $u_i$  matice  $A$ . Vidíme, že vektory Krylovovy báze v prvním řádku mají velké projekce ve směru prvního singulárního vektoru  $u_1$ . Naproti tomu je  $i$ -tý vektor Arnoldiho báze dominován singulárními vektory  $u_j$  pro které  $j \approx i$ .



Obrázek 3.3: Porovnání singulárních čísel matice  $A$  z problému v Příkladu 1 a její aproximace  $H_k$  z Arnoldiho algoritmu po řadě pro  $k = 2, 5, 8, 15, 50$  a  $80$ ,  $\sigma_i$  jsou singulární čísla  $A$  a  $\tau_i$  singulární čísla  $H_k$ . Všimněme si, že čísla  $\tau_i$  aproximují singulární čísla matice  $A$  postupně od těch největších.

Obrázek 3.2 ilustruje, že každý vektor  $v_i$  Arnoldiho báze je dominován jen několika levými singulárními vektory  $u_j$  pro které  $j \approx i$ . Jinými slovy, prostor  $\mathcal{K}_k(A, b)$  nese v sobě informace o prvních  $k$  levých singulárních vektorech. Proto má projekce na  $\mathcal{K}_k(A, b)$  regularizační vlastnosti. Pro vyšší  $k$  se díky ortogonalizaci dostávají do Arnoldiho báze také vyšší frekvence.

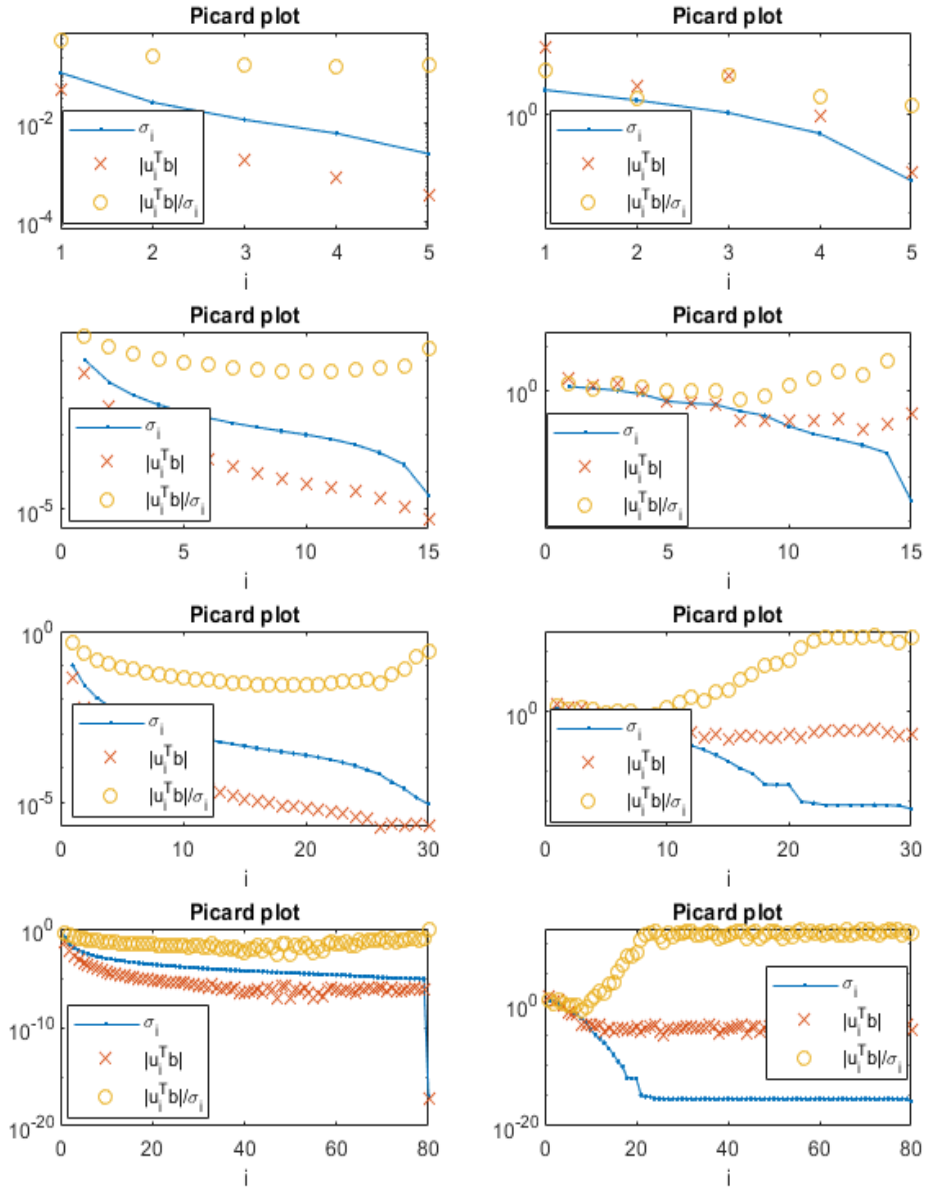
Platí, že singulární čísla matice  $H_k$  aproximují singulární čísla matice  $A$  od těch největších, jak vidíme na Obrázku 3.3 pro různé indexy  $k$ . Pro větší  $k$  aproximujeme větší počet singulárních čísel matice  $A$ . Obrázek 3.4 porovnává Picardovy grafy pro projektovaný problém (3.3) pro Příklad 1 a 2. Vidíme, že v závislosti na vlastnostech úlohy se efekt projekce na Krylovův prostor bude lišit. V každém případě má projektovaný problém lepší vlastnosti než problém původní. Obecně lze říci, že pro zvyšující se  $k$  aproximuje problém (3.3)-(3.4) problém (1.2) obdobně jako problém (1.6). Výpočet (3.3)-(3.4) je však výrazně méně nákladný oproti (1.6).

## 3.2 Vybrané Krylovovské metody

V této kapitole uvedeme konkrétní Krylovovské metody pro řešení ill-posed problému (1.2). Ve všech případech budeme pro jednoduchost uvažovat počáteční aproximaci  $x_0$  rovnu nule.

### GMRES a RRGMRRES

První metoda, kterou se budeme zabývat, je metoda GMRES, viz. [18], a od ní odvozená RRGMRRES, viz. [4]. Oba přístupy lze využít pouze v případě, že matice  $A$  v (1.2) je čtvercová, tedy  $m = n$ . Metoda GMRES hledá aproximaci



Obrázek 3.4: Porovnání Picardových grafů pro úlohy  $H_k x \approx \|b\| e_1$ , které aproximují (1.2) pro hodnoty  $k = 5, 15, 30, 80$  (odshora dolů). Levý sloupec přísluší problému z Příkladu 1, pravý problému z Příkladu 2. Vykresleno pomocí funkce  $picard(U_k, s_k)$  z [8], kde  $H_k = U_k \text{diag}(s_k) V_k$  je singulární rozklad matice  $H_k$ .

řešení (1.2) v prostoru  $\mathcal{K}_k(A,b)$ , a to tak, aby byla eukleidovská norma rezidua minimální. Metodu tedy můžeme formulovat následujícím způsobem. V každé iteraci  $k = 1, 2, \dots$  hledáme

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

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

Nejprve spočítáme bázi  $\mathcal{K}_k(A,b)$  Algoritmem 1. Necht  $V_k$  je matice z Arnoldiho procesu pro matici  $M \equiv A$  a vektor  $v \equiv b$ . Potom můžeme aproximaci řešení vyjádřit jako  $x_k = V_k y_k$ , kde  $y_k$  je vektor koeficientů  $x_k$  vzhledem k bázi ve sloupcích matice  $V_k$ . Pomocí tohoto vyjádření a vztahu (2.1) dostáváme:

$$\begin{aligned} \|r_k\| &= \|b - Ax_k\| = \\ &= \|b - AV_k y_k\| = \\ &= \|\|b\|V_{k+1}e_1 - V_{k+1}H_{k+1,k}y_k\| = \\ &= \|V_{k+1}(\|b\|e_1 - H_{k+1,k}y_k)\| = \\ &= \|\|b\|e_1 - H_{k+1,k}y_k\|, \end{aligned}$$

kde jsme využili faktu, že  $V_k$  má ortonormální sloupce a tedy nemění normu. Tudíž vektor koeficientů  $y_k$  je řešení ve smyslu nejmenších čtverců problému

$$H_{k+1,k}y_k \approx \|b\|e_1.$$

Zpětnou transformací  $x_k = V_k y_k$  dostaneme aproximaci řešení (1.2) metodou GMRES.

Takto formulovanou metodu lze dále upravovat. Jednou z možných modifikací je metoda RRGMRES, která hledá aproximaci  $x_k$  analogickým způsobem, avšak místo prostoru  $\mathcal{K}_k(A,b)$  využívá tak zvaný *shiftovaný prostor*, jehož definici nyní uvedeme.

**Definice 19** (Shiftovaný Krylovův prostor). *Mějme čtvercovou matici  $A \in \mathbb{R}^{n \times n}$ , vektor  $b \in \mathbb{R}^n$  a  $1 \leq k \leq n$ , kde  $k, n \in \mathbb{N}$ . Pak prostor*

$$\overrightarrow{\mathcal{K}_k(A,b)} = \text{span}\{Ab, A^2b, \dots, A^k b\}$$

*nazveme  $k$ -tým shiftovaným Krylovovým prostorem příslušným matici  $A$  a vektoru  $b$ .*

Shiftovaný prostor  $\overrightarrow{\mathcal{K}_k(A,b)}$  má lepší regularizační vlastnosti než Krylovův prostor  $\mathcal{K}_k(A,b)$ . To vychází ze skutečnosti, že neobsahuje vektor  $b$ , který je zatížen šumem  $e$ . Výpočet ortonormální báze prostoru  $\overrightarrow{\mathcal{K}_k(A,b)}$  probíhá analogicky Algoritmem 1, kde však volíme  $M \equiv A$  a  $v \equiv Ab$ . Metodu RRGMRES můžeme tedy formulovat jako hledání

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

$$\|r_k\| = \min_{x \in \overrightarrow{\mathcal{K}_k(A,b)}} \|b - Ax\|.$$

Vhodnou úpravou lze pak docílit toho, že metodu lze aplikovat i na úlohy s  $A$  singulární. Pro popis implementace viz. [19].

## CGLS a LSQR

Metody CGLS a LSQR jsou si matematicky ekvivalentní a lze je využít i pro obdélníkovou matici  $A \in \mathbb{R}^{m \times n}$ . Uvažujme řešení problému (1.2) ve smyslu nejmenších čtverců, neboli *naivní řešení* dle Definice 12. Vektor  $x_{naive}$  potom splňuje tak zvanou soustavu *normálních rovnic*

$$A^T A x_{naive} = A^T b.$$

Má-li matice  $A$  plnou sloupcovou hodnotu ( $rank(A) = n$ ), je potom matice  $A^T A$  symetrická a pozitivně definitní. V tom případě můžeme na tuto soustavu aplikovat metodu *sdrůžených gradientů* (CG), viz. [2, Kap. 8]. Tento postup lze potom zapsat jako hledání

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

$$\|x_k - x_{naive}\|_{A^T A} = \min_{x \in \mathcal{K}_k(A^T A, A^T b)} \|x - x_{naive}\|_{A^T A},$$

postupně pro  $k = 1, 2, \dots$ . Potom

$$\begin{aligned} \|x_k - x_{naive}\|_{A^T A}^2 &= \\ &= \langle A^T A(x_k - x_{naive}); x_k - x_{naive} \rangle = \\ &= \langle Ax_k - Ax_{naive}; Ax_k - Ax_{naive} \rangle = \\ &= \langle Ax_k - b; Ax_k - b \rangle = \\ &= \|r_k\|^2. \end{aligned}$$

Tedy vidíme, že vektor  $x_k$  je konstruován tak, aby minimalizoval normu rezidua. Ekvivalentně v každé iteraci  $k = 1, 2, \dots$  platí

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

$$\|r_k\| = \min_{x \in \mathcal{K}_k(A^T A, A^T b)} \|b - Ax\|.$$

Prostor  $\mathcal{K}_k(A^T A, A^T b)$  má lepší regularizační vlastnosti než prostor  $\mathcal{K}_k(A, b)$ , protože při násobení maticí  $A^T A$  zhlazujeme konstruovaný bázeový vektor dvakrát za sebou.

Metody CGLS a LSQR jsou sice matematicky ekvivalentní, liší se ovšem v implementaci. Algoritmus metody CGLS uvádíme jako Algoritmus 3. Získáme jej dosazením  $A^T A$  a  $A^T b$  do metody sdrůžených gradientů, viz. [5, Kap. 7.4.1], a následnou úpravou. Algoritmus je založen na tříčlenných rekurencích.

---

### Algoritmus 3 CGLS

---

**Vstup:**  $A, b$

**Výstup:**  $x_k$

$$x_0 = 0$$

$$r_0 = b$$

$$p_0 = A^T r_0$$

**for**  $j = 1, \dots, k$  **do**

$$\alpha_j = \|A^T r_{j-1}\|^2 / \|Ap_{j-1}\|^2$$

$$x_j = x_{j-1} + \alpha_j p_{j-1}$$

$$r_j = r_{j-1} - \alpha_j Ap_{j-1}$$

$$\beta_j = \|A^T r_j\|^2 / \|A^T r_{j-1}\|^2$$

$$p_j = A^T r_j + \beta_j p_{j-1}$$

**end for**

---

Metoda LSQR, viz. [6], je na rozdíl od CGLS založena na *Golub-Kahanově iterační bidiagonalizaci*, která slouží k výpočtu ortonormální báze Krylovova prostoru  $\mathcal{K}_k(A^T A, A^T b)$ . Její algoritmus uvedeme jako Algoritmus 4. Poznamenejme, že pro případ  $\text{rank}(A) = n$  lze Algoritmus 4 odvodit z Algoritmu 2 pro volbu  $M \equiv A^T A$  a  $v \equiv A^T b$ , viz. [2, Kap. 7.3]. Výpočet je tedy založen opět na krátkých rekurencích. Oproti Algoritmu 3 však pracuje jinak.

---

**Algoritmus 4** Golub-Kahanova iterační bidiagonalizace

---

**Vstup:**  $A, b$

**Výstup:**  $W_k, S_{k+1}$

$$w_0 = 0$$

$$\delta_1 = \|b\|$$

$$s_1 = b/\delta_1$$

**for**  $j = 1, \dots, k$  **do**

$$p = A^T s_j - \delta_j w_{j-1}$$

$$\gamma_j = \|p\|$$

$$w_j = p/\gamma_j$$

$$q = Aw_j - \gamma_j s_j$$

$$\delta_{j+1} = \|q\|$$

$$s_{j+1} = q/\delta_{j+1}$$

**end for**

---

Výstupem Algoritmu 4 jsou matice  $W_k = (w_1, \dots, w_k) \in \mathbb{R}^{n \times k}$  a  $S_{k+1} = (s_1, \dots, s_{k+1}) \in \mathbb{R}^{m \times (k+1)}$  s ortonormálními sloupci. Platí, že  $S_{k+1}$  má ve sloupcích bázi prostoru  $\mathcal{K}_{k+1}(AA^T, b)$  a  $W_k$  bázi prostoru  $\mathcal{K}_k(A^T A, A^T b)$ , pro důkaz viz. [2, s. 183-188]. Označme matici normalizačních koeficientů

$$L_k = \begin{pmatrix} \gamma_1 & & & & \\ \delta_2 & \gamma_2 & & & \\ & \ddots & \ddots & & \\ & & & \delta_k & \gamma_k \\ & & & & \delta_{k+1} \end{pmatrix} \in \mathbb{R}^{(k+1) \times k}.$$

Nechť jsou všechny prvky na diagonále a poddiagonále  $L_k$  nenulové. Pak můžeme popsat Algoritmus 4 maticově jako

$$A^T S_{k+1} = W_k L_k^T + \gamma_{k+1} w_{k+1} e_{k+1}^T, \quad (3.5)$$

$$AW_k = S_{k+1} L_k. \quad (3.6)$$

Metoda LSQR hledá (stejně jako CGLS) aproximaci  $x_k \in \mathcal{K}_k(A^T A, A^T b)$ . Tedy můžeme psát  $x_k = W_k y_k$ , kde  $y_k$  je vektor koeficientů  $x_k$  vzhledem k  $W_k$ . Protože  $s_1 = S_{k+1} e_1 = b/\delta_1$ , dostáváme

$$\|r_k\| = \|b - Ax_k\| = \|\delta_1 s_1 - AW_k y_k\| = \|S_{k+1}(\delta_1 e_1 - L_k y_k)\| = \|\delta_1 e_1 - L_k y_k\|,$$

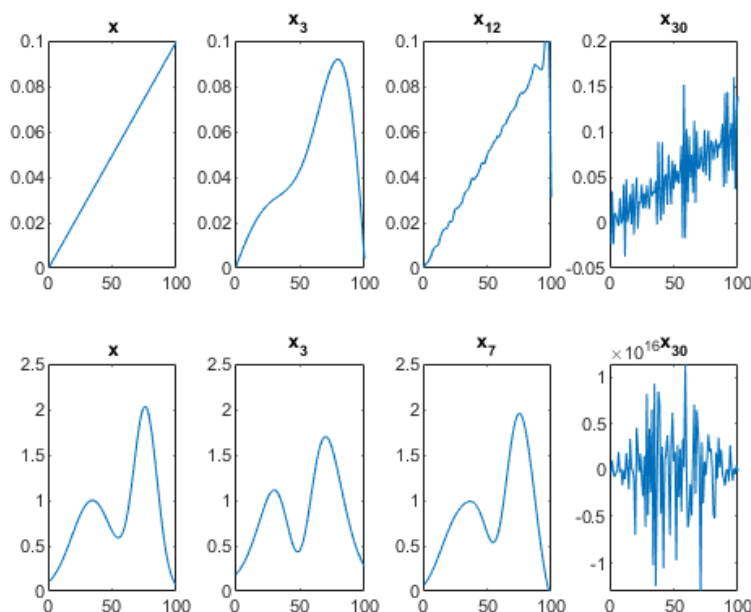
kde jsme využili vztah (3.5) a dále fakt, že  $S_{k+1}$  má ortonormální sloupce a nemění velikost euklidovské normy. Nyní chceme minimalizovat  $\|r_k\|$  vhodnou volbou  $x_k$ . Vektor  $y_k$  musí tím pádem být řešením ve smyslu nejmenších čtverců problému

$$L_k y_k \approx \delta_1 e_1. \quad (3.7)$$

Jinými slovy (3.7) je projekcí problému (1.2) na prostor  $\mathcal{K}_k(A^T A, A^T b)$  a  $x_k = W_k y_k$  představuje aproximaci řešení (1.2) vzhledem k tomuto Krylovovu prostoru.

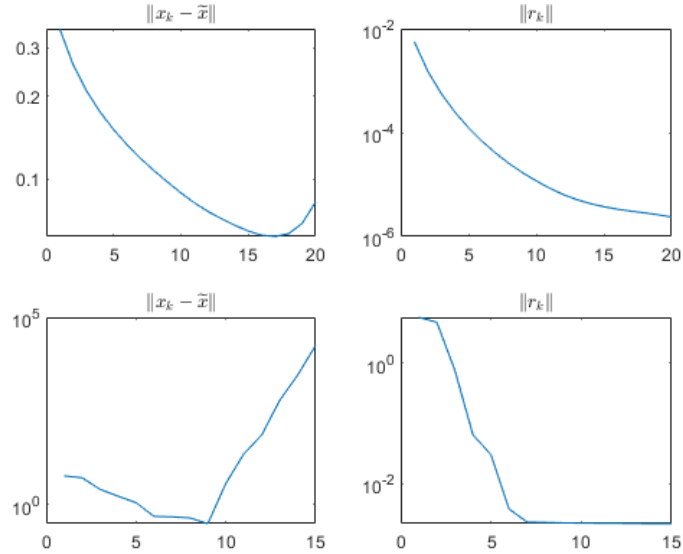
### 3.3 Numerické srovnání

Poslední kapitola je věnována numerickému chování metod popsaných v Kapitole 3.2. Všechny metody ilustrujeme na Příkladech 1 a 2. Používáme zde dva příklady, jelikož reprezentují různé typy ill-posed úloh, viz. Obrázek 1.1. Singulární čísla matice  $A$  pro Problém 2 klesají velmi rychle na strojovou přesnost a stačí nám pouze několik málo iterací k získání vhodné aproximace, zároveň je množství chyby zahrnuté do aproximace velmi malé. V Příkladu 1 je situace trochu složitější. Protože singulární čísla klesají poměrně plynule, je velmi obtížné nalézt vhodnou míru aproximace. Z těchto důvodů se chování každé metody na těchto příkladech liší.



Obrázek 3.5: Porovnání přesných řešení (vlevo) a aproximací pomocí metody RRGMRES pro úlohy z Příkladů 1 a 2 pro různé varianty parametru  $k$ . První řádek přísluší problému z Příkladu 1, druhý řádek problému z Příkladu 2. Třetí obrázky na obou řádcích představují vhodnou aproximaci řešení. Vidíme, že pro Příklad 1 je třeba více iterací k získání dobré aproximace, což je dáno tím, že singulární čísla matice v Příkladu 1 klesají pomalu, na rozdíl od singulárních čísel v Příkladu 2, viz. Obrázek 1.1. V Příkladu 2 dochází k rychlejší projekci šumu do  $x_k$  s rostoucím  $k$ .

Použití metody RRGMRES ilustrujeme na Obrázku 3.5. Podobně jako při použití metody TSVD zde vidíme, že pro velká  $k$  je aproximace  $x_k$  blízká naivnímu řešení a pro velmi nízké hodnoty  $k$  je aproximace naopak příliš hladká. Číslo  $k$  zde hraje roli regularizačního parametru, který kontroluje množství regularizace v aproximaci řešení. Iterační Krylovovské metody tedy vykazují tak zvanou *semikonvergenci*, viz. [1, Kap. 6.3]. Jak ukazuje Obrázek 3.6, skutečná chyba řešení  $\|x_k - \tilde{x}\|$  se nejprve snižuje s rostoucím  $k$ . Pokud však počítáme déle, začne být projektovaný problém na Krylovově prostoru zatížen chybami (viz. Kap. 3.1) a chyba opět roste. Současně  $\|r_k\|$  stále klesá, neboť  $x_k$  konverguje k  $x_{naive}$ . Vi-

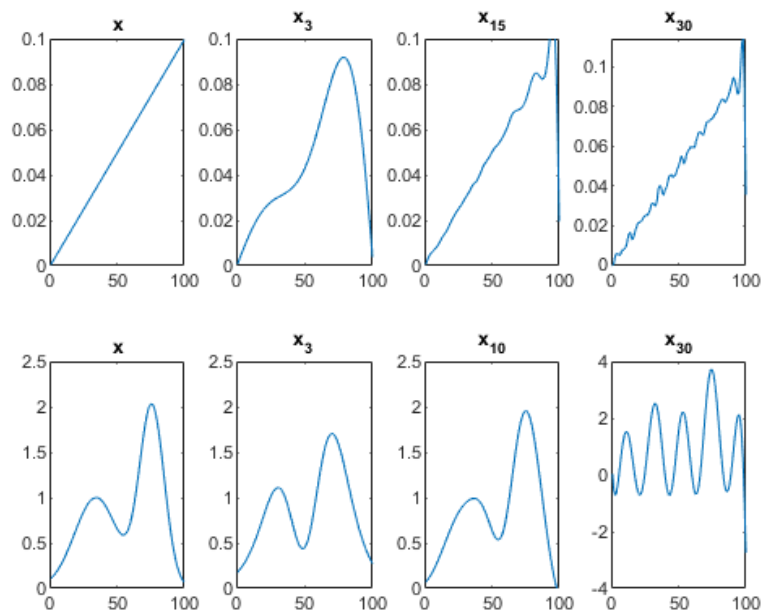


Obrázek 3.6: Srovnání normy chyby  $\|x_k - \tilde{x}\|$  a normy rezidua  $\|r_k\|$  pro  $k = 1, 2, \dots$ . Nahoře pro Příklad 1, dole pro Příklad 2. Vidíme, že norma rezidua stále klesá. Naproti tomu norma chyby klesá pouze pro několik prvních iterací, potom začne růst. Tomuto chování říkáme *semikonvergence*. Také zde vidíme, že pro Příklad 2 je třeba méně iterací k nalezení vhodné aproximace řešení, ale také se v dalších iteracích spočtená aproximace rychleji vzdaluje od přesného řešení  $\tilde{x}$  než je tomu v Příkladě 1.

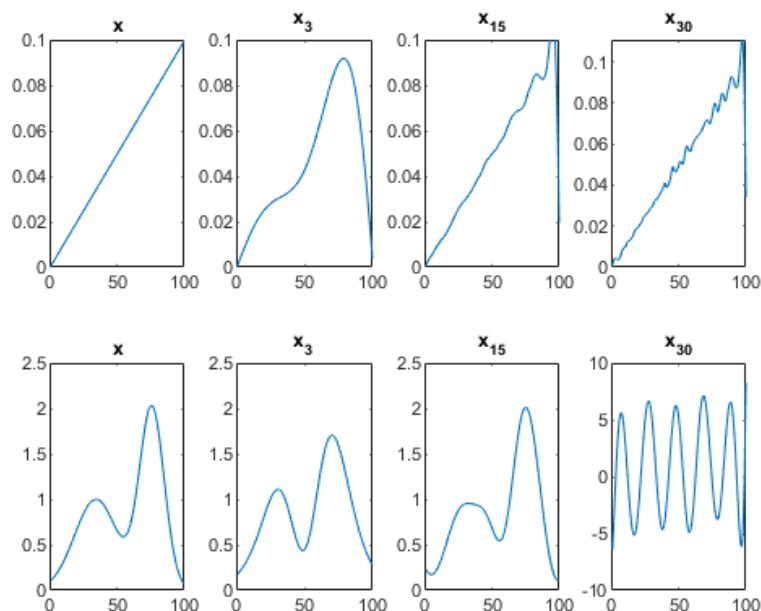
díme, že velikost normy rezidua není spolehlivý indikátor přesnosti aproximace  $x_k$ . Z tohoto důvodu se při řešení ill-posed úloh musí používat speciální zastavovací kritéria, jako například *L-křivka* a podobně, viz. [1, Kap. 5].

Na Obrázku 3.7 vidíme podobné výsledky jako na Obrázku 3.5 pro metodu CGLS. I zde vidíme, že pro velká  $k$  jsou aproximace velmi rozoscilované. Bylo zde třeba volit vyšší  $k$  než při použití metody RRGMRRES, abychom získali dobrou aproximaci, kterou vidíme na Obrázku 3.7 ve třetím sloupci. Důvodem je zřejmě fakt, že v kroku  $k$  je Krylovův prostor v CGLS generován hladšími vektory než Krylovův prostor v RRGMRRES. Mějme na paměti, že metoda CGLS (i LSQR) má širší využití než RRGMRRES, neboť lze použít i na obdélníkovou matici  $A$ . Výpočetní náklady na jednu iteraci jsou v RRGMRRES navíc vyšší než v CGLS (a LSQR) a tento rozdíl s  $k$  roste.

Poslední z testovaných je metoda LSQR. Její výsledky ilustruje Obrázek 3.8, v jehož třetím sloupci opět vidíme nejlepší aproximace pro Příklady 1 a 2. Celkově dostáváme pro metodu LSQR podobné výsledky jako pro metodu CGLS, což lze očekávat z jejich matematické ekvivalence.



Obrázek 3.7: Porovnání přesných řešení (vlevo) a aproximací pomocí metody CGLS pro úlohy z Příkladů 1 a 2 pro různé varianty parametru  $k$ . První řádek přísluší problému z Příkladu 1, druhý řádek problému z Příkladu 2. Třetí sloupec představuje v obou případech vhodnou aproximaci. Opět zde vidíme podobný efekt jako na Obrázku 3.5, avšak potřebný počet iterací je mírně vyšší.



Obrázek 3.8: Porovnání přesných řešení (vlevo) a aproximací pomocí metody LSQR pro úlohy z Příkladů 1 a 2 pro různé varianty parametru  $k$ . První řádek přísluší problému z Příkladu 1, druhý řádek problému z Příkladu 2. Třetí sloupec představuje v obou případech vhodnou aproximaci. Metoda se chová analogicky jako metoda CGLS na Obrázku 3.7.

# Závěr

Cílem této práce bylo studovat a popsat iterační regularizační metody pro řešení diskrétních ill-posed úloh založené na jejich projekci na sekvenci Krylovových prostorů. Nejprve jsme zavedli základní pojmy s tímto tématem spojené. Popsali jsme důležité vlastnosti ill-posed problémů, jako jsou špatná podmíněnost matice  $A$ , schopnost zhlazovat, skutečnost, že její singulární čísla klesají v průměru k nule bez větší mezery, zvyšující se frekvenci v singulárních vektorech a rozdíly v chování mezi přesnou pravou stranou  $b$  a vektorem chyby  $e$ . Tyto vlastnosti jsme ilustrovali na příkladech. Díky analýze metody nejmenších čtverců, která vedla na neuspokojivé výsledky, jsme došli k nutnosti používat pro řešení ill-posed problémů regularizační metody. Představili jsme základní regularizační metodu TSVD a shrnuli důležitost vhodné volby regularizačního parametru.

Dále jsme se zaměřili na Krylovovy prostory. Zabývali jsme se souvisejícími pojmy a věnovali se problematice hledání vhodné báze Krylovova prostoru pro případ obecné i symetrické matice  $A$ . Dále jsme se věnovali regularizačním vlastnostem projekcí na Krylovovy prostory. Ukázali jsme, že díky zhlazující vlastnosti matice  $A$  se Krylovova báze (pro malý počet iterací) skládá z hladkých vektorů, z čehož vyplývají požadované vlastnosti projekcí na generovaný prostor. Arnoldiho báze pak díky ortonormalitě umožňuje aproximovat na Krylovově prostoru i méně hladké složky řešení. Regularizační efekt pak jasně dokládají Picardovy grafy pro původní matici  $A$  a její projekci  $H_k$ .

Na závěr jsme uvedli základní Krylovovské regularizační metody RRGMRRES, CGLS a LSQR. Zabývali jsme se otázkou, pro jaké druhy problémů jsou tyto metody vhodné a ilustrovali jsme jejich chování na dvou příkladech s rozdílnými vlastnostmi. Pozorovali jsme tak zvanou semikonvergenci, tedy že pro velké hodnoty parametru  $k$  (počet iterací) se řešení spočtená zkoumanými metodami blížila k naivnímu řešení. Konvergence metody RRGMRRES byla oproti dalším dvěma metodám o něco rychlejší, na druhou stranu má tato metoda užší využití než metody CGLS a LSQR. Při srovnání těchto dvou metod jsme pozorovali pouze malé rozdíly, což je zřejmě způsobeno tím, že si jsou matematicky ekvivalentní.

Obecně je třeba dodat, že je velmi těžké nalézt vhodnou hodnotu regularizačního parametru, který je u Krylovovských metod reprezentován počtem iterací  $k$ . Klasická zastavovací kritéria používaná při řešení soustav lineárních rovnic zde z důvodu semikonvergence nelze použít. Vzhledem k rozsahu práce jsme se tímto tématem hlouběji nezabývali. Různé metody lze nalézt v [1, Kap. 5].

# Seznam použité literatury

- [1] P. Ch. Hansen. *Discrete inverse problems: Insight and algorithms*, volume 7 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2010.
- [2] J. D. Tebbens, I. Hnětynková, M. Plešinger, Z. Strakoš, and P. Tichý. *Analýza metod pro maticové výpočty: Základní metody*. Matfyzpress, Praha, 2012.
- [3] P. Ch. Hansen. *Rank-deficient and discrete ill-posed problems*. SIAM Monographs on Mathematical Modeling and Computation. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1998.
- [4] D. Calvetti, B. Lewis, and L. Reichel. GMRES-type methods for inconsistent systems. *Linear Algebra and its Applications*, 316(1-3):157–169, 2000.
- [5] A. Björck. *Numerical methods for least squares problems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1996.
- [6] Ch. C. Paige and M. A Saunders. Algorithm 583. LSQR: Sparse linear equations and least squares problems. *ACM Transactions on Mathematical Software (TOMS)*, 8(2):195–209, 1982.
- [7] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, second edition, 2003.
- [8] P. Ch. Hansen. Regtools, 2002. [Regularizační toolbox pro Matlab].
- [9] H. A. Van der Vorst. *Iterative Krylov methods for large linear systems*, volume 13 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2003.
- [10] V. Hájková, M. Johanis, O. John, O. F. K. Kalenda, and M. Zelený. *Matematika*. Matfyzpress, Praha, 2012.
- [11] Wikipedie. Vlastní vektory a vlastní čísla — wikipedie: Otevřená encyklopedie, 2018. [Online; navštíveno 19. 12. 2018, [https://cs.wikipedia.org/w/index.php?title=Vlastní\\_vektory\\_a\\_vlastní\\_čísla&oldid=16129222](https://cs.wikipedia.org/w/index.php?title=Vlastní_vektory_a_vlastní_čísla&oldid=16129222)].
- [12] G. H. Golub and Ch. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, third edition, 1996.
- [13] B. N. Parlett. *The symmetric eigenvalue problem*, volume 20 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1998.
- [14] J. Liesen and Z. Strakoš. *Krylov subspace methods: Principles and analysis*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, 2013.

- [15] W. E. Arnoldi. The principle of minimized iteration in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9:17–29, 1951.
- [16] L. Giraud, J. Langou, M. Rozložník, and J. van den Eshof. Rounding error analysis of the classical Gram-Schmidt orthogonalization process. *Numerische Mathematik*, 101(1):87–100, 2005.
- [17] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Natl. Bur. Stand. B*, 45:255–282, 1950.
- [18] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.
- [19] A. Neuman, L. Reichel, and H. Sadok. Algorithms for range restricted iterative methods for linear discrete ill-posed problems. *Numerical Algorithms*, 59(2):325–331, 2012.