

Bakalářská práce

Konvergence Krylovovských metod pro maticové rovnice z pohledu hodnosti

Studijní program:

B0541A170015 Matematika

Autor práce:

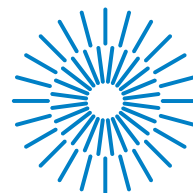
Jana Lungová

Vedoucí práce:

doc. Ing. Martin Plešinger, Ph.D.

Katedra matematiky a didaktiky matematiky

Liberec 2023



Zadání bakalářské práce

Konvergence Krylovovských metod pro maticové rovnice z pohledu hodnotnosti

<i>Jméno a příjmení:</i>	Jana Lungová
<i>Osobní číslo:</i>	P20000725
<i>Studijní program:</i>	B0541A170015 Matematika
<i>Zadávající katedra:</i>	Katedra matematiky a didaktiky matematiky
<i>Akademický rok:</i>	2021/2022

Zásady pro vypracování:

Abstrakt: Mnoho reálných úloh např. v teorii řízení vede na řešení lineárních maticových rovnic, tj. rovnic, ve kterých jsou koeficienty i neznámé matice. Příkladem může být tzv. Ljapunova rovnice $AXM^T + MXA^T = -BB^T$, kde A , M a X jsou čtvercové matice řádu n .

Tyto rovnice mohou vznikat například při diskretizaci parciálních diferenciálních rovnic, popisujících řízený jev, metodou konečných prvků. V takovém případě může být úloha značně velká, ale matice A a M jsou tzv. řídké díky čemuž s nimi lze pracovat. To však neplatí pro řešení X , které bude obecně husté. Pokud má však matice B nízkou hodnotnost, lze i řešení X aproximovat maticí nízké hodnotnosti. Při řešení rovnice pak lze využít metod tzv. maticové *low-rank* aritmetiky, která využívá vlastností matic nízké hodnotnosti.

Aby však bylo řešení výpočetně zvládnutelné, nesmí v průběhu výpočtu dojít k nekontrolovanému nárůstu hodnotnosti matic, se kterými se pracuje. Cílem této práce je zkoumat vývoj (numerické) hodnotnosti jednotlivých matic v závislosti na iteraci pro vybrané iterační metody. Požadavky: Základní znalosti z lineární algebry, základní znalost anglického jazyka a práce s MATLABem. Práce bude psaná v LaTeXu.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování práce:

tištěná/elektronická

Jazyk práce:

Čeština

Seznam odborné literatury:

A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM, Philadelphia PA, 2005.

<https://epubs.siam.org/doi/book/10.1137/1.9780898718713>

J. Liesen, Z. Strakoš, *Krylov Subspace Methods: Principles and Analysis*, Oxford Publications, Oxford, 2013.

<https://oxford.universitypressscholarship.com/view/10.1093/acprof:oso/9780199655410.001.0001/acprof-9780199655410>

D. Kressner, M. Plešinger, C. Tobler, A preconditioned low-rank CG method for parameter-dependent Lyapunov matrix equations, *Numerical Linear Algebra and its Applications* 21(5), (2014), pp. 666–684.

<https://onlinelibrary.wiley.com/doi/10.1002/nla.1919>

V. Simoncini, Y. Hao, Analysis of the truncated conjugate gradient method for linear matrix equations. Preprint HAL, ID hal-03579267f, 2022.

<https://hal.archives-ouvertes.fr/hal-03579267/document>

Vedoucí práce:

doc. Ing. Martin Plešinger, Ph.D.

Katedra matematiky a didaktiky matematiky

Datum zadání práce:

1. září 2022

Předpokládaný termín odevzdání: 28. dubna 2023

L.S.

doc. Ing. Martin Plešinger, Ph.D.
vedoucí bakalářské práce

doc. RNDr. Jana Příhonská, Ph.D.
vedoucí katedry

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracovala samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědoma toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědoma povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědoma následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

Konvergence krylovovských metod pro maticové rovnice z pohledu hodnosti

Abstrakt

Bakalářská práce je zaměřena na řešení maticových rovnic z teorie řízení dynamických systémů, konkrétně se soustředí na tzv. Ljapunovskou rovnici ve tvaru $AX + XA^T = F$, kde A , X a F jsou čtvercové matice řádu n . Úlohy v praxi jsou velkých rozměrů ale matice A je zpravidla tzv. *řídka*. Řešení X tuto vlastnost ale obecně nemá. Může se tedy stát, že s ním nebudeme schopni pracovat — nemusí být uložitelné v počítači. Pokud má pravá strana F *nízkou hodnost*, bude však řešení X možné aproximovat maticí nízké hodnosti — budeme moci pracovat s tzv. *low-rank* aritmetikou.

Úlohu budeme řešit iterační metodou (sdruženými gradienty). Práci s velkými maticemi budeme pouze simulovat — budeme používat matice malé a bude nás zajímat numerická hodnost jednotlivých matic v průběhu iterací. Hlavním cílem je zjistit, kdy hodnosti matic zůstanou nízké po celou dobu a kdy hodnost exponenciálně narůstá, případně zda lze tomuto nárůstu nějakým způsobem zabránit.

Klíčová slova: Ljapunovská rovnice; metoda sdružených gradientů (CG); low-rank aritmetika

Convergence of matrix-oriented Krylov-subspace methods in terms of rank

Abstract

The bachelor thesis is focused on solving matrix equations from control theory of dynamic systems and is specifically concentrating on the so-called Lyapunov equation in the form $AX + XA^T = F$, where A , X and F are square matrices of order n . Realistic problems are of large dimensions but matrix A is usually so-called *sparse*. The solution X does not generally have this property. So it can happen, that we may not be able to work with it — it may not even be storable in computer. However, if the matrix F is of *low-rank*, then the solution X can be approximated by a low-rank matrix — it is possible to use so-called *low-rank* arithmetic.

For solving such problem we use an iterative method (the method of conjugate gradients). We will only simulate the computation with large matrices — we use small matrices and study their numerical ranks during the iterative process. The main goal is to find out when the rank of all the matrices remains low and when it starts to grow exponentially, eventually whether this growth can be prevented in some way.

Keywords: Lyapunov equation; conjugate gradient method (CG); low-rank arithmetic

Poděkování

Ráda bych poděkovala, svému vedoucímu bakalářské práce, panu Martinu Plešingerovi, za cenné rady, věcné připomínky a vstřícnost při konzultacích a vypracování bakalářské práce.

Obsah

Abstrakt	5
Abstract	6
Seznam obrázků	10
Seznam algoritmů	11
Použité značení a zkratky	12
Úvod	13
1 Způsoby uložení (velkých) matic v počítači	14
1.1 Řídké matice	14
1.2 Hierarchické matice	15
1.3 Matice nízké hodnoti	18
2 Maticové rovnice	20
2.1 Formulace problému	21
2.2 Základní vlastnosti matice řešení ljuvenovské rovnice	21
2.2.1 Symetrie řešení	22
2.2.2 Aproximovatelnost řešení maticí nízké hodnoti	23
2.3 Základní vlastnosti matice soustavy ljuvenovské rovnice	23
2.3.1 Motivace	23
2.3.2 Kroneckerův součin	24
2.3.3 Vektorizace matic	26
2.3.4 Regularita matice soustavy ljuvenovské rovnice	28
3 Metoda sdružených gradientů	31
3.1 Maticová formulace sdružených gradientů	33
3.2 Vlastnosti maticově formulovaných sdružených gradientů	34
3.2.1 Symetrie algoritmu MCG	34
3.2.2 Low-rankness algoritmu MCG	35

4	Numerické experimenty	38
4.1	Základní experiment – 2D Laplaceova matice	38
4.1.1	Ověření předpokladů	39
4.1.2	Nežádoucí nárůst hodnoty	39
4.1.3	Vliv zaokrouhlovacích chyb	41
4.2	Různé pravé strany	46
4.2.1	Pravá strana z jedniček, nul a minus jedniček	46
4.2.2	Pravá strana z náhodných čísel	48
4.2.3	Pravá strana zkonstruovaná tak, aby nenarostla hodnota	51
4.3	Použití předpodmiňovače	52
4.4	Jiné zobrazení \mathcal{L}	55
	Závěr	57
	Reference	58

Seznam obrázků

1.1	Příklad řídké matice.	16
1.2	Příklad hierarchicky uložené matice.	17
4.1	Vlastní čísla matice X	39
4.2	Největších 20 vlastních čísel matice X	40
4.3	Průběh 20 největších vlastních čísel matic X_k	40
4.4	Absolutní hodnoty vlastních čísel matic X_k	41
4.5	Rezidua a směrové vektory v průběhu MCG.	42
4.6	Rezidua bez a s reortogonalizací.	44
4.7	Odchylka od ortonormality.	45
4.8	Vlastní čísla matic X_k bez absolutních hodnot.	47
4.9	Vlastní čísla matic X_k pro pravou stranu s kladným i záporným vlastním číslem.	47
4.10	Průběh 40 největších vlastních čísel matic X_k pro pravou stranu s kladným i záporným vlastním číslem.	48
4.11	Průběh hodnotí matic X_k pro pravé strany hodnoti jedna.	49
4.12	Vlastní čísla matic X_k pro pravou stranu hodnoti jedna.	49
4.13	Průběh hodnotí matic X_k pro pravé strany hodnoti jedna II.	50
4.14	Vlastní čísla matic X_k pro pravou stranu hodnoti jedna II.	50
4.15	Průběh vlastních čísel matic X_k – příklad bez nárůstu hodnoti.	51
4.16	Průběh vlastních čísel matic X_k s použitým předpodmiňovačem.	53
4.17	Průběh předpodmiňovaných reziduí.	53
4.18	Průběh R_k , P_k a W_k s použitým předpodmiňovačem.	54
4.19	Vlastní čísla matice Strakoš(100).	55
4.20	Průběh vlastních čísel matic X_k pro $A = \text{Strakoš}(100)$	56

Seznam algoritmů

1	CG (sdružené gradienty) pro řešení soustavy $Lx = f$	31
2	MCG pro řešení l'apunovské rovnice $AX + XA^T = F$	33
3	MCG s reortogonalizací (simulace přesné aritmetiky).	43
4	MCG s předpodmíněním.	52

Použité značení a zkratky

Značení	Význam
$A, A = (a_{ij}) \in \mathbb{R}^{m \times n}$	reálná matice s rozměry m krát n , s prvky a_{ij}
$\text{rank}(A)$	hodnota matice definovaná jako počet lineárně nezávislých řádků, resp. sloupců matice A
A^T	transpozice matice A
A^{-1}	inverze čtvercové regulární matice A
I, I_n	jednotková matice, resp. jednotková matice řádu n
e_ℓ	ℓ -tý eukleidovský vektor vhodné délky
$0, 0_{m,n}$	nulová matice, resp. nulová matice velikosti m krát n
$A \otimes B$	Kroneckerův součin matic A a B
$\text{vec}(A)$	vektORIZACE matice A
$\ v\ $	eukleidovská norma vektoru $\ v\ = (\sum_{i=1}^n v_i^2)^{\frac{1}{2}}$
$\ A\ $	Frobeniova norma matice $\ A\ = (\sum_{i=1}^m \sum_{j=1}^n a_{ij}^2)^{\frac{1}{2}}$
$\langle u, v \rangle$	standardní skalární součin vektorů $\langle u, v \rangle = v^T u$
$\langle A, B \rangle$	Frobeniův skalární součin matic $\langle A, B \rangle = \text{vec}(B)^T \text{vec}(A)$
$\det(A)$	determinant čtvercové matice A
$\text{sp}(A)$	spektrum čtvercové matice A
λ, λ_n	vlastní číslo, resp. n -té vlastní číslo čtvercové matice

Použité zkratky a akronymy

Zkratka	Význam
SPD	symetrická a pozitivně definitní matice
CG	sdužené gradienty
MCG	maticová varianta sdužených gradientů

Úvod

V této práci se budeme zabývat řešením specifických soustav lineárních rovnic. Tedy úloh, které v principu lze zapsat ve tvaru

$$Lx = f.$$

Naše úlohy ovšem budou dále strukturované, viz kapitolu 2. Typicky ale bude *matic* soustavy L reálná, čtvercová a regulární, a navíc ještě symetrická a pozitivně definitní,

$$L \in \mathbb{R}^{n \times n}, \quad \det(L) \neq 0, \quad L^T = L, \quad \lambda_\ell(L) > 0, \quad \ell = 1, 2, 3, \dots, n,$$

kde $\lambda_\ell(L)$ značí ℓ -té největší vlastní číslo matice L včetně násobností.

Konkrétně budeme řešit tzv. Ljapunovské rovnice, které se často vyskytují např. v úlohách z teorie řízení dynamických systémů, viz např. [2] a [17]. Cílem je zjistit, zda jsme schopni najít řešení soustavy i pokud je úloha tak velkého řádu, že se vektor (resp. matice) řešení jako takový nevejde do počítače. Matice soustavy je přitom typicky řídká a vektor (resp. matice) pravé strany je nízké hodnoti.

V kapitole 1 shrneme jakým způsobem lze uložit velké matice vhodných vlastností do počítače, v kapitole 2 popíšeme několik druhů maticových rovnic, blíže se zaměříme na tzv. Ljapunovskou rovnici. V kapitole 3 rozebereme metodu sdružených gradientů, její vlastnosti a implementaci. Abychom mohli s velkými maticemi pracovat, budeme se soustředit na low-rank aritmetiku. Kapitola 4 je věnovaná několika numerickým experimentům a stručné analýze jejich výsledků.

1 Způsoby uložení (velkých) matic v počítači

Pokud pracujeme s obecnou (reálnou, čtvercovou) maticí A a chceme ji uložit do počítače, potřebujeme uložit všechny její prvky, v našem případě n^2 prvků. Existují ale druhy matic, které lze ukládat efektivněji. Příkladem takových matic jsou zejména tzv. *řídke matice*, ale také tzv. *hierarchické matice*, resp. *matice nízké hodnoty*.

1.1 Řídké matice

Matici nazýváme řídkou, pokud obsahuje dostatečně mnoho nulových a tedy dostatečně málo nenulových prvků. Počet nenulových prvků matice A budeme označovat $\text{nnz}(A)$, z anglického *number of non-zeros*. Pro počet nenulových prvků řídké matice neexistuje striktní kritérium. Obecně ale chceme aby řídká matice velikosti $n \times n$ měla řádově n nenulových prvků, resp. aby

$$\text{nnz}(A) \ll n^2.$$

Řídká matice pro nás bude hrát roli *matice soustavy*, která je regulární; navíc tedy bude platit

$$n \leq \text{nnz}(A) \ll n^2.$$

Oblíbeným kritériem řídkosti matice také je, zda umíme velkého množství nulových prvků efektivně využít. Poznamenejme, že opakem řídkých matic jsou matice *husté*, jsou to takové které nejsou řídké.

Matice o nichž jednoznačně řekneme, že jsou řídké, jsou matice *diagonální*, typicky matice *jednotková*,

$$I_n = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

K uložení (obecné) diagonální matice řádu n zřejmě stačí uložit právě n hodnot.

Dalším typickým příkladem řídké matice je (symetrická) *třídiagonální* matice,

$$T_n = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & 0 & \cdots & 0 \\ \beta_1 & \alpha_2 & \beta_2 & 0 & \cdots & 0 \\ 0 & \beta_2 & \alpha_3 & \beta_3 & & \vdots \\ 0 & 0 & \beta_3 & \alpha_4 & \ddots & 0 \\ \vdots & \vdots & & \ddots & \ddots & \beta_{n-1} \\ 0 & 0 & \cdots & 0 & \beta_{n-1} & \alpha_n \end{bmatrix}. \quad (1.1)$$

V případě prvků obecné třídiagonální matice je zřejmě nutné uložit právě $3n - 2$ hodnot, pro symetrickou třídiagonální matici pouze $2n - 1$ hodnot.

Poznámka 1. *Pokud budeme pracovat s obecnou řídkou maticí A (např. permutací jednotkové matice), která má $m = \text{nnz}(A)$ nenulových prvků, je v principu potřeba uložit $3m$ hodnot. Pro každý nenulový prvek $a_{i,j}$ potřebujeme řádkový a sloupcový index a prvek samotný. Obecně tedy u řídkých matic stačí uložit množinu uspořádaných trojic*

$$(i, j, a_{i,j}).$$

Pro více informací o řídkých maticích odkazujeme na knihy [16] a [5, zejména kapitoly 10 a 11]. Příklad řídké matice, resp. struktury jejích nenulových prvků vidíme na obrázku 1.1.

1.2 Hierarchické matice

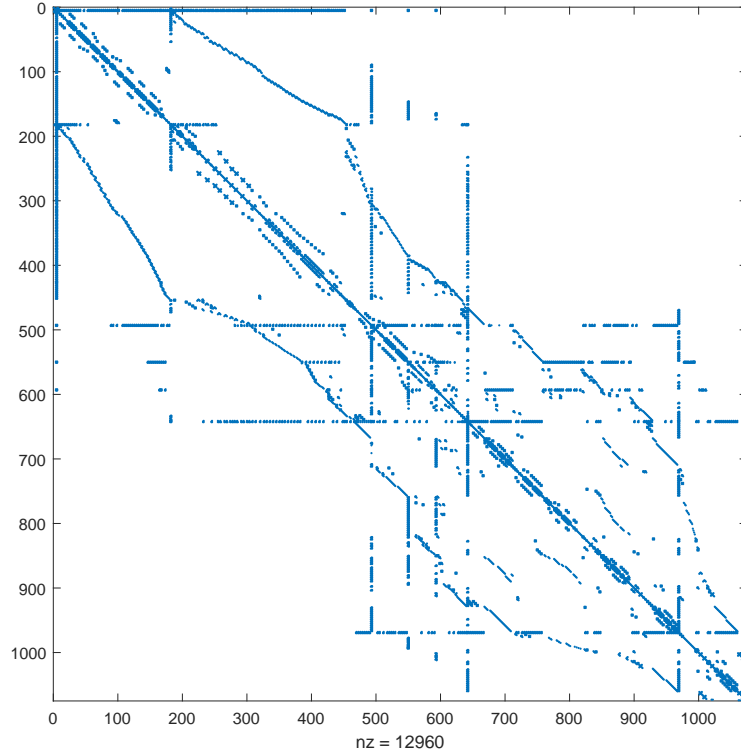
V praxi existují i jiné možnosti jak velké matice efektivně ukládat, např. tzv. Hierarchické matice, viz [1], [7]; my budeme vycházet z prací [9], [10]. Jako motivace nám poslouží naše třídiagonální matice, resp. její inverze. Spočítáme-li ji, zjistíme, že má obecně n^2 nenulových prvků, je tedy hustá; [9]. Pomocí hierarchického přístupu k maticím lze ale tuto hustou matici uložit úsporněji.

Uvažujme inverzi třídiagonální matice (1.1) a její rozdělení do bloků

$$T_n^{-1} = \begin{bmatrix} X_1 & Y^T \\ Y & X_2 \end{bmatrix}. \quad (1.2)$$

Nechť podmatice $X_1 \in \mathbb{R}^{k \times k}$ a $X_2 \in \mathbb{R}^{(n-k) \times (n-k)}$, pro nějaké $k = 1, \dots, n - 1$. Pak se dá ukázat, že matice $Y \in \mathbb{R}^{(n-k) \times k}$ má hodnotu 1; viz [9]. Takovou matici lze napsat jako tzv. *vnější součin* dvou vektorů, resp. dvou matic z nichž první má jeden sloupec a druhá jeden řádek

$$Y = UV^T = uv^T, \quad U = [u] \in \mathbb{R}^{(n-k) \times 1}, \quad V = [v] \in \mathbb{R}^{k \times 1}, \quad u \in \mathbb{R}^{n-k}, \quad v \in \mathbb{R}^k. \quad (1.3)$$



Obrázek 1.1: Matice BCSSTK08 je příkladem řídké matice. Pochází z úlohy BCSSTRUC1 (BCS Structural Engineering Matrices) ze sbírky Harwell-Boeing, viz [20], <https://math.nist.gov/MatrixMarket/data/Harwell-Boeing/bcsstruc1/bcsstruc1.html>. Matice je řádu $n = 1074$, modře jsou vyznačeny její nenulové prvky, bílá místa jsou nulové prvky.

Poznámka 2 (Náznak důkazu rovnosti (1.3)). Matici T_n rozdělíme kompatibilním způsobem jako T_n^{-1}

$$I_n = T_n T_n^{-1} = \begin{bmatrix} K_1 & M^T \\ M & K_2 \end{bmatrix} \begin{bmatrix} X_1 & Y^T \\ Y & X_2 \end{bmatrix},$$

$$\begin{bmatrix} I_k & 0_{k,n-k} \\ 0_{n-k,k} & I_{n-k} \end{bmatrix} = \begin{bmatrix} K_1 X_1 + M^T Y & K_1 Y^T + M^T X_2 \\ M X_1 + K_2 Y & M Y^T + K_2 X_2 \end{bmatrix}$$

vyjádříme podmatici Y za předpokladu, že K_2 je regulární,

$$\begin{aligned} 0_{n-k,k} &= M X_1 + K_2 Y, \\ K_2^{-1} \cdot \begin{cases} K_2 Y = -M X_1, \\ Y = -K_2^{-1} M X_1, \end{cases} \end{aligned}$$

kde

$$M = \begin{bmatrix} 0 & \cdots & 0 & \beta_k \\ 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix} = e_1 \beta_k e_k^T \in \mathbb{R}^{(n-k) \times k}$$

tedy

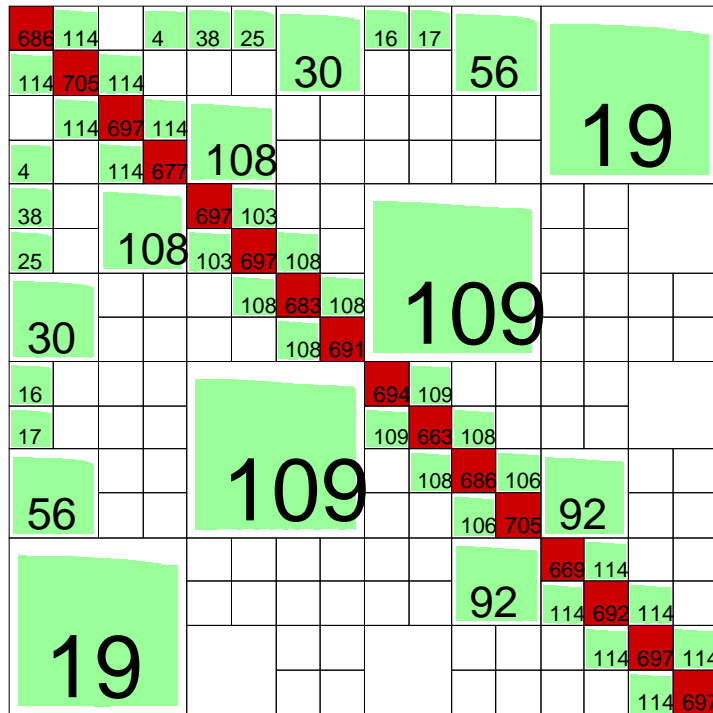
$$Y = -K_2^{-1} e_1 \beta_k e_k^T X_1 = \underbrace{\left(-K_2^{-1} e_1 \right)}_u \underbrace{\left(X_1^T e_k \beta_k \right)^T}_v.$$

V případě kdy blok K_2 není regulární, není regulární ani blok K_1 a situace je rozbrána v [9].

Pokud tedy matici T_n^{-1} uložíme tak jak je, potřebujeme uložit cca $\frac{1}{2}n^2$ čísel (s využitím symetrie). Pokud ji uložíme v naznačeném blokovém dělení, potřebujeme uložit $\frac{1}{2}k^2 + \frac{1}{2}(n-k)^2 + k + (n-k)$ čísel. Pokud je $k \approx \frac{1}{2}n$, tj. dělíme matici přibližně v polovině, spotřebujeme

$$\frac{1}{4}n^2 + n \ll \frac{1}{2}n^2,$$

pro dostatečně velké n . V praxi, a odtud pramení pojem hierarchické matice, můžeme matice X_1 a X_2 matice T_n^{-1} (viz (1.2)) opět rozdělit na 2×2 bloky a celý proces zopakovat tolikrát, kolikrát to dává smysl. Příklad hierarchicky uložené matice vidíme na obrázku 1.2.



Obrázek 1.2: Zde je příklad matice řádu $n = 11036$, která pochází z úlohy [11] a byla použita i v práci [9]. Červené čtvercové bloky na diagonále jsou uloženy přímo (jedná se o husté podmatice), čísla v oblastech označují řád příslušného bloku. Bílé bloky jsou nulové matice. Zelené jsou matice nízké hodnoty s hodnotí r , které jsou uloženy jako r lineárně nezávislých řádků a r lineárně nezávislých sloupců.

1.3 Matice nízké hodnosti

Matice nízké hodnosti (angl. low-rank) jsou takové, které obsahují velké množství lineárně závislých sloupců. Obecně nejsou regulární, proto pro účely našich úloh nebudou hrát roli matice soustavy, ale budou např. reprezentovat data. Příkladem matice nízké hodnosti je nulová matice. Uvažujme tedy

$$M \in \mathbb{R}^{n \times n}, \quad r = \text{rank}(M), \quad r \ll n.$$

Lemma 1. *Matice $M \in \mathbb{R}^{n \times k}$ hodnosti $r = \text{rank}(M)$ lze napsat jako součin*

$$M = UV^T, \quad U \in \mathbb{R}^{n \times r}, \quad V \in \mathbb{R}^{k \times r}.$$

Důkaz. Matice M je hodnosti r , tedy má právě r lineárně nezávislých sloupců. Nechtě to jsou sloupce s indexy j_1, j_2, \dots, j_r . Sestavme z těchto sloupců matici

$$U = [m_{j_1} \quad m_{j_2} \quad \cdots \quad m_{j_r}] \in \mathbb{R}^{n \times r}.$$

Obecný sloupec m_ℓ matice M můžeme napsat jako lineární kombinaci těchto vybraných sloupců, tj.

$$m_\ell = \sum_{s=1}^r m_{j_s} \alpha_{s,\ell} = U v_\ell, \quad \text{kde} \quad v_\ell = \begin{bmatrix} \alpha_{1,\ell} \\ \alpha_{2,\ell} \\ \vdots \\ \alpha_{r,\ell} \end{bmatrix} \quad \text{a} \quad \ell = 1, 2, \dots, k.$$

Tedy

$$V^T = [v_1 \quad v_2 \quad \cdots \quad v_k] \in \mathbb{R}^{r \times k},$$

přičemž z maticového násobení je zřejmé, že platí $M = UV^T$. □

Poznámka 3 (k lemmatu 1). *Pokud matice U, V doplníme stejným počtem nulových sloupců, jejich součin to neovlivní. Tedy*

$$M = UV^T = [U, 0] [V, 0]^T.$$

To je pro nás praktické, pokud uvažujeme matici M hodnosti nula, tj. nulovou matici. Dle lemmatu 1 bychom ji měli být schopni napsat jako součin dvou matic s nulovým počtem sloupců. Abychom s takovými objekty nemuseli pracovat, budeme pro matici

$$M = 0_{n,k}, \quad \text{rank}(M) = 0,$$

uvažovat rozklad ve tvaru

$$M = UV^T, \quad \text{kde} \quad U = 0_{n,1}, \quad V = 0_{k,1}.$$

Pro nás bude matice M často symetrická

$$M = M^T, \quad M = UV^T = VU^T.$$

Symetrii proto budeme požadovat i po jejím rozkladu. Ideálně bychom si přáli, aby

$$U = V,$$

to však obecně nelze splnit. Místo toho budeme uvažovat rozklad z následujícího lemmatu.

Lemma 2. Čtvercovou symetrickou maticí $M \in \mathbb{R}^{n \times n}$, $M = M^T$, hodnosti $r = \text{rank}(M)$ lze napsat jako součin

$$M = U\Lambda U^T, \quad U \in \mathbb{R}^{n \times r}, \quad \Lambda \in \mathbb{R}^{r \times r},$$

kde matice Λ je symetrická, tj. $\Lambda = \Lambda^T$.

Důkaz. Toto lemma lze snadno dokázat z vlastností normálních, resp. symetrických matic a z jejich Schurova (případně také Jordanova) rozkladu, viz [2]. Konkrétně lze matice Λ a U volit např. tak, že matice Λ je diagonální a obsahuje nenulová vlastní čísla matice M na diagonále (pozn. že ta jsou vždy reálná) a sloupce matice U jsou navzájem ortonormální vlastní vektory matice M odpovídající těmto vlastním číslům. \square

Poznámka 4 (k lemmatu 2). Analogicky jako v poznámce 3 lze matice U a Λ doplnit stejným počtem nulových sloupců, respektive sloupců a řádků. Výsledný součin matic se tím nezmění

$$M = U\Lambda U^T = [U, 0] \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} [U, 0]^T.$$

Opět je možné tímto způsobem uvažovat pro nulové matice

$$M = 0_{n,k}, \quad \text{rank}(M) = 0,$$

formální rozklad

$$M = U\Lambda U^T, \quad U = 0_{n,1}, \quad \Lambda = 0_{1,1}.$$

K uložení matice M je třeba uložit matice U a Λ , což je nr prvků a r nenulových vlastních čísel původní matice M . Pro

$$r \ll n, \quad nr + r = (n+1)r \ll n^2$$

tedy paměťové nároky výrazně poklesly.

V této kapitole jsme stručně popsali některé případy, jak lze velkou matici uložit úsporně do počítače. V další práci budeme pracovat zejména s řídkými maticemi (v roli matice soustavy, tj. matice zobrazení) a maticemi nízké hodnosti (v roli zobrazovaných objektů).

2 Maticové rovnice

Řešení maticových rovnic je v praxi velice důležité, protože na maticové rovnice vede řada inženýrských úloh. Zde jsou některé často řešené rovnice, pro které $A, B, F, X \in \mathbb{R}^{n \times n}$, kde X je neznámá a F je (často symetrická $F = F^T$) pravá strana.

Úplně nejjednodušší maticová rovnice na kterou se dá nahlížet jako na n soustav lineárních rovnic (a je tedy snadno řešitelná) je

$$AX = F. \quad (2.1)$$

Rovnice která se často objevuje v analýze stability a optimalizaci řízení dynamických systémů a na kterou se budeme v této práci převážně soustředit se nazývá *ljapunovská*. Rovnice bývá ve dvou variantách, první uvedená je tzv. *spojitá* (má původ ve spojitém dynamickém systému)

$$AX + XA^T = F, \quad (2.2)$$

druhá tzv. *diskrétní* (má původ v diskrétním dynamickém systému) varianta ljapunovské rovnice

$$AXA^T + X = F. \quad (2.3)$$

Následující rovnice jsou také ljapunovské, ale na rozdíl od (2.2) a (2.3) jsou tzv. *předpodmíněné*. Opět můžeme uvažovat spojitou

$$AXB^T + BXA^T = F \quad (2.4)$$

a diskrétní variantu této rovnice

$$AXA^T + BXB^T = F. \quad (2.5)$$

Poznamenejme, že předpodmíněnou rovnici (2.4) dostaneme z rovnice (2.2) např. tak, že (2.2) vynásobíme zleva regulární maticí B a zprava její transpozicí B^T

$$B \cdot \backslash \quad AX + XA^T = F \quad / \cdot B^T,$$

získáme

$$\underbrace{BA}_{\tilde{A}} XB^T + BX \underbrace{A^T B^T}_{\tilde{A}^T} = \underbrace{BF B^T}_{\tilde{F}},$$

kde vhodné součiny přejmenujeme, jak je naznačeno. Analogicky lze získat rovnici (2.5) úpravou rovnice (2.3).

Další z klasických rovnic, také z teorie řízení dynamických systémů, je rovnice Sylvesterova

$$AX + XB = F. \quad (2.6)$$

Poslední rovnice kterou zmíníme se nazývá Riccatiho rovnice,

$$A^T X + X A + X B X = F. \quad (2.7)$$

Zde upozorníme, že (na rozdíl od všech předchozích) není lineární (poslední člen na levé straně XBX je kvadratický). Její řešení je tudíž složitější.

V tomto textu budeme dále pracovat především s Ljapunovskou rovnicí (2.2), resp. místy s její předpokládanou variantou (2.4). Okrajově také zmíníme podobnosti s řešením rovnic (2.3) a (2.5).

2.1 Formulace problému

V práci se budeme zabývat situací, kdy chceme vyřešit Ljapunovskou rovnici

$$AX + XA^T = F \quad A, F, X \in \mathbb{R}^{n \times n}. \quad (2.8)$$

Navíc budeme předpokládat že:

- (i) Matice A je řídká (viz sekce 1.1) a navíc symetrická a pozitivně definitní (SPD).
- (ii) Matice F má nízkou hodnotu (viz sekce 1.3) a je symetrická.
- (iii) Budeme uvažovat tak velké n , že n^2 se nevejde do paměti počítače.

Pro upřesnění poslední odrážky: v textu chceme chování rovnic pouze simulovat, budeme proto pracovat s malou hodnotou n . Nicméně naším cílem je analyzovat, zda lze obejít situaci, kdy bychom potřebovali uložit všech n^2 čísel. Otázka tedy zní:

Jsme schopni nalézt řešení X , aniž bychom v procesu výpočtu zkonstruovali hustou matici velikosti $n \times n$ plné hodnoty?

2.2 Základní vlastnosti matice řešení Ljapunovské rovnice

Prvně poznamenejme, že Ljapunovská rovnice (2.8) je lineární, lze ji tedy v principu přepsat jako soustavu n^2 rovnic pro n^2 neznámých, řekněme

$$Lx = f, \quad \text{kde } L \in \mathbb{R}^{n^2 \times n^2}, \quad x, f \in \mathbb{R}^{n^2} \quad (2.9)$$

a vektory x a f obsahují prvky matic X a F . Pokud je matice soustavy L regulární, pak má soustava $Lx = f$ právě jedno řešení pro každou pravou stranu. Poznamenejme, že regularitou matice L se budeme zabývat v následující sekci.

2.2.1 Symetrie řešení

Důležitou vlastností Ljapunovské rovnice je, že její řešení, tedy matice X , je za shora uvedených předpokladů symetrické. Uvažujme transpozici rovnice (2.8), tj.

$$(AX)^T + (XA^T)^T = F^T.$$

Protože transpozice součinu matic je součin transpozic v opačném pořadí a s využitím symetrie pravé strany (předpoklad (ii)) a komutativity sčítání dostáváme postupně

$$\begin{aligned} (AX)^T + (XA^T)^T &= F^T, \\ X^T A^T + AX^T &= F^T, \\ AX^T + X^T A^T &= F. \end{aligned} \quad (2.10)$$

Dostáváme tak stejnou Ljapunovskou rovnici jako (2.8), pouze s transponovaným řešením. Odečtením této nové rovnice (2.10) od rovnice původní (2.8), tj.

$$\begin{aligned} AX + XA^T &= F, \\ - \setminus \quad AX^T + X^T A^T &= F, \end{aligned}$$

dostaneme, po vytknutí A doleva, resp. A^T doprava opět Ljapunovskou rovnici

$$A \underbrace{(X - X^T)}_Y + \underbrace{(X - X^T)}_Y A^T = 0 \quad (2.11)$$

ovšem s nulovou pravou stranou.

Ljapunovskou rovnici (2.11) můžeme opět interpretovat jako soustavu rovnic analogickou k (2.9), tedy jako $Ly = 0$ se stejnou maticí L . Pokud je L regulární, pak má i soustava $Ly = 0$ právě jedno řešení, které ovšem musí být nulové. Tedy

$$\begin{aligned} Y = X - X^T &= 0, \\ X &= X^T. \end{aligned}$$

Jinými slovy, z regularity matice L a symetrie pravé strany F plyne symetrie matice řešení X .

Podobná implikace platí i naopak. Je-li X symetrická matice, pak bude také výraz

$$AX + XA^T$$

symetrický. To snadno ověříme jeho transponováním, zřejmě

$$(AX + XA^T)^T = (AX)^T + (XA^T)^T = X^T A^T + AX^T = AX^T + X^T A^T = AX + XA^T.$$

Neboli:

Za předpokladu regularity matice L pro $AX + XA^T = F$ platí:

$$X = X^T \quad \iff \quad F = F^T.$$

2.2.2 Aproximovatelnost řešení maticí nízké hodnoti

Dále lze ukázat, že symetrickou matici X lze snadno aproximovat symetrickou maticí nízké hodnoti, tedy že existuje \tilde{X} takové, že

$$\|X - \tilde{X}\| \approx 0, \quad \tilde{X} = \tilde{X}^T, \quad \text{rank}(\tilde{X}) \ll n.$$

Aproximaci přitom můžeme uvažovat např. v tzv. *Frobeniově normě*¹. Tvrzení zde nebudeme dokazovat. Důkaz lze nalézt např. v [15].

2.3 Základní vlastnosti matice soustavy Ljapunovské rovnice

V předchozím textu jsme si ukázali že Ljapunovskou rovnici (2.8) můžeme zapsat jako soustavu $Lx = f$ a že pro její řešení je důležitá regularita matice L , otázka je jak matici sestavit. V případě některých jednodušších maticových rovnic lze o regularitě odpovídající matice soustavy rozhodnout snadno, následující příklad poslouží jako motivace.

2.3.1 Motivace

Začneme nejjednodušší maticovou rovnicí (2.1), tedy $AX = F$. Je zřejmé, že existence a jednoznačnost řešení pro každou pravou stranu F přímo souvisí s regularitou matice A . Vynásobením rovnice inverzní maticí A^{-1} zleva získáme řešení

$$X = A^{-1}F.$$

Zdá se tedy, že zde je regularita matice velké soustavy (řádu n^2) ekvivalentní přímo regularitě matice A .

A opravdu, stačí si matice X a F rozdělit na sloupce, tedy

$$X = [x_1, x_2, \dots, x_n], \quad F = [f_1, f_2, \dots, f_n]$$

a maticové násobení AX interpretovat po sloupcích, tedy

$$Ax_\ell = f_\ell, \quad \ell = 1, 2, \dots, n.$$

Abychom získali matici X potřebujeme vyřešit n právě zmíněných soustav (malých, řádu n). Velkou soustavu (řádu n^2) pak lze napsat např. ve tvaru

$$\underbrace{\begin{bmatrix} A & & & \\ & A & & \\ & & \ddots & \\ & & & A \end{bmatrix}}_{\text{diag}(A, A, \dots, A)} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}. \quad (2.12)$$

¹Frobeniovou normou matice $K = (k_{ij}) \in \mathbb{R}^{m \times n}$ rozumíme číslo $\|K\| = (\sum_{i=1}^m \sum_{j=1}^n k_{ij}^2)^{\frac{1}{2}}$.

Zbývá rozmyslet, že velká matice je regulární tehdy a jen tehdy, když je matice A regulární, tj.

$$\text{diag}(A, A, \dots, A) \text{ je regulární} \iff A \text{ je regulární.}$$

To však plyne přímo z blokově-diagonální struktury velké matice, viz např. [3].

Pro sestavení matice soustavy L odpovídající naší l'apunovské rovnici (2.8) budeme potřebovat tento postup trochu zobecnit. Pro tyto potřeby se zavádí tzv. *Kroneckerův součin a vektorizace matice*.

2.3.2 Kroneckerův součin

Kroneckerův maticový součin je binární operace aplikovatelná na dvojici libovolně velkých matic. Uvažujme tedy matici $K \in \mathbb{R}^{m \times n}$ s prvky k_{ij} a matici $P \in \mathbb{R}^{r \times s}$ s prvky p_{uv} . Jejich Kroneckerův součin (v tomto pořadí) je pak definován² jako

$$K \otimes P = \begin{bmatrix} k_{11}P & \cdots & k_{1n}P \\ \vdots & \ddots & \vdots \\ k_{m1}P & \cdots & k_{mn}P \end{bmatrix} \in \mathbb{R}^{(mr) \times (ns)}. \quad (2.13)$$

Protože

$$k_{ij}P = \begin{bmatrix} k_{ij}p_{11} & \cdots & k_{ij}p_{1s} \\ \vdots & \ddots & \vdots \\ k_{ij}p_{r1} & \cdots & k_{ij}p_{rs} \end{bmatrix},$$

pak je výsledná matice

$$K \otimes P = \begin{bmatrix} k_{11}p_{11} & \cdots & k_{11}p_{1s} & & k_{1n}p_{11} & \cdots & k_{1n}p_{1s} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ k_{11}p_{r1} & \cdots & k_{11}p_{rs} & & k_{1n}p_{r1} & \cdots & k_{1n}p_{rs} \\ & & \vdots & \cdots & & & \vdots \\ k_{m1}p_{11} & \cdots & k_{m1}p_{1s} & & k_{mn}p_{11} & \cdots & k_{mn}p_{1s} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ k_{m1}p_{r1} & \cdots & k_{m1}p_{rs} & & k_{mn}p_{r1} & \cdots & k_{mn}p_{rs} \end{bmatrix}. \quad (2.14)$$

Poznamenejme, že Kroneckerův součin obsahuje zřejmě všechny součiny dvojic prvků $k_{ij}p_{uv}$, každý právě jednou. Zřejmě také obecně není komutativní, $K \otimes P \neq P \otimes K$. Platí pro něj však řada důležitých vztahů.

Věta 1. *Nechť K a P jsou libovolné matice, pak platí*

$$K^T \otimes P^T = (K \otimes P)^T.$$

Neboli, pořadí operací Kroneckerova součinu a transpozice je zaměnitelné.

²Poznamenejme, že se lze v literatuře občas (ojediněle) setkat s jinou definicí Kroneckerova součinu, kde hrají obě matice opačné role. Příkladem je kniha Miroslava Fiedlera [3], na kterou zde budeme hojně odkazovat.

Důkaz. Uvažujme konkrétně $K = (k_{ij}) \in \mathbb{R}^{m \times n}$ a $P = (p_{uv}) \in \mathbb{R}^{r \times s}$. Rozepíšeme-li součin na levé straně, získáme $K^T \otimes P^T =$

$$\begin{bmatrix} k_{11}P^T & \cdots & k_{m1}P^T \\ \vdots & \ddots & \vdots \\ k_{1n}P^T & \cdots & k_{mn}P^T \end{bmatrix} = \begin{bmatrix} k_{11}p_{11} & \cdots & k_{11}p_{r1} & & k_{m1}p_{11} & \cdots & k_{m1}p_{r1} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ k_{11}p_{1s} & \cdots & k_{11}p_{sr} & & k_{m1}p_{1s} & \cdots & k_{m1}p_{sr} \\ & & \vdots & \ddots & & & \vdots \\ k_{1n}p_{11} & \cdots & k_{1n}p_{r1} & & k_{mn}p_{11} & \cdots & k_{mn}p_{r1} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ k_{1n}p_{1s} & \cdots & k_{1n}p_{sr} & & k_{mn}p_{1s} & \cdots & k_{mn}p_{sr} \end{bmatrix}.$$

Dostáváme tak zjevně transpozici matice (2.14), tedy matici $(K \otimes P)^T$. \square

Věta 2. *Nechť $K, G, P,$ a H jsou libovolné matice takové, že existují součiny KG a $PH,$ pak platí*

$$(KG) \otimes (PH) = (K \otimes P)(G \otimes H).$$

Neboli, Kroneckerův a maticový součin jsou v tomto smyslu zaměnitelné.

Důkaz. Uvažujme $K = (k_{ij}) \in \mathbb{R}^{m \times n}, G = (g_{j\ell}) \in \mathbb{R}^{n \times o}, P \in \mathbb{R}^{r \times s}$ a $H \in \mathbb{R}^{s \times t}$. V první řadě si uvědomíme, že

$$(K \otimes P) \in \mathbb{R}^{(mr) \times (ns)}, \quad (G \otimes H) \in \mathbb{R}^{(ns) \times (ot)},$$

tedy tyto dvě matice lze v tomto pořadí násobit. Jejich součin je pak

$$\begin{aligned} (K \otimes P)(G \otimes H) &= \begin{bmatrix} k_{11}P & \cdots & k_{1n}P \\ \vdots & \ddots & \vdots \\ k_{m1}P & \cdots & k_{mn}P \end{bmatrix} \begin{bmatrix} g_{11}H & \cdots & g_{1o}H \\ \vdots & \ddots & \vdots \\ g_{n1}H & \cdots & g_{no}H \end{bmatrix} \\ &= \begin{bmatrix} (k_{11}P)(g_{11}H) + \cdots + (k_{1n}P)(g_{n1}H) & \cdots & (k_{11}P)(g_{1o}H) + \cdots + (k_{1n}P)(g_{no}H) \\ \vdots & \ddots & \vdots \\ (k_{m1}P)(g_{11}H) + \cdots + (k_{mn}P)(g_{n1}H) & \cdots & (k_{m1}P)(g_{1o}H) + \cdots + (k_{mn}P)(g_{no}H) \end{bmatrix} \\ &= \begin{bmatrix} (k_{11}g_{11})(PH) + \cdots + (k_{1n}g_{n1})(PH) & \cdots & (k_{11}g_{1o})(PH) + \cdots + (k_{1n}g_{no})(PH) \\ \vdots & \ddots & \vdots \\ (k_{m1}g_{11})(PH) + \cdots + (k_{mn}g_{n1})(PH) & \cdots & (k_{m1}g_{1o})(PH) + \cdots + (k_{mn}g_{no})(PH) \end{bmatrix}. \end{aligned}$$

V každém jednotlivém sčítanci je součin nějakých dvou čísel (různých prvků matic K a G) a dvou vždy stejných matic, totiž PH . Součin matic tak lze vytknout, dostáváme $(K \otimes P)(G \otimes H) =$

$$\begin{bmatrix} (k_{11}g_{11} + \cdots + k_{1n}g_{n1})(PH) & \cdots & (k_{11}g_{1o} + \cdots + k_{1n}g_{no})(PH) \\ \vdots & \ddots & \vdots \\ (k_{m1}g_{11} + \cdots + k_{mn}g_{n1})(PH) & \cdots & (k_{m1}g_{1o} + \cdots + k_{mn}g_{no})(PH) \end{bmatrix}.$$

Porovnejme tento výsledek se vztahem definujícím Kroneckerův součin (2.13). Vidíme, že výsledek je právě v takovém tvaru. Konkrétně platí

$$\begin{bmatrix} (k_{11}g_{11} + \cdots + k_{1n}g_{n1}) & \cdots & (k_{11}g_{1o} + \cdots + k_{1n}g_{no}) \\ \vdots & \ddots & \vdots \\ (k_{m1}g_{11} + \cdots + k_{mn}g_{n1}) & \cdots & (k_{m1}g_{1o} + \cdots + k_{mn}g_{no}) \end{bmatrix} \otimes (PH) = (KG) \otimes (PH),$$

což jsme chtěli dokázat. \square

Kroneckerův součin jsme v podstatě již použili při přepsání maticové rovnice $AX = F$ na soustavu (2.12). Všimněme si, že platí

$$\text{diag}(A, A, \dots, A) = \begin{bmatrix} A & & & \\ & A & & \\ & & \ddots & \\ & & & A \end{bmatrix} = I_n \otimes A.$$

Řadu dalších vztahů týkajících se Kroneckerova součinu lze nalézt např. v knize [3].

2.3.3 Vektorizace matic

Operace vektorizace převádí matici na sloupcový vektor. Konkrétně matici rozdělí na sloupce, které ve stejném pořadí složí do dlouhého vektoru, tj.

$$K = [k_1, \dots, k_n] \in \mathbb{R}^{m \times n}, \quad \text{vec}(K) = [k_1^T, \dots, k_n^T]^T.$$

Rozepsáno po prvcích,

$$\text{vec}(K) = [k_{11}, \dots, k_{m1}, \dots, k_{1n}, \dots, k_{mn}]^T.$$

Pro vektorizaci (v kombinaci s Kroneckerovým součinem) platí opět řada zajímavých vztahů.

Lemma 3. *Nechť u a v jsou sloupcové vektory. Pak platí*

$$\text{vec}(uv^T) = v \otimes u. \quad (2.15)$$

Pozn. že s vektorem nakládáme jako s maticí s jediným sloupcem.

Důkaz. Uvažujme

$$u \in \mathbb{R}^s, \quad v = [v_1, v_2, \dots, v_m]^T \in \mathbb{R}^m, \quad \text{pak} \quad uv^T = [uv_1, uv_2, \dots, uv_m].$$

Vektorizací dostaneme

$$\text{vec}(uv^T) = \text{vec}([uv_1, uv_2, \dots, uv_m]) = \begin{bmatrix} v_1 u \\ v_2 u \\ \vdots \\ v_m u \end{bmatrix} = v \otimes u,$$

viz (2.13). \square

Nejdůležitějším pro naši další práci bude vztah vyjádřený v následující větě.

Věta 3. *Nechť P , X a K jsou libovolné matice takové, že existuje součin PXK^T a I je jednotková matice příslušné velikosti, pak platí*

$$\text{vec}(PX) = (I \otimes P)\text{vec}(X), \quad (2.16)$$

$$\text{vec}(XK^T) = (K \otimes I)\text{vec}(X), \quad (2.17)$$

$$\text{vec}(PXK^T) = (K \otimes P)\text{vec}(X). \quad (2.18)$$

Důkaz. Uvažujme konkrétně $P \in \mathbb{R}^{r \times s}$, $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{s \times n}$ a $K \in \mathbb{R}^{m \times n}$. Pro důkaz rovnosti (2.16) si stačí uvědomit, že v maticovém součinu PX získáme každý sloupec tak, že vynásobíme matici P a odpovídající sloupec matice X , tj.

$$(PX)e_\ell = P(Xe_\ell),$$

což plyne např. z asociativity maticového násobení, kde e_ℓ značí ℓ -tý eukleidovský vektor vhodné délky (a tedy $Xe_\ell = x_\ell$). Levá strana rovnosti (2.16) tedy obsahuje vektory $P(Xe_\ell)$ seřazené pod sebou. Na pravé straně rovnosti pak vidíme v součinu matici

$$I_n \otimes P = \begin{bmatrix} P & 0 & & 0 \\ 0 & P & & 0 \\ & & \ddots & \\ 0 & 0 & & P \end{bmatrix}$$

s vektorem $\text{vec}(X)$. V ℓ -tém blokovém řádku tohoto součinu se opět nachází Px_ℓ , tj. součin matice P s ℓ -tým sloupcem matice X . Tím je první z rovností dokázána.

Pro důkaz rovnosti (2.17) matici X rozepíšeme jako součet vnějších součinů jejich sloupců a eukleidovských vektorů, tedy

$$X = \sum_{\ell=1}^n x_\ell e_\ell^T, \quad (2.19)$$

Levou stranu rovnosti (2.17) lze pak rozepsat jako

$$\text{vec}(XK^T) = \text{vec}\left(\left(\sum_{\ell=1}^n x_\ell e_\ell^T\right) K^T\right) = \text{vec}\left(\sum_{\ell=1}^n (x_\ell e_\ell^T K^T)\right) = \sum_{\ell=1}^n \text{vec}(x_\ell e_\ell^T K^T).$$

Podívejme se na jeden ze sčítanců. Zřejmě platí

$$\text{vec}(x_\ell e_\ell^T K^T) = \text{vec}\left(x_\ell (Ke_\ell)^T\right) = (Ke_\ell) \otimes x_\ell, \quad x_\ell \in \mathbb{R}^s, \quad (Ke_\ell) \in \mathbb{R}^m,$$

dle (2.15), viz lemma 3. Tento Kroneckerův součin můžeme dále dle věty 2 upravit

$$(Ke_\ell) \otimes x_\ell = (Ke_\ell) \otimes (I_s x_\ell) = (K \otimes I_s)(e_\ell \otimes x_\ell).$$

Vysčítáním jednotlivých sčítanců dostaneme

$$\text{vec}(XK^T) = \sum_{\ell=1}^n (K \otimes I_s)(e_\ell \otimes x_\ell) = (K \otimes I_s) \left(\sum_{\ell=1}^n (e_\ell \otimes x_\ell) \right).$$

Sumu ve druhém součiniteli můžeme nakonec upravit

$$\sum_{\ell=1}^n (e_\ell \otimes x_\ell) = \sum_{\ell=1}^n \text{vec}(x_\ell e_\ell^T) = \text{vec} \left(\sum_{\ell=1}^n x_\ell e_\ell^T \right) = \text{vec}(X),$$

kde opět používáme rovnosti (2.15) z lemmatu 3 a rozepsání matice X na součet vnějších součinů (2.19). Tím je druhá z rovností dokázána.

Důkaz poslední rovnosti (2.18) plyne ihned z předchozích dvou, popořadě

$$\text{vec}(PXK^T) = \text{vec}(P(XK^T)) = (I_m \otimes P)\text{vec}(XK^T) = (I_m \otimes P)(K \otimes I_s)\text{vec}(X).$$

S využitím věty 2 dostáváme

$$(I_m \otimes P)(K \otimes I_s) = (I_m K) \otimes (P I_s) = K \otimes P,$$

což jsme chtěli dokázat. □

Vektorizaci jsme v podstatě již použili při přepsání maticové rovnice $AX = F$ na soustavu (2.12), vektor neznámých je zřejmě $\text{vec}(X)$, pravá strana $\text{vec}(F)$, celá soustava pak $(I_n \otimes A)\text{vec}(X) = \text{vec}(F)$. Řadu dalších vztahů týkajících se vektorizace matic a její interakce s Kroneckerovým součinem matic lze nalézt např. v knize [3].

2.3.4 Regularita matice soustavy Ljapunovské rovnice

V sekci 2.1 jsme definovali problém kterým se zabýváme, jako rovnici (2.8), což je pro připomenutí

$$AX + XA^T = F, \quad A, F, X \in \mathbb{R}^{n \times n},$$

kde jsme z požadované symetrie matice F odvodili symetrii řešení X .

Na celou rovnici lze aplikovat operaci vektorizace, s použitím věty 3 pak můžeme součet na levé straně postupně přepsat na

$$\text{vec}(AX + XA^T) = \text{vec}(AX) + \text{vec}(XA^T) = (I_n \otimes A)\text{vec}(X) + (A \otimes I_n)\text{vec}(X).$$

Z obou sčítanců lze doprava vytknout $\text{vec}(X)$, získáme tím rovnici ve tvaru $Lx = f$,

$$Lx = \left((I_n \otimes A) + (A \otimes I_n) \right) \text{vec}(X) = \text{vec}(F) = f.$$

Nyní víme, jak přesně vypadá matice L a můžeme poprvé využít symetrii matice A . Víme, že matice je symetrická právě tehdy, když je tzv. *normální* a zároveň má reálná vlastní čísla (viz [2]). Budeme tedy uvažovat její *spektrální rozklad*

$$A = U\Lambda U^T, \quad \text{kde } \Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n) \in \mathbb{R}^{n \times n}$$

je diagonální matice obsahující vlastní čísla matice A na diagonále a kde sloupce matice $U = [u_1, u_2, \dots, u_n] \in \mathbb{R}^{n \times n}$ jsou navzájem ortonormální vlastní vektory matice A (viz [2]). Matice U je tedy tzv. *ortogonální*, tj.

$$\forall i, j = 1, \dots, n, \quad i \neq j \implies u_i^T u_j = 0, \quad \|u_i\| = 1, \quad \text{neboli} \quad U^{-1} = U^T.$$

Všechny zmíněné vlastnosti matice A lze využít k analýze vlastností matice L . Z ortogonality matice U plyne $I_n = U I_n U^T$, a tak můžeme matici L přepsat do tvaru

$$L = (I_n \otimes A) + (A \otimes I_n) = (U I_n U^T \otimes U \Lambda U^T) + (U \Lambda U^T \otimes U I_n U^T).$$

Nyní s výhodou využijeme tvrzení vět 1–2. Soustředit se budeme na první z obou sčítanců. Použitá tvrzení přitom naznačíme popiskem u jednotlivých rovnítek. Platí

$$\begin{aligned} U I_n U^T \otimes U \Lambda U^T &= U (I_n U^T) \otimes U (\Lambda U^T) \\ &=^{\text{věta 2}} (U \otimes U) (I_n U^T \otimes \Lambda U^T) \\ &=^{\text{věta 2}} (U \otimes U) (I_n \otimes \Lambda) (U^T \otimes U^T) \\ &=^{\text{věta 1}} (U \otimes U) (I_n \otimes \Lambda) (U \otimes U)^T \end{aligned}$$

Analogicky pro druhý sčítanec platí

$$U \Lambda U^T \otimes U I_n U^T = (U \otimes U) (\Lambda \otimes I_n) (U \otimes U)^T.$$

Celkově tak dostáváme matici soustavy ve tvaru

$$L = (U \otimes U) \left((I_n \otimes \Lambda) + (\Lambda \otimes I_n) \right) (U \otimes U)^T,$$

tedy faktorizovanou na součin tří matic – ze spektrálního rozkladu matice A dostáváme spektrální rozklad matice L .

Prostřední z těchto matic je zřejmě diagonální

$$(I_n \otimes \Lambda) + (\Lambda \otimes I_n) = \begin{bmatrix} \Lambda & & & \\ & \Lambda & & \\ & & \ddots & \\ & & & \Lambda \end{bmatrix} + \begin{bmatrix} I_n \lambda_1 & & & \\ & I_n \lambda_2 & & \\ & & \ddots & \\ & & & I_n \lambda_n \end{bmatrix}$$

a na diagonále tak obsahuje vlastní čísla matice L . Ta jsou všechna ve tvaru

$$\lambda_i + \lambda_j, \quad i, j = 1, 2, \dots, n,$$

neboli

$$\text{sp}(L) = \left\{ \lambda_i + \lambda_j : \lambda_i, \lambda_j \in \text{sp}(A) \right\}.$$

Matice na krajích jsou ortogonální a navzájem transponované (tedy navzájem inverzní); to plyne z rovnosti

$$(U \otimes U)(U \otimes U)^T = (U \otimes U)(U^T \otimes U^T) = (U U^T \otimes U U^T) = I_n \otimes I_n = I_{n^2},$$

kde jsme opět využili tvrzení vět 1–2. Sloupce matice $U \otimes U$ tedy představují navzájem ortonormální vlastní vektory matice L . Naše pozorování můžeme zformulovat do následující věty.

Věta 4. Necht A je reálná symetrická matice a necht $L = (I \otimes A) + (A \otimes I)$. Matice L je regulární tehdy a jen tehdy, když

$$\forall \lambda_i, \lambda_j \in \text{sp}(A), \quad \lambda_i + \lambda_j \neq 0.$$

Neboli matice L je singulární tehdy a jen tehdy, když existuje $\lambda \in \text{sp}(A)$ takové, že $-\lambda \in \text{sp}(A)$.

Důkaz. Připomeňme, že matice je regulární právě tehdy, když má nenulový determinant, a že determinant je součin všech vlastních čísel matice včetně násobností. Součin je navíc nenulový právě tehdy, když jsou nenulové všechny jeho součinitele. Matice L je tedy regulární právě tehdy, když má všechna vlastní čísla nenulová,

$$\begin{aligned} \det(L) &= \det \left((U \otimes U) \left((I_n \otimes \Lambda) + (\Lambda \otimes I_n) \right) (U \otimes U)^T \right) \\ &= \det(U \otimes U) \det \left((I_n \otimes \Lambda) + (\Lambda \otimes I_n) \right) \det((U \otimes U)^T) \\ &= \det(U \otimes U) \det \left((I_n \otimes \Lambda) + (\Lambda \otimes I_n) \right) \det((U \otimes U)^{-1}) \\ &= \det(U \otimes U) \det \left((I_n \otimes \Lambda) + (\Lambda \otimes I_n) \right) \frac{1}{\det(U \otimes U)} \\ &= \det(\Lambda) = \prod_{i=1}^n \prod_{j=1}^n (\lambda_i + \lambda_j). \end{aligned}$$

□

Poznamenejme, že věta ve skutečnosti platí pro libovolnou čtvercovou matici A . V roli spektrálního rozkladu pak můžeme použít např. Schurův rozklad, matice Λ bude obecně horní trojúhelníková, matice U unitární (viz [2]).

My budeme ve zbytku textu pracovat s maticí symetrickou a pozitivně definitní (SPD; viz sekce 2.1 předpoklad (i)), tedy takovou, že

$$\forall \lambda \in \text{sp}(A), \quad \lambda > 0.$$

Zřejmě platí A je SPD $\implies L$ je SPD (a samozřejmě regulární).

3 Metoda sdružených gradientů

Máme tedy danou maticovou rovnici

$$AX + XA^T = F, \quad \text{kde } A, X, F \in \mathbb{R}^{n \times n}$$

a kde A je SPD, F je symetrická a tedy i X je symetrická. Tuto rovnici lze interpretovat jako soustavu

$$Lx = f, \quad \text{kde } L = (I_n \otimes A) + (A \otimes I_n), \quad x = \text{vec}(X), \quad f = \text{vec}(F)$$

a L je SPD a tedy regulární. K řešení této úlohy tedy můžeme použít metodu *sdružených gradientů* (CG; z anglického *conjugate gradient*), viz algoritmus 1, která je určená k řešení soustav s SPD maticí [8], [2, kapitola 8] a [12].

Algoritmus 1 CG (sdružené gradienty) pro řešení soustavy $Lx = f$.

```

1: input  $L, f, x_0$       {SPD matice, pravá strana, počáteční odhad (std.  $x_0 = 0$ )}
2:  $r_0 \leftarrow f - Lx_0$       {počáteční reziduum}
3:  $\rho_0 \leftarrow \langle r_0, r_0 \rangle$ 
4:  $p_0 \leftarrow r_0$       {počáteční směrový vektor}
5: for  $k = 1, 2, \dots$  do
6:    $w_k \leftarrow Lp_{k-1}$ 
7:    $\alpha_k \leftarrow \rho_{k-1} / \langle w_k, p_{k-1} \rangle$ 
8:    $\beta_k \leftarrow \rho_{k-1}$ 
9:    $x_k \leftarrow x_{k-1} + p_{k-1}\alpha_k$       {aproximace řešení}
10:   $r_k \leftarrow r_{k-1} - w_k\alpha_k$       {spočtené reziduum}
11:   $\rho_k \leftarrow \langle r_k, r_k \rangle$ 
12:   $\beta_k \leftarrow \rho_k / \beta_k$ 
13:   $p_k \leftarrow r_k + p_{k-1}\beta_k$       {směrový vektor}
14: end for
15: return  $x_k$ 

```

Vektor x_k je vektorizovaná aproximace řešení, takže abychom získali matici aproximovaného řešení, je nutné x_k „devektorizovat“, tj. najít matici X_k , pro kterou platí

$$\text{vec}(X_k) = x_k.$$

Velikost matice řešení X (a tedy i její aproximace X_k) známe, takže vektor x_k rozdělíme na sloupce odpovídajícím způsobem a z těchto sloupců požadovanou matici složíme.

Oproti standardní „učebnicové implementaci“ zavádíme veličiny w_k a ρ_k proto, abychom každou (maticovou, resp. vektorovou) operaci výpočtu prováděli právě jednou. V „učebnicové implementaci“ tak zpravidla chybí řádky 3, 6, 8 a 11 a řádky 7, 10 a 12 jsou obvykle nahrazeny řádky

$$\begin{aligned} 7: & \quad \alpha_k \leftarrow \langle r_{k-1}, r_{k-1} \rangle / \langle Lp_{k-1}, p_{k-1} \rangle, \\ 10: & \quad r_k \leftarrow r_{k-1} - Lp_{k-1}\alpha_k, \\ 12: & \quad \beta_k \leftarrow \langle r_k, r_k \rangle / \langle r_{k-1}, r_{k-1} \rangle. \end{aligned}$$

Duplicitně prováděné operace jsou v „učebnicové implementaci“ zřejmé.

V algoritmu 1 metody sdružených gradientů se v každé iteraci (tj. v cyklu kroků 5–14) provádí:

- jedno násobení matice a vektoru (krok 6),
- dva skalární součiny dvou vektorů (kroky 7 a 11) a
- tři lineární kombinace dvou vektorů, tzv. *axpy* operace [19] (kroky 9, 10 a 13).

Z těchto operací je nejnáročnější součin matice a vektoru. Při výpočtu veličiny

$$w_k = Lp_{k-1}$$

provádíme součin matice L velikosti $n^2 \times n^2$ a vektoru p_{k-1} délky n^2 . Na to potřebujeme řádově n^4 elementárních operací, tj. součtů a součinů čísel. Symbolicky *složitost* výpočtu součinu Lp_{k-1} zapíšeme

$$\mathcal{S}(Lp_{k-1}) = \mathcal{O}(n^4).$$

Pokud bychom však p_{k-1} devektorizovali na matici P_{k-1} , můžeme pro výpočet použít původní maticový vztah, tj.

$$w_k = \text{vec}(AP_{k-1} + P_{k-1}A^T).$$

K výpočtu w_k je nyní třeba provést dva součiny dvou matic, přičemž obě, A i P_{k-1} , jsou velikosti $n \times n$. Na to potřebujeme řádově n^3 elementárních operací, symbolicky

$$\mathcal{S}(AP_{k-1} + P_{k-1}A^T) = \mathcal{O}(n^3).$$

Vidíme, že druhý způsob je výrazně úspornější na počet elementárních operací.

Poznamenejme, že pro výpočet skalárního součinu dvou vektorů délky n^2 , součinu vektoru s číslem i součtu dvou vektorů (*axpy* operace) potřebujeme řádově n^2 operací. Maticový zápis nám zde zjevně nemůže ušetřit (ale ani přidělat) žádnou práci.

3.1 Maticová formulace sdružených gradientů

Významný pokles výpočetních nákladů při výpočtu veličiny w_k nás přirozeně tlačí k jeho použití při výpočtu. Zároveň se nezdá být užitečné jednotlivé objekty v průběhu výpočtu stále dokola vektorizovat a devectorizovat. Je zřejmé, že je možné celý výpočet provést v „devectorizovaném“, tedy původním, maticovém tvaru. To nás vede k přepsání celého algoritmu metody sdružených gradientů pro řešení ljaminovské rovnice do maticového tvaru — tuto variantu budeme nazývat *maticové sdružené gradienty* (MCG). Pozn. že vektorizovaný tvar rovnice sloužil zejména k tomu, abychom byli schopni popsat vlastnosti matice soustavy L a rozhodnout o řešitelnosti úlohy.

Algoritmus 1 se při přepsání do maticové podoby prakticky nezmění. Víceméně jen formálně vyměníme vektory za matice (malá písmena za velká), pouze v případě řádků 2 a 6, kde byla použita matice L použijeme *ljaminovské zobrazení*

$$\mathcal{L} : M \in \mathbb{R}^{n \times n} \mapsto \mathcal{L}(M) = AM + MA^T \in \mathbb{R}^{n \times n}. \quad (3.1)$$

Dostaneme tak algoritmus 2.

Algoritmus 2 MCG pro řešení ljaminovské rovnice $AX + XA^T = F$.

```

1: input  $A, F, X_0$  {SPD matice, pravá strana, počáteční odhad (std.  $X_0 = 0$ )}
2:  $R_0 \leftarrow F - AX_0 - X_0A^T$  {počáteční reziduum}
3:  $\rho_0 \leftarrow \langle R_0, R_0 \rangle$ 
4:  $P_0 \leftarrow R_0$  {počáteční směrový vektor}
5: for  $k = 1, 2, \dots$  do
6:    $W_k \leftarrow AP_{k-1} + P_{k-1}A^T$ 
7:    $\alpha_k \leftarrow \rho_{k-1} / \langle W_k, P_{k-1} \rangle$ 
8:    $\beta_k \leftarrow \rho_{k-1}$ 
9:    $X_k \leftarrow X_{k-1} + P_{k-1}\alpha_k$  {aproximace řešení}
10:   $R_k \leftarrow R_{k-1} - W_k\alpha_k$  {spočtené reziduum}
11:   $\rho_k \leftarrow \langle R_k, R_k \rangle$ 
12:   $\beta_k \leftarrow \rho_k / \beta_k$ 
13:   $P_k \leftarrow R_k + P_{k-1}\beta_k$  {směrový vektor}
14: end for
15: return  $X_k$ 

```

V algoritmu 2 používáme pouze standardní maticové operace, snad s výjimkou skalárního součinu dvou matic (stejných rozměrů, řekněme $s \times t$). Jedná se o tzv. *Frobeniův skalární součin* definovaný zřejmým způsobem

$$\langle M, K \rangle = \langle \text{vec}(M), \text{vec}(K) \rangle = \text{vec}(K)^T \text{vec}(M) = \sum_{j=1}^s \sum_{i=1}^t k_{ij} m_{ij}.$$

Zřejmě indukuje Frobeniovu normu matice, tj. $\langle K, K \rangle = \|K\|^2$.

3.2 Vlastnosti maticově formulovaných sdružených gradientů

V následující části budeme studovat dvě vlastnosti maticové formulace sdružených gradientů, které budeme nazývat *symetrie* a *low-rankness*. První z nich bude navazovat na symetrii ljaminovské rovnice, resp. symetrii jejího řešení (viz sekce 2.2.1), druhá na aproximovatelnost řešení maticí nízké hodnosti (viz sekce 2.2.2). Ukážeme, že tyto vlastnosti se v jistém smyslu zachovávají pro všechny matice v každém kroku metody.

3.2.1 Symetrie algoritmu MCG

Nejprve dokážeme dvě jednoduchá tvrzení. První se bude týkat lineární kombinace symetrických matic, druhé použití ljaminovského zobrazení na symetrickou matici.

Lemma 4. *Součet (resp. lineární kombinace) dvou symetrických matic M a K je opět symetrická matice.*

Důkaz. Uvažujme symetrické matice M a $K \in \mathbb{R}^{n \times n}$, $M = M^T$, $K = K^T$ a $\phi, \psi \in \mathbb{R}$. Pak

$$S = M\phi + K\psi = M^T\phi + K^T\psi = (M\phi)^T + (K\psi)^T = (M\phi + K\psi)^T = S^T.$$

□

Lemma 5. *Aplikací ljaminovského zobrazení \mathcal{L} (3.1) na symetrickou matici M vznikne opět symetrická matice $\mathcal{L}(M)$.*

Důkaz. Chceme ukázat, že

$$M = M^T \quad \implies \quad \mathcal{L}(M) = \left(\mathcal{L}(M)\right)^T.$$

Toto tvrzení bylo již dokázáno v sekci 2.2.1 (dokonce jako ekvivalence; za předpokladu regularity zobrazení \mathcal{L}). Pro připomenutí

$$S = AM + MA^T = AM^T + M^T A^T = (AM)^T + (MA^T)^T = (AM + MA^T)^T = S^T.$$

□

Následující věta je důležitým důsledkem dvou předchozích lemmat.

Věta 5. *Nechť $A = A^T$ je pozitivně definitní, $F = F^T$ a $X_0 = X_0^T$, pak jsou všechny matice W_k , X_k , R_k , P_k , pro $k = 1, 2, \dots$, v algoritmu 2 symetrické.*

Důkaz. Důkaz provedeme indukcí podle indexu k . Nejprve si ale uvědomme, že všechny matice s indexem 0, tj. X_0, R_0, P_0 jsou symetrické. Symetrii počátečního odhadu X_0 předpokládá tvrzení věty. Z lemmatu 5 a symetrie matice X_0 plyne symetrie $\mathcal{L}(X_0) = AX_0 + X_0A^T$. Z lemmatu 4 pak ihned plyne symetrie R_0 (krok 2), z té zase plyne symetrie P_0 (krok 4).

Indukční předpoklad: Předpokládejme symetrii matic $X_{s-1}, R_{s-1}, P_{s-1}$ pro nějaké kladné přirozené s . Připomeňme že α a β jsou reálná čísla. Z lemmatu 5 a symetrie matice P_{s-1} plyne symetrie $\mathcal{L}(P_{s-1}) = W_s$ (krok 6). Z lemmatu 4 a symetrie matic X_{s-1} a P_{s-1} plyne symetrie X_s (krok 9). Z lemmatu 4 a symetrie matic R_{s-1} a W_s plyne symetrie R_s (krok 10) a konečně z lemmatu 4 a symetrie P_{s-1} a R_s plyne symetrie P_s (krok 13). Symbolicky

$$\text{symetrie } X_{s-1}, R_{s-1} \text{ a } P_{s-1} \quad \implies \quad \text{symetrie } W_s, X_s, R_s \text{ a } P_s.$$

Všechny matice v algoritmu MCG jsou tedy symetrické. \square

Poznamenejme, že standardní volbou počátečního odhadu X_0 je nulová matice, která je symetrická.

3.2.2 Low-rankness algoritmu MCG

Druhou vlastností kterou do algoritmu vnášíme je nízká hodnota matice F na pravé straně. Chtěli bychom, aby všechny matice X_k vypočtené v průběhu sdružených gradientů také měly nízkou hodnotu.

Lemma 6. *Součet (resp. lineární kombinace) dvou matic M a K hodností r_M a r_K je matice hodnosti nejvýše $r_M + r_K$.*

Důkaz. Uvažujme matice M a $K \in \mathbb{R}^{s \times t}$, $\text{rank}(M) = r_M$, $\text{rank}(K) = r_K$ a $\phi, \psi \in \mathbb{R}$. Z lemmatu 1 plyne

$$\begin{aligned} M &= U_M V_M^T, \quad \text{kde } U_M \in \mathbb{R}^{s \times r_M}, \quad V_M \in \mathbb{R}^{t \times r_M}, \quad \text{a} \\ K &= U_K V_K^T, \quad \text{kde } U_K \in \mathbb{R}^{s \times r_K}, \quad V_K \in \mathbb{R}^{t \times r_K}. \end{aligned}$$

Pak

$$S = M\phi + K\psi = U_M V_M^T \phi + U_K V_K^T \psi = \underbrace{\begin{bmatrix} U_M \phi, U_K \psi \end{bmatrix}}_{U_S} \underbrace{\begin{bmatrix} V_M, V_K \end{bmatrix}^T}_{V_S} = U_S V_S^T,$$

kde $U_S \in \mathbb{R}^{s \times (r_M + r_K)}$, $V_S \in \mathbb{R}^{t \times (r_M + r_K)}$. Matice W tedy nemůže být hodnosti větší než $r_M + r_K$ opět dle lemmatu 1. \square

Poznamenejme, že v případě symetrických matic M a $K \in \mathbb{R}^{n \times n}$ lze namísto lemmatu 1 analogicky použít tvrzení lemmatu 2. Dostaneme tak $M = U_M \Lambda_M U_M^T$, $K = U_K \Lambda_K U_K^T$ a

$$\begin{aligned} S = M\phi + K\psi &= U_M \Lambda_M U_M^T \phi + U_K \Lambda_K U_K^T \psi \\ &= \underbrace{\begin{bmatrix} U_M & U_K \end{bmatrix}}_{U_S} \underbrace{\begin{bmatrix} \Lambda_M \phi & 0 \\ 0 & \Lambda_K \psi \end{bmatrix}}_{\Lambda_S} \underbrace{\begin{bmatrix} U_M & U_K \end{bmatrix}^T}_{U_S} = U_S \Lambda_S U_S^T, \end{aligned}$$

kde $U_S \in \mathbb{R}^{s \times (r_M + r_K)}$, $\Lambda_S \in \mathbb{R}^{(r_M + r_K) \times (r_M + r_K)}$. Tedy i výsledná matice je rozložena na součin tří matic tak, jako oba sčítance.

Lemma 7. *Aplikací ljaminovského zobrazení \mathcal{L} (3.1) na matici M hodnosti r_M vznikne matice $\mathcal{L}(M)$ hodnosti nejvýše $2r_M$.*

Důkaz. Uvažujme matice $M \in \mathbb{R}^{n \times n}$, $\text{rank}(M) = r_M$. Z lemmatu 1 plyne

$$M = U_M V_M^T, \quad \text{kde } U_M \in \mathbb{R}^{n \times r_M}, \quad V_M \in \mathbb{R}^{n \times r_M}.$$

Pak

$$\begin{aligned} S = \mathcal{L}(M) = AM + MA^T &= AU_M V_M^T + U_M V_M^T A^T = (AU_M) V_M^T + U_M (AV_M)^T \\ &= \underbrace{\begin{bmatrix} AU_M & U_M \end{bmatrix}}_{U_S} \underbrace{\begin{bmatrix} V_M & AV_M \end{bmatrix}^T}_{V_S} = U_S V_S^T, \end{aligned}$$

kde $U_S \in \mathbb{R}^{n \times 2r_M}$, $V_S \in \mathbb{R}^{n \times 2r_M}$. Matice $\mathcal{L}(M)$ tedy nemůže být hodnosti větší než $2r_M$ opět dle lemmatu 1. \square

Poznamenejme, že v případě symetrické matice $M \in \mathbb{R}^{n \times n}$ lze místo lemmatu 1 opět analogicky použít tvrzení lemmatu 2. Dostaneme tak $M = U_M \Lambda_M U_M^T$ a

$$\begin{aligned} S = \mathcal{L}(M) = AM + MA^T &= AU_M \Lambda_M U_M^T + U_M \Lambda_M U_M^T A^T \\ &= (AU_M) \Lambda_M U_M^T + U_M \Lambda_M (AU_M)^T \\ &= \underbrace{\begin{bmatrix} AU_M & U_M \end{bmatrix}}_{U_S} \underbrace{\begin{bmatrix} 0 & \Lambda_M \\ \Lambda_M & 0 \end{bmatrix}}_{\Lambda_S} \underbrace{\begin{bmatrix} AU_M & U_M \end{bmatrix}^T}_{U_S} = U_S \Lambda_S U_S^T, \end{aligned}$$

kde $U_S \in \mathbb{R}^{s \times 2r_M}$, $\Lambda_S \in \mathbb{R}^{2r_M \times 2r_M}$. Tedy i výsledná matice je rozložena na součin tří matic tak, jako oba sčítance.

Nyní by mohla následovat věta podobně jako u symetrie, ale viděli bychom, že hodnosti všech matic (resp. jejich horní odhad; protože lemmata 6 a 7 poskytují pouze horní odhad hodností; dolní odhad je 0) v závislosti na k obecně prostou

exponenciálně (dokud budou moci růst, nemohou překročit rozměry matice) i pokud zvolíme $X_0 = 0$, tj matici hodnoti 0.

Konkrétně označíme-li $r_{M,\ell} = \text{rank}(M_\ell)$ hodnot matice $M \in \{W, X, R, P\}$ z ℓ -té iterace MCG. Zřejmě pak platí

$$\begin{aligned} r_{W,k} &\leq 2r_{P,k-1} && \text{(krok 6),} \\ r_{X,k} &\leq r_{X,k-1} + r_{P,k-1} && \text{(krok 9),} \\ r_{R,k} &\leq r_{R,k-1} + r_{W,k} \leq r_{R,k-1} + 2r_{P,k-1} && \text{(krok 10),} \\ r_{P,k} &\leq r_{R,k} + r_{P,k-1} \leq r_{R,k-1} + 3r_{P,k-1} && \text{(krok 13).} \end{aligned}$$

Pokud navíc označíme

$$r_{\max,k} = \max\{r_{X,k}, r_{R,k}, r_{P,k}\}$$

pak

$$r_{W,k} \leq 2r_{\max,k-1}, \quad r_{X,k} \leq 2r_{\max,k-1}, \quad r_{R,k} \leq 3r_{\max,k-1}, \quad r_{P,k} \leq 4r_{\max,k-1}$$

neboli

$$r_{\max,k} \leq 4r_{\max,k-1}.$$

Platí tedy

$$r_{\max,k} \leq 4^k r_{\max,0} = 4^k \max\{r_{X,0}, r_{R,0}, r_{P,0}\} \leq 4^k (r_F + 2r_{X,0}),$$

kde $r_F = \text{rank}(F)$. Stejný horní odhad samozřejmě platí i pro hodnoty matic W_k .

Na druhou stranu víme (viz sekce 2.2.2), že výsledné řešení by mělo být aproximovatelné opět maticí nízké hodnoti. Fakticky tedy musí exponenciální růst hodnotí (minimálně u matic X_k) někdy přestat (a to nejen z důvodu omezených rozměrů matice).

Pozn. že obecně nechceme argument rozměrů matice používat, protože např. pro $n = 10^6$ žije naše soustava rovnic a tedy také sdružené gradienty v prostoru dimenze $n^2 = 10^{12}$. My však zpravidla chceme hledat aproximaci řešení dosažitelnou po řádově méně krocích [12].

4 Numerické experimenty

V kapitole 3 jsme pro řešení vybrané rovnice

$$AX + XA^T = F, \quad \text{kde } A, X, F \in \mathbb{R}^{n \times n}$$

zvolili metodu sdružených gradientů. V této kapitole se podíváme na výsledky některých konkrétních numerických experimentů. Poznamenejme, že používáme MATLAB verzi 9.12.0.1884302 (R2022a) pod operačním systémem Microsoft Windows 11 Home Version 10.0 na procesoru Intel CORE i5.

4.1 Základní experiment – 2D Laplaceova matice

Jako matici soustavy budeme používat 2D diskretizaci Laplaceova operátoru. Tedy matice A , která je řídká, symetrická a pozitivně definitní, je

$$A = \begin{bmatrix} 2 & -1 & & 0 \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 2 \end{bmatrix} \in \mathbb{R}^{100 \times 100}.$$

Matice soustavy je tak ve tvaru $(A \otimes I + I \otimes A) =$

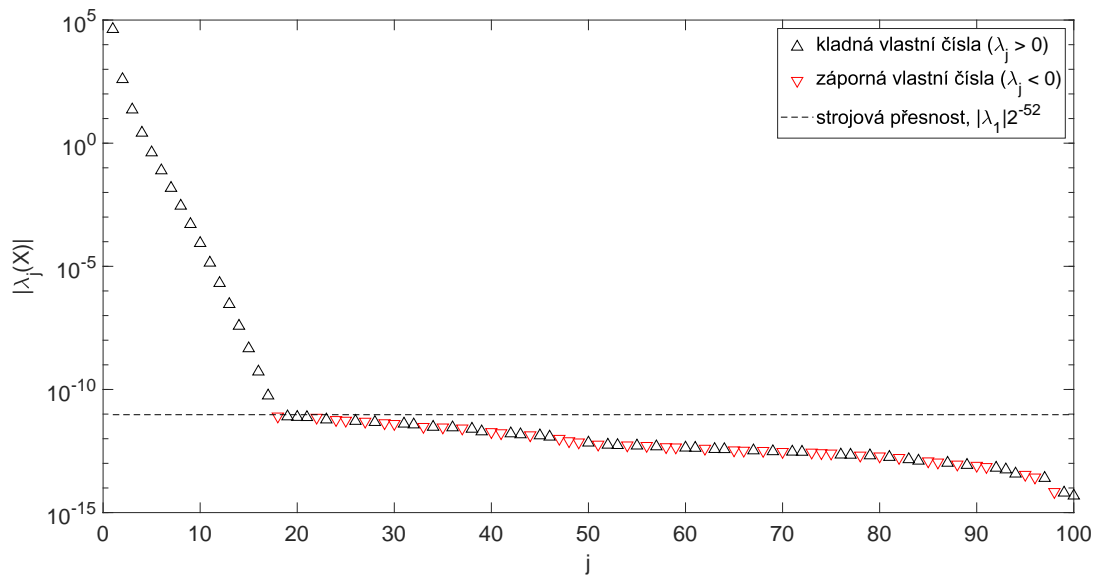
$$\begin{bmatrix} \begin{array}{cc|cc} 4 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{array} & \begin{array}{cc|cc} -1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & -1 \end{array} & & \\ \hline \begin{array}{cc|cc} -1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & -1 \end{array} & & & \\ \hline & & & \begin{array}{cc|cc} -1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & -1 \end{array} & & \\ \hline & & \begin{array}{cc|cc} -1 & & & \\ & \ddots & & \\ & & \ddots & \\ & & & -1 \end{array} & \begin{array}{cc|cc} 4 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 4 \end{array} \end{array} \in \mathbb{R}^{100^2 \times 100^2}.$$

Jako pravou stranu rovnice jsme při prvním pokusu zvolili jedničkovou matici, která splňuje požadavek na symetrii i nízkou hodnotu, tedy

$$F = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} [1 \quad \cdots \quad 1].$$

4.1.1 Ověření předpokladů

V sekci 2.2.2 jsme předpokládali, že matici řešení X lze aproximovat maticí nízké hodnoty. Graf 4.1 tento předpoklad potvrzuje. Dále v grafu vidíme, že hodnoty vlastních čísel neklesají k 0, ale kvůli strojové přesnosti klesnou přibližně na 10^{-11} .



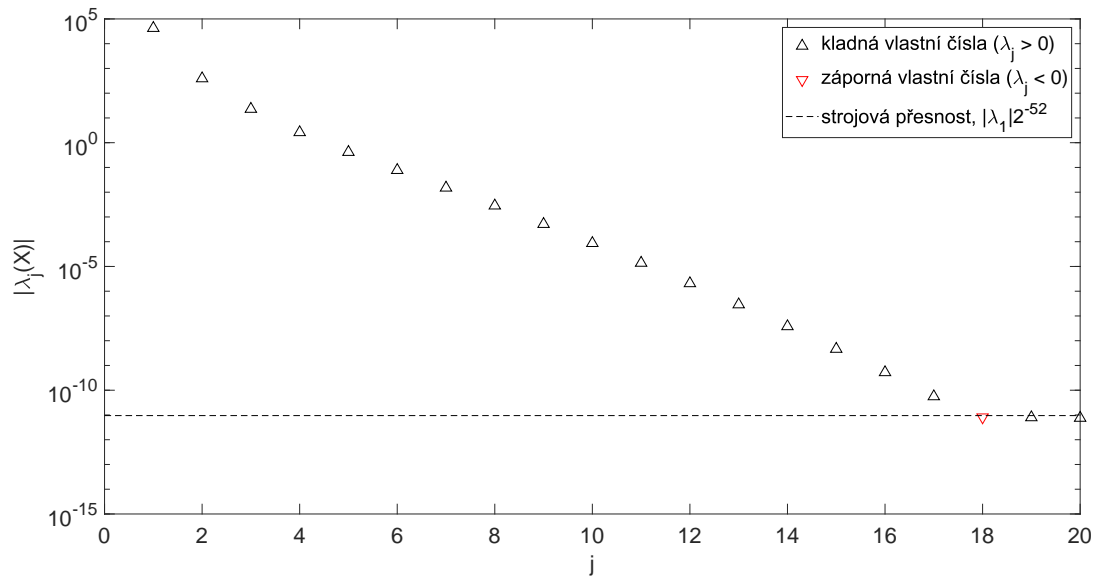
Obrázek 4.1: V grafu jsou znázorněná vlastní čísla matice X , černě kladná a červeně záporná, přerušovaná čára ukazuje největší vlastní číslo matice X vynásobené strojovou přesností, tedy $|\lambda_1|2^{-52}$.

Z grafu 4.1, resp. 4.2 je vidět hned několik věcí. Vidíme že hodnoty λ_1 až λ_{17} klesají exponenciálně a jsou kladné, matice řešení by proto mohla být pozitivně definitní. Dále také vidíme, že se pod zvýrazněnou hladinou strojové přesnosti objevuje numerický šum.

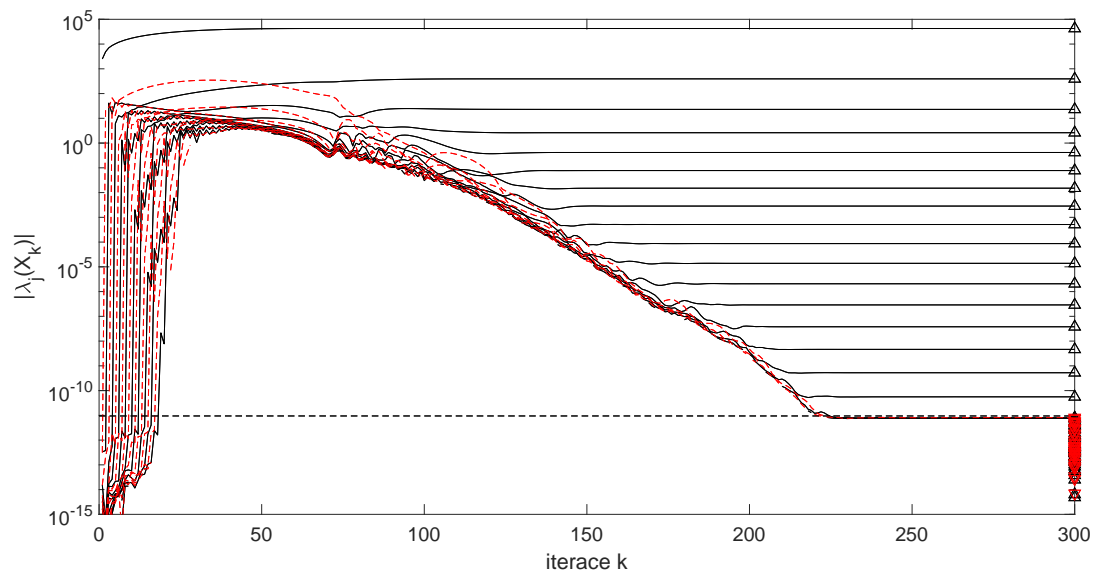
Dodejme, že pracujeme pouze s řešením získaným pomocí CG, analytické řešení pro tento výběr matice A k dispozici nemáme.

4.1.2 Nežádoucí nárůst hodnoty

Nyní víme že matice řešení X je nízké hodnoty, tím pádem uložitelná. Otázka tedy zní, zda i všechny matice X_k mají nízkou hodnotu a zda tak lze výpočet provést. Připomeňme že chceme řešit hypotetickou situaci, kdy nejsme schopni uložit n^2 prvků husté matice.



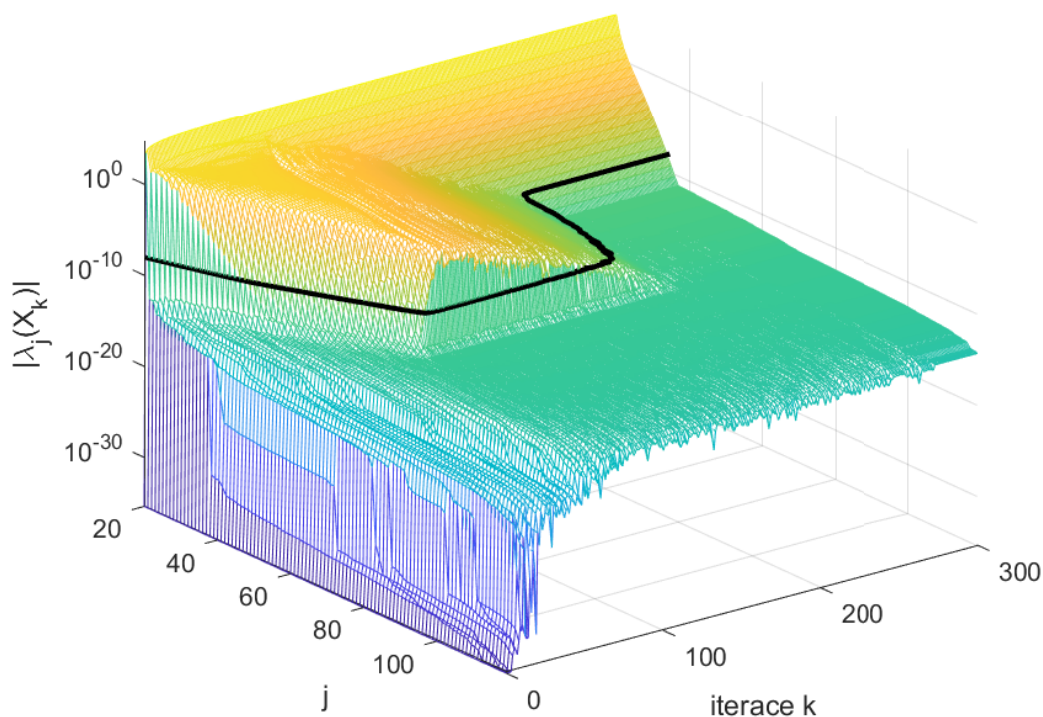
Obrázek 4.2: Graf zobrazuje přibližnou první část grafu 4.1, konkrétně prvních 20 největších vlastních čísel matice X .



Obrázek 4.3: V grafu je znázorněn průběh 20 v absolutní hodnotě největších vlastních čísel matic X_k . Osa x udává iteraci, černě jsou kladná, červeně záporná vlastní čísla. Přerušovaná příčka znázorňuje hladinu strojové přesnosti vynásobenou v absolutní hodnotě největším vlastním číslem, tedy hodnotu ke které vlastní čísla matice X klesají. Trojúhelníčky jsou, stejně jako v grafu 4.1, černě kladná a červeně záporná vlastní čísla matice řešení X .

Na obrázku 4.3 vidíme 20 v absolutní hodnotě největších, na obrázku 4.4 všechna vlastní čísla iteračních matic X_k . Z obrázku 4.4 je patrné, že i pro tuto volbu matic A a F dosáhne v průběhu výpočtu matice X_k hodnoty přibližně $n/2$. Opět vidíme že vlastní čísla neklesají k nule, ale přibližně k 10^{-11} .

Odpověď je tedy: Ne, hodnost matic v průběhu výpočtu naroste a výpočet se tak stává neproveditelným.



Obrázek 4.4: V grafu jsou vykresleny absolutní hodnoty vlastních čísel matic X_k . Povšimněme si, že křivka ve třísté iteraci je totožná s myšlenou spojnicí bodů v grafu 4.1 a dále, že boční řez grafem odpovídá křivkám na obrázku 4.3. Černá křivka znázorňuje průběh hodnosti matice X_k .

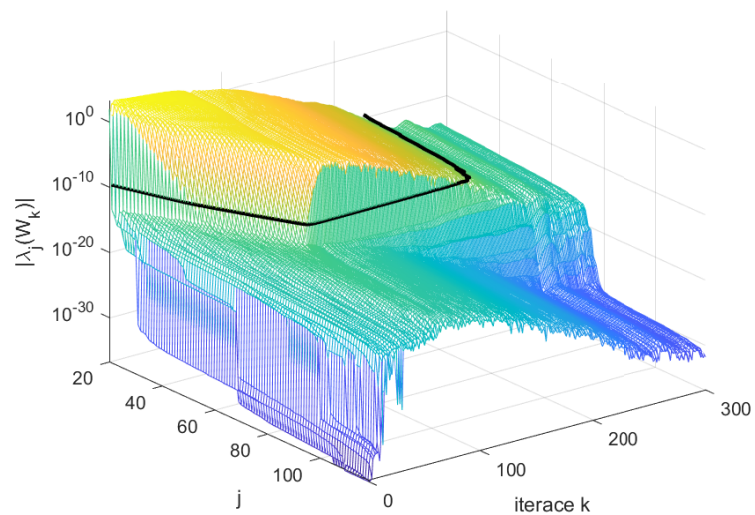
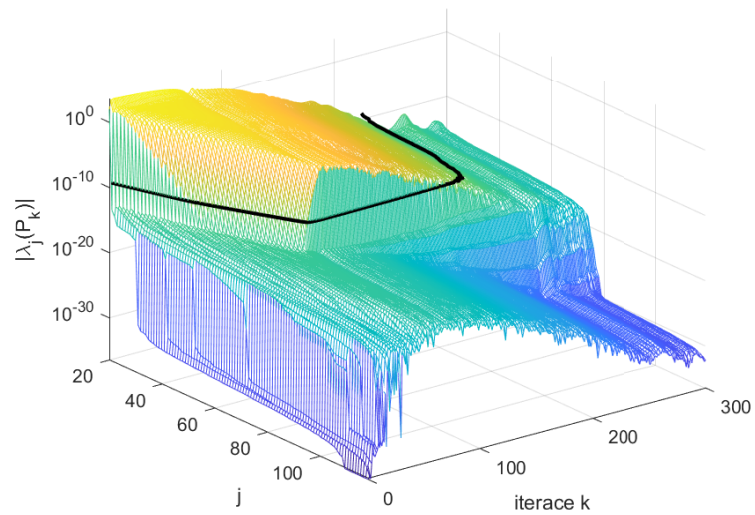
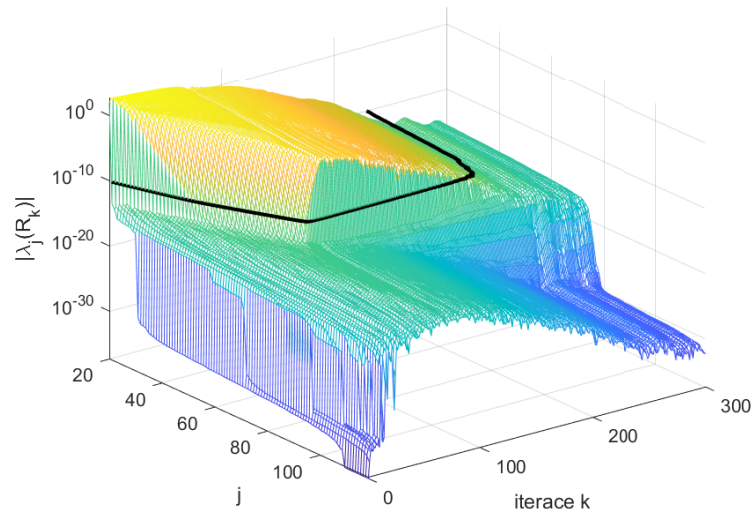
Křivka ve třísté iteraci na obrázku 4.4 je totožná s pomyslnou křivkou spojující vlastní čísla zobrazená v grafu 4.1. Křivky exponenciálně klesají, díky logaritmickému měřítku příslušných os se pokles jeví jako lineární.

Úplně stejný trend nárůstu a poklesu hodnot jako v grafu 4.4 pro X_k pozorujeme i pro směrové vektory, spočtená rezidua a směrové vektory vynásobené maticí A , viz obrázek 4.5. Na rozdíl od aproximací řešení směrové vektory konvergují blíže k nule, přibližně k 10^{-17} , tento jev by mohl být způsoben strojovou přesností nebo zaokrouhlovacími chybami.

4.1.3 Vliv zaokrouhlovacích chyb

Při použití numerické metody CG mohou vznikat, jako při každém výpočtu na počítači, zaokrouhlovací chyby. V této sekci rozebereme významnost jejich vlivu na pozorované jevy.

Zaokrouhlovacích chyb se lze zbavit simulováním přesné aritmetiky, což lze provést reortogonalizací reziduí, viz [14], [6] a [18]. Vzhledem k tomu, že metoda je



Obrázek 4.5: V prvním grafu jsou zobrazena rezidua R_k , ve druhém grafu jsou směrové vektory P_k zobrazeny analogicky jako vektory X_k v obrázku 4.4. Třetí graf znázorňuje absolutní hodnotu vektorů $W_k = AP_{k-1} + P_{k-1}A^T$.

v jistém smyslu ortogonální (rezidua r_k metody CG jsou ortogonální ve standardním skalárním součinu), v přesné aritmetice pak stačí reortogonalizaci provést právě jednou, platí pravidlo „*twice is enough*“ [6], [4, str. 4].

Reortogonalizace je vytvoření ortonormální báze vektorového prostoru, v našem případě prostoru reziduí. Implementaci vidíme v algoritmu 3.

Algoritmus 3 MCG s reortogonalizací (simulace přesné aritmetiky).

```

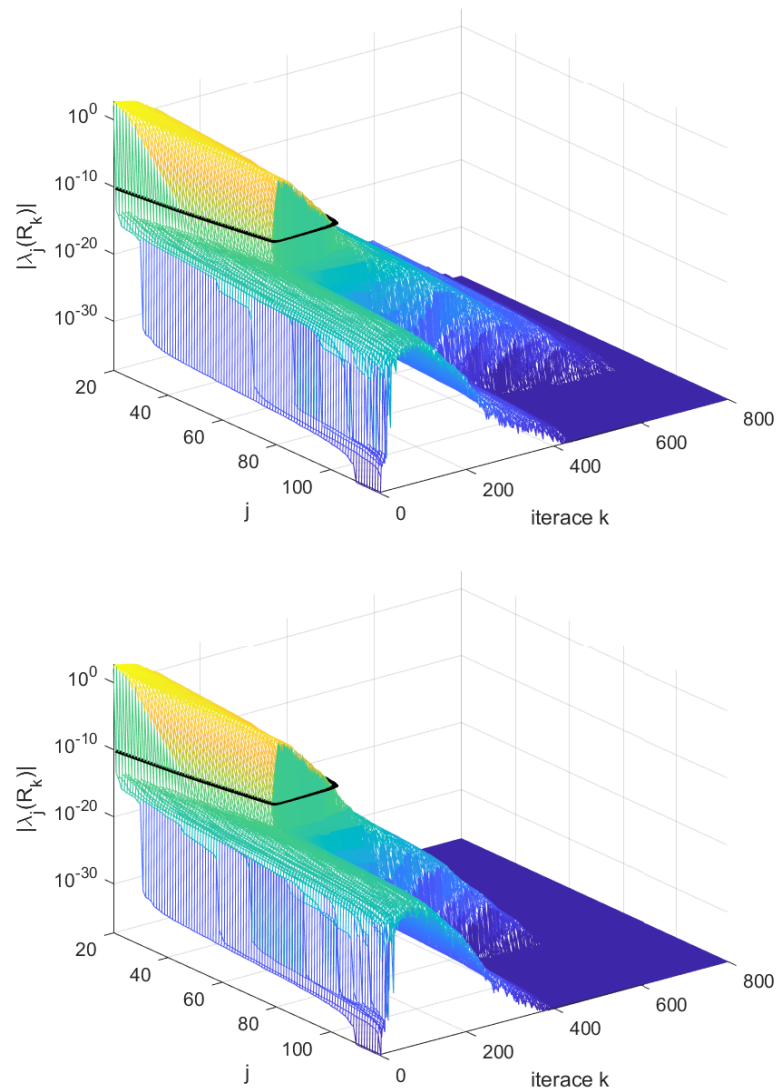
1: input  $A, F, X_0$   {SPD matice, pravá strana, počáteční odhad (std.  $X_0 = 0$ )}
2:  $R_0 \leftarrow F - AX_0 - X_0A^T$   {počáteční reziduum}
3:  $\rho_0 \leftarrow \langle R_0, R_0 \rangle$ 
    $Q_0 \leftarrow R_0 / \sqrt{\rho_0}$   {GS reortogonalizace}
4:  $P_0 \leftarrow R_0$   {počáteční směrový vektor}
5: for  $k = 1, 2, \dots$  do
6:    $W_k \leftarrow AP_{k-1} + P_{k-1}A^T$ 
7:    $\alpha_k \leftarrow \rho_{k-1} / \langle W_k, P_{k-1} \rangle$ 
8:    $\beta_k \leftarrow \rho_{k-1}$ 
9:    $X_k \leftarrow X_{k-1} + P_{k-1}\alpha_k$   {aproximace řešení}
10:   $R_k \leftarrow R_{k-1} - W_k\alpha_k$   {spočtené reziduum}
    $R_k \leftarrow R_k - \sum_{\ell=0}^{k-1} Q_\ell \langle Q_\ell, R_k \rangle$   {aktualizace rezidua; GS reortogonalizace}
11:   $\rho_k \leftarrow \langle R_k, R_k \rangle$ 
    $Q_k \leftarrow R_k / \sqrt{\rho_k}$   {GS reortogonalizace}
12:   $\beta_k \leftarrow \rho_k / \beta_k$ 
13:   $P_k \leftarrow R_k + P_{k-1}\beta_k$   {směrový vektor}
14: end for
15: return  $X_k$ 

```

Oproti implementaci maticových sdružených gradientů (algoritmus 2) přibyly tři neočíslované řádky, následují po řádcích 3, 10 a 11 a realizují Grammovu-Schmidtovu ortogonalizaci. Provádíme tak vlastně QR-rozklad matice jejíž sloupce jsou $\text{vec}(R_k)$, $k = 0, 1, 2, \dots$; ortonormální báze jejího oboru hodnot je tvořena $\text{vec}(Q_k)$, $k = 0, 1, 2, \dots$, koeficienty (skalární součiny ve druhém nečíslovaném řádku) neukládáme. Aktualizovaná rezidua se od ortonormálních liší pouze normou, tj. násobkem $\sqrt{\rho_k}$.

Na obrázku 4.6 je vidět, že prvních 300 iterací se neliší od grafu na obrázku 4.4. Z dřívějších grafů víme, že po provedení 300 iterací máme dostatečně přesnou matici řešení X , tedy výrazné zmenšení reziduí po čtyřtisté iteraci nemá pro praktický výpočet žádný význam. Pozorovaný efekt nárůstu hodnoty nezmizel, není tedy způsobený ztrátou ortogonality.

V grafech na obrázku 4.7 jsou znázorněny odchylky reziduí od ortonormality v prvních sto iteracích. V prvním grafu jsou rezidua bez průběžné reortogonalizace, ve druhém rezidua reortogonalizovaná. Rezidua metody MCG (sloupcové vektory)



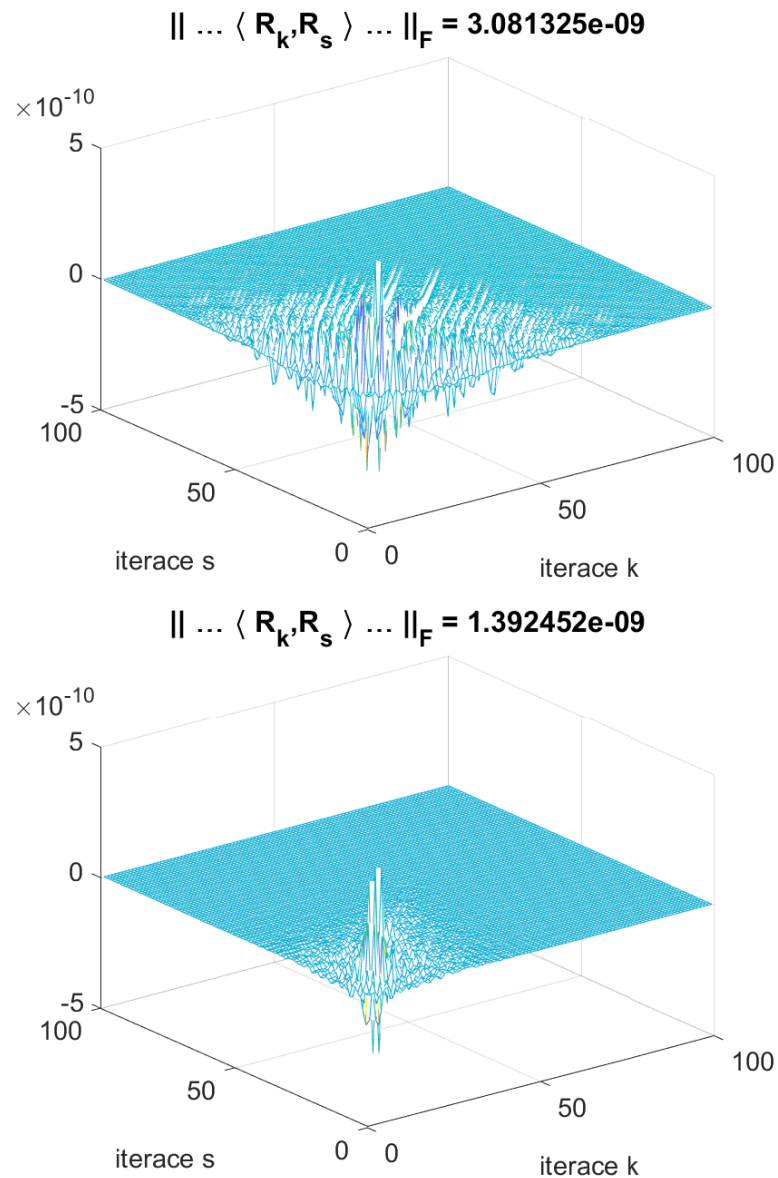
Obrázek 4.6: Oba grafy znázorňují rezidua v průběhu MCG, v prvním grafu jsou zobrazena rezidua stejná jako na prvním grafu obrázku 4.5, ve druhém grafu jsou rezidua reortogonalizovaná. Pro viditelnost rozdílu bylo provedeno 800 iterací – reortogonalizovaná rezidua klesají k nule rychleji.

jsme použili k sestrojení matice Θ ,

$$\Theta = \begin{cases} \langle R_k, R_s \rangle, & s \neq k \\ 0, & s = k \end{cases} .$$

Diagonála je nulová, byly by na ní normy reziduí. Na obrázku 4.7 jsou zobrazeny Frobeniovy normy matic Θ .

Z výsledku pokusů jasně vyplývá, že reortogonalizace výpočet nezrychlí (ve smyslu snížení počtu iterací), ani neusnadní (hodnost matic narůstá v obou případech zhruba stejně, viz grafy na obrázku 4.6; zcela analogické grafy dostaneme



Obrázek 4.7: Z reziduí metody MCG (sloupců) jsme složili matici M , v obou grafech jsou zobrazeny Frobeniovy normy matic Θ , jejichž diagonály jsou nulové. V prvním grafu je matice M sestavená z reziduí bez průběžné reortogonalizace, ve druhém byla použita rezidua reortogonalizované metody, v obou je znázorněno pouze prvních sto iterací.

pro matice X_k , P_k i W_k). Reortogonalizaci tedy není nutné provádět, nárůst hodnoty není artefakt výpočtu v aritmetice s konečnou přesností. Ve všech dalších experimentech budeme tedy používat původní metodu MCG.

4.2 Různé pravé strany

Matici A tedy i matici soustavy ponecháme, ale provedeme několik experimentů s různými pravými stranami F . Jak bylo již uvedeno, budeme používat metodu MCG bez reortogonalizace. Matici pravé strany budeme volit tak, aby měla různá, tedy kladná i záporná vlastní čísla.

4.2.1 Pravá strana z jedniček, nul a minus jedniček

Nejprve se ale vrátíme k pokusu popsaného v sekci 4.1 s jedničkovou maticí na pravé straně

$$F = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} [1 \quad \cdots \quad 1] \in \mathbb{R}^{100 \times 100}.$$

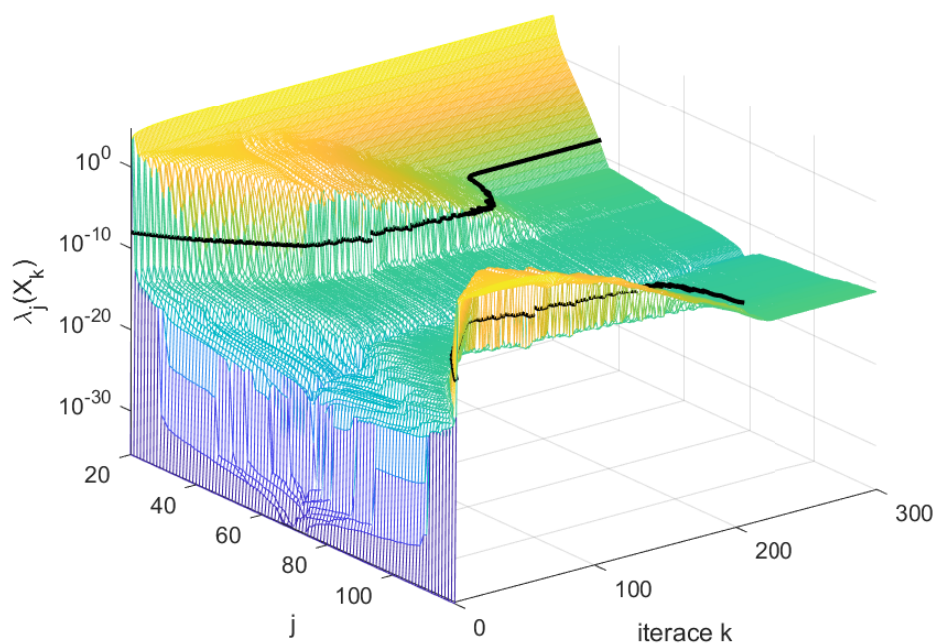
Na obrázku 4.8 jsou zobrazena vlastní čísla matic X_k , resp. jejich absolutní hodnoty, stejně jako na obrázku 4.4. Na obou obrázcích jsou ale jinak seřazena. Zatímco na obrázku 4.4 jsou seřazena podle absolutních hodnot, na obrázku 4.8 jsou seřazena podle hodnot skutečných — vidíme tedy zvlášť kladná a zvlášť záporná vlastní čísla. Z obrázku je dobře patrné, že záporná vlastní čísla v průběhu iterací zmizí a zůstanou jen kladná. Nyní se podíváme, zda to tak platí obecně.

Pro tento účel použijeme jinou pravou stranu

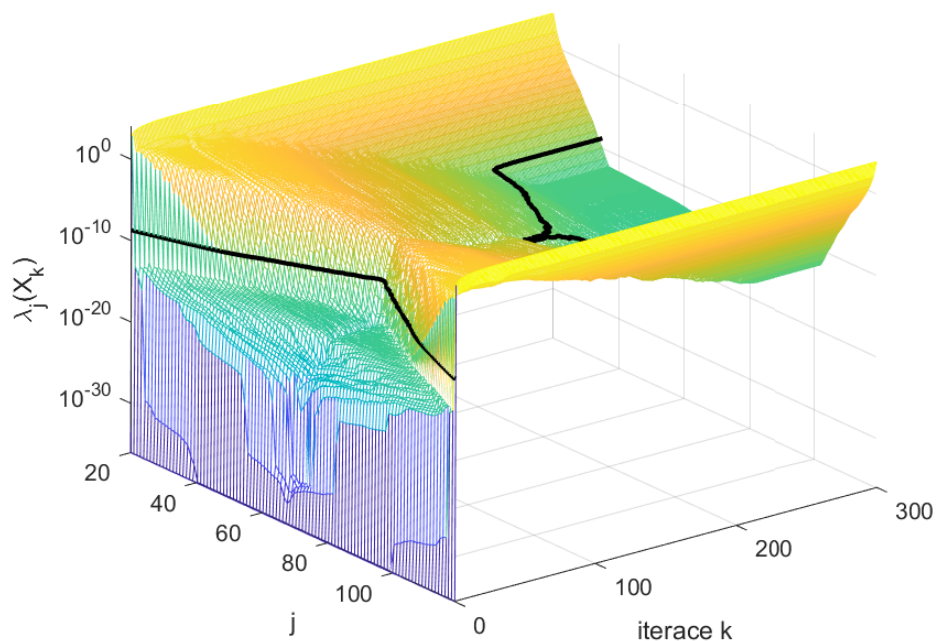
$$F = \begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & -1 & \cdots & -1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & \cdots & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} [1 \quad \cdots \quad 1 \quad 0 \quad \cdots \quad 0] - \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} [0 \quad \cdots \quad 0 \quad 1 \quad \cdots \quad 1] \in \mathbb{R}^{100 \times 100}.$$

Poznamenejme, že F má hodnotu dva, tj. obsahuje dvě nenulová vlastní čísla, přičemž jedno je kladné a jedno záporné, konkrétně ± 50 (matice F je rozdělená na čtyři bloky stejného řádu). V grafu na obrázku 4.9 opět odlišujeme kladná a záporná vlastní čísla matic X_k . Vidíme, že tentokrát záporná čísla nemizí, což je přirozené. Poznamenejme, že pokud zvolíme pravou stranu ve tvaru součtu dvou matic, řešení bude ve tvaru součtu řešení spočteného pro jednotlivé matice, totéž platí pro libovolnou lineární kombinaci, protože zobrazení $\mathcal{L}(X) = AX + XA^T$ je lineární.

Pro porovnání s jedničkovou pravou stranou (obrázek 4.3) je v grafu na obrázku 4.10 znázorněn průběh 40 v absolutní hodnotě největších vlastních čísel matic X_k

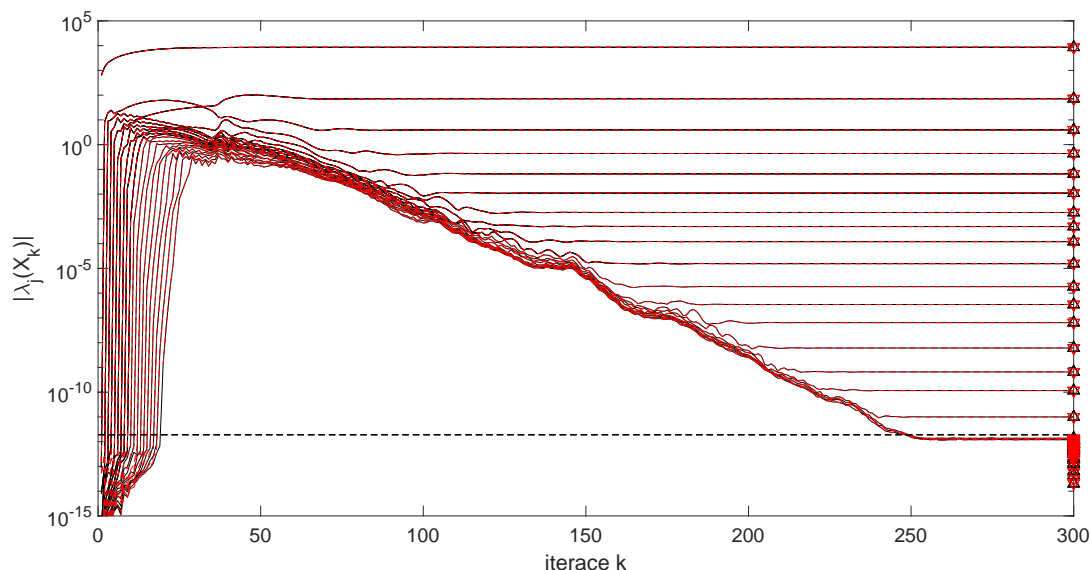


Obrázek 4.8: Graf zobrazuje vlastní čísla matic X_k při řešení stejné úlohy jako graf na obrázku 4.4 s tím rozdílem, že nyní jsou vlastní čísla rozdělená na kladná a záporná, přičemž záporná v průběhu výpočtu vymizí.



Obrázek 4.9: V grafu jsou vlastní čísla matic X_k při použití pravé strany hodnoti 2 složené ze čtyř čtvercových bloků $n/2 \times n/2$ ($n = 100$), první blok je z jedniček, druhý a třetí jsou nuly a čtvrtý blok tvoří minus jedničky.

pro pravou stranu složenou z popsaných bloků. Opět jsou černou vykreslena kladná a červenou barvou záporná vlastní čísla, tentokrát mají po dvou stejnou absolutní hodnotu a proto se v grafu překrývají. Analogicky se překrývají i trojúhelníčky ve třísté iteraci.



Obrázek 4.10: Graf je analogický ke grafu na obrázku 4.3 a znázorňuje průběh tentokrát 40 v absolutní hodnotě největších vlastních čísel matic X_k . Osa x udává iteraci, opět jsou černě kladná a červeně záporná vlastní čísla. Přerušovaná přímka je opět hladina strojové přesnosti vynásobená v absolutní hodnotě největším vlastním číslem. Trojúhelníčky jsou, stejně jako v grafu 4.1, černě kladná a červeně záporná vlastní čísla matice řešení X .

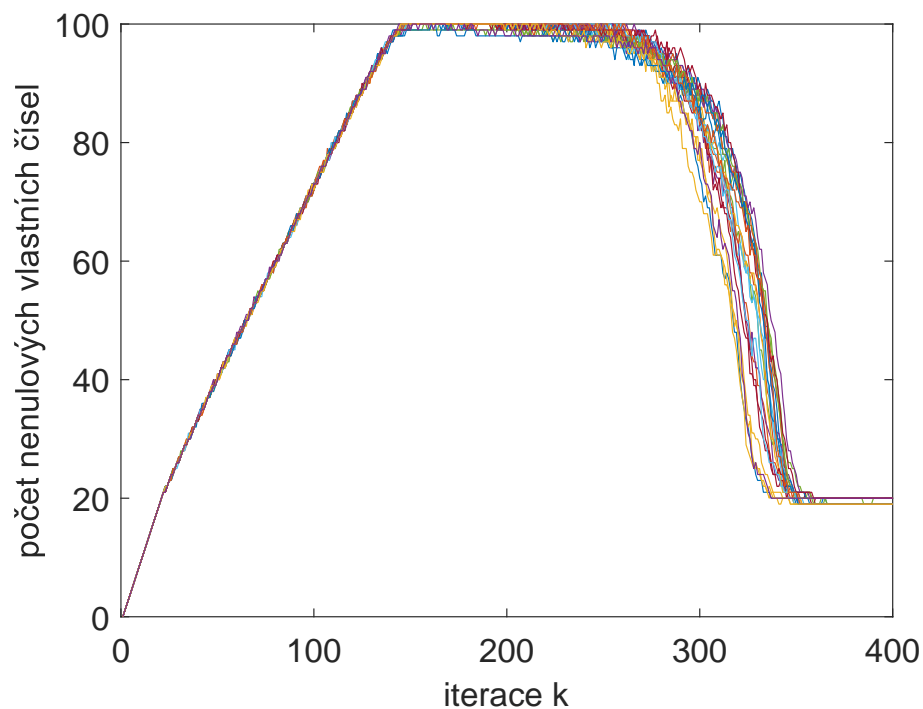
4.2.2 Pravá strana z náhodných čísel

Jako pravá strana F byla zvolena matice hodnosti jedna z náhodných čísel generovaných nejprve pomocí `rand(100, 1)`, poté pomocí `randn(100, 1)`. Je dobré říct, že `rand` generuje náhodná čísla v rozmezí od 0 do 1 rovnoměrného rozdělení a `randn` generuje čísla normálního rozdělení, mají tedy střední hodnotu 0.

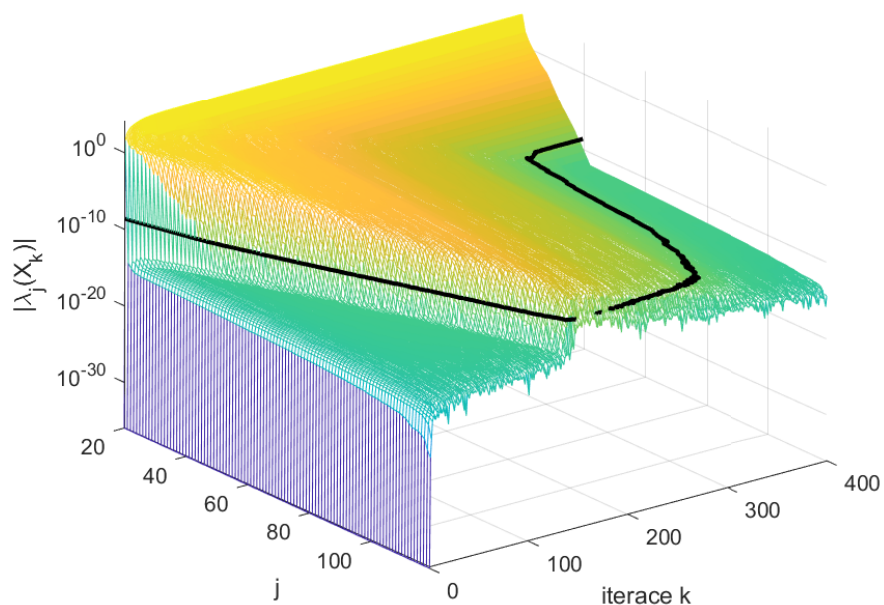
Každým příkazem bylo postupně vygenerováno 25 vektorů, jejich vnější součiny nám poskytly 25 symetrických pravých stran hodnosti jedna. Použití příkazu `rand(100, 1)` ukazují grafy na obrázcích 4.11 a 4.12, použití příkazu `randn(100, 1)` obrázky 4.13 a 4.14.

Graf na obrázku 4.11 pro zjednodušení zobrazuje pouze křivky průběhů hodnot matic X_k , tedy vlastně černou křivku v grafu 4.4, na obrázku 4.12 jsou vlastní čísla matice X_k v průběhu jednoho pokusu. Grafy 4.13 a 4.14 jsou analogické ke grafům 4.11 a 4.12, pouze byl použit generátor `randn(100, 1)`.

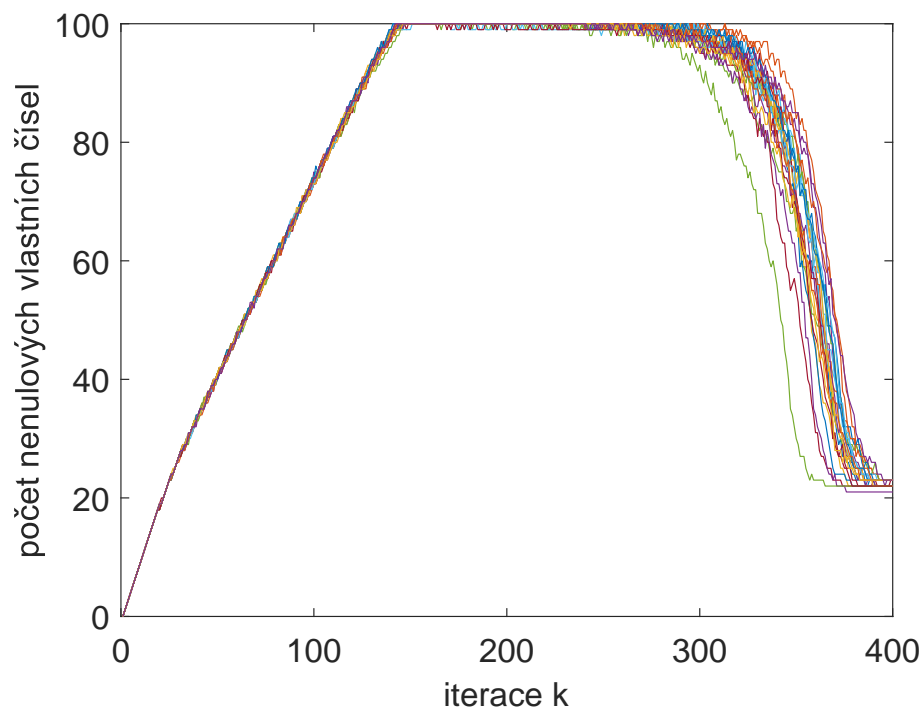
Na obrázcích 4.13 a 4.14 mají náhodná čísla normální rozdělení, matice proto mohou obsahovat kladné i záporné komponenty. V porovnání s rovnoměrným rozdělením je jejich *konvergence pomalejší*. Exponenciální pokles vlastních čísel viditelný



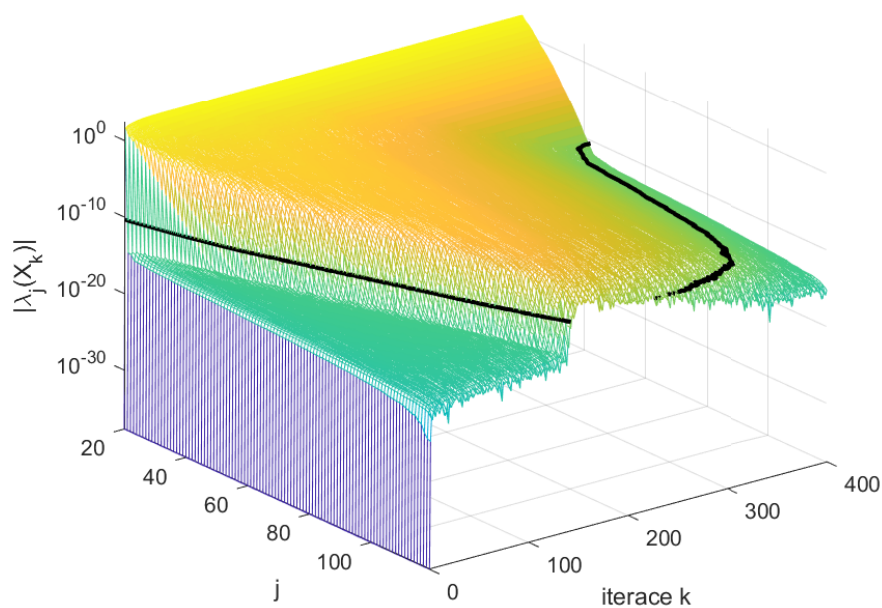
Obrázek 4.11: V grafu je 25 křivek, každá přísluší jednomu výpočtu metodou MCG. Jako pravé strany byly použity matice náhodných čísel hodnoti jedna (generované pomocí `rand(100, 1)`). Křivky představují průběh hodnoti vlastních čísel matic X (tedy se jedná o černou křivku z obrázku 4.12). Na rozdíl od případu s jedničkovou maticí na pravé straně bylo potřeba provést 400 iterací.



Obrázek 4.12: V grafu jsou vykresleny absolutní hodnoty vlastních čísel matic X_k pro pravou stranu hodnoti jedna z náhodných čísel (generovaných `rand(100, 1)`). Černá křivka ukazuje průběh hodnoti matice X_k a je to jedna z křivek vykreslených v grafu na obrázku 4.11.



Obrázek 4.13: V grafu je 25 křivek, každá přísluší jednomu výpočtu metodou MCG. Jako pravé strany byly použity matice náhodných čísel hodnosti jedna (generované `randn(100, 1)`). Opět bylo provedeno 400 iterací.



Obrázek 4.14: V grafu jsou vykresleny absolutní hodnoty vlastních čísel matic X_k pro pravou stranu hodnosti jedna z náhodných čísel (generovaných `randn(100, 1)`). Černá křivka ukazuje průběh hodnoty matice X_k a je to jedna z křivek vykreslených v grafu na obrázku 4.13.

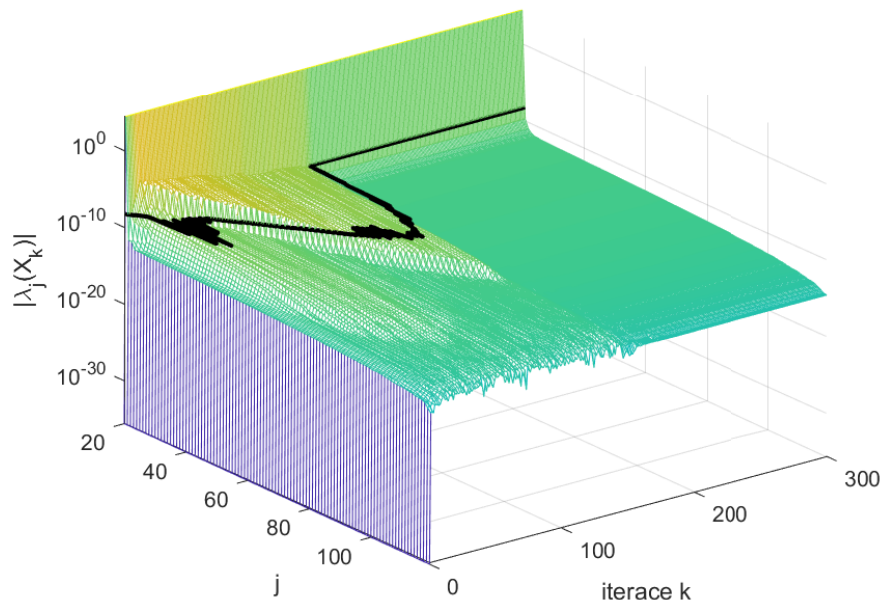
ve čtyřsté iteraci je pro druhý případ *pomalejší*. Provedeme-li porovnání obou těchto výsledků s výsledky prvního experimentu v sekci 4.1 vidíme, že pro pravou stranu z náhodných čísel bylo potřeba provést více iterací. Matice X_k tentokrát v průběhu výpočtu získaly plnou hodnotu, na rozdíl od prvního pokusu kdy měly hodnotu přibližně poloviční.

4.2.3 Pravá strana zkonstruovaná tak, aby nenarostla hodnota

Může vyvstat otázka, zda vůbec existuje pravá strana F pro kterou k nárůstu hodnoty matice X_k nedojde. Jako vhodné se nabízí vyzkoušet vlastní vektory Laplaceovy matice, F tedy zvolit jako vnější součin vlastního vektoru matice A se sebou samým. Než byl proveden vnější součin, byl k vektoru přidán na páté desetinné místo náhodný šum, jinak by vykreslená černá křivka byla přímkou. Pravá strana tedy vypadá následovně:

$$\begin{aligned} F &= f f^T, & f &= v_p + s, & f, v_p, s &\in \mathbb{R}^n, & n &= 100, \\ \langle v_p, e_j \rangle &= \sin(p \cdot j \cdot \pi / (n + 1)), & j, p &\in \{1, \dots, n\}, \\ s &= s_0 \cdot (10^{-6}) \cdot \|v\| / \|s_0\|, & s_0 &= \text{rand}(n, 1). \end{aligned}$$

Parametr p udává počet půlperiod sinu ve vektoru v_p , e_j je j -tý eukleidovský vektor; použito bylo $p = 1$. Graf příslušný tomuto pokusu je na obrázku 4.15. Tento konkrétní příklad je samozřejmě zkonstruovaný tak, aby vyšel hezky, v praxi hodnota matic narůstá.



Obrázek 4.15: V grafu jsou zobrazeny absolutní hodnoty vlastních čísel pro pokus s vhodně zvolenou pravou stranou s přidáním šumu na pátém desetinném místě.

4.3 Použití předpodmiňovače

Matici A ponecháme a provedeme jeden experiment s předpodmíněnou variantou MCG. Předpodmínění provedeme pomocí zobrazení \mathcal{M} , které je realizované symetrickou pozitivně definitní maticí. Obecně je smyslem předpodmiňovače aproximovat (v nějakém smyslu) inverzi k zobrazení \mathcal{L} tak, aby se urychlil výpočet. V našem případě by měl zamezit nárůstu hodnoty matic X_k . Upravená rovnice vypadá následujícím způsobem

$$\mathcal{L}(X) = F \quad \rightarrow \quad \mathcal{M}(\mathcal{L}(X)) = \mathcal{M}(F).$$

Konkrétně jsme zvolili

$$\mathcal{M} \sim A^{-1} \otimes A^{-1},$$

aplikace tohoto předpodmiňovače ekvivalentně znamená použití standardní MCG na úloze

$$\begin{array}{l} A^{-1} \backslash \quad AX + XA = F \quad /A^{-1} \\ \quad \quad \quad XA^{-1} + A^{-1}X = A^{-1}FA^{-1} \end{array}$$

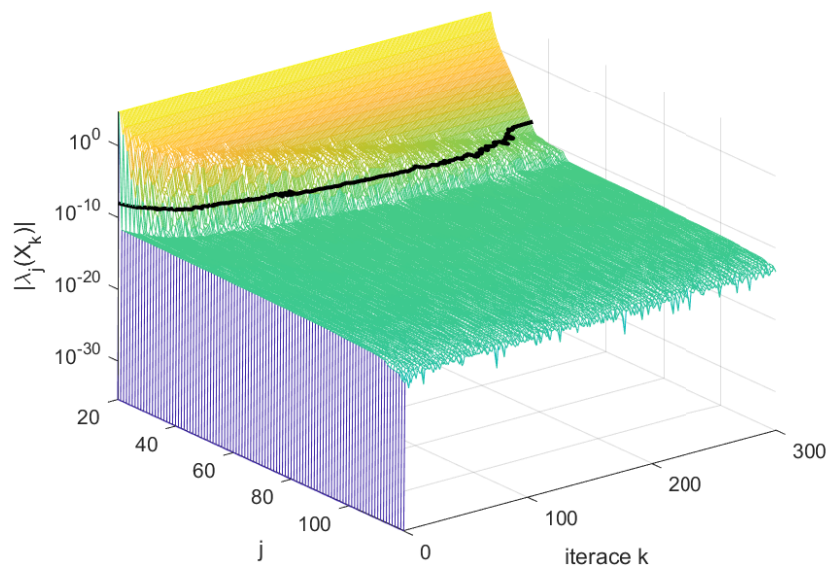
V algoritmu 4 tak oproti původní MCG vidíme přidaný řádek

$$Z_k = A^{-1}R_kA^{-1}.$$

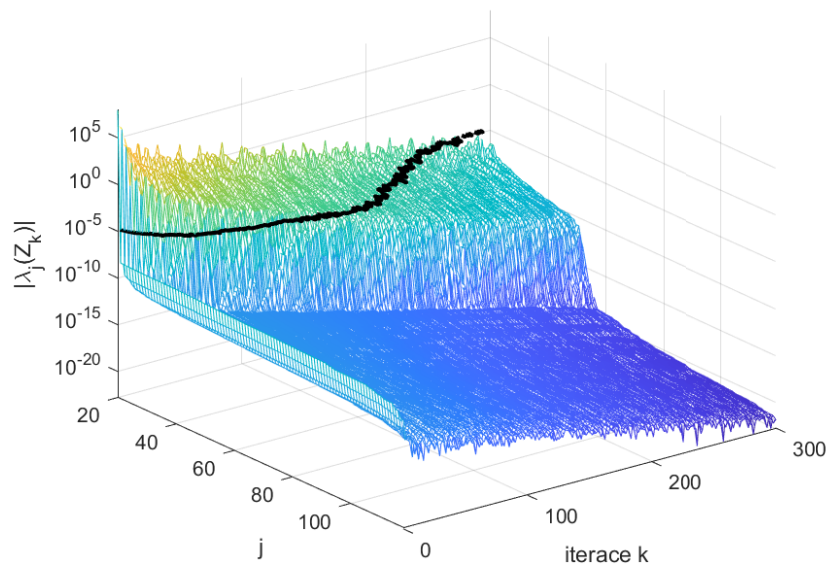
Algoritmus 4 MCG s předpodmíněním.

- 1: **input** A, \mathcal{M}, F, X_0 {SPD matice, předpodmiňovač, pravá str., poč. odhad}
 - 2: $R_0 \leftarrow F - AX_0 - X_0A^T$ {počáteční reziduum}
 - $Z_0 \leftarrow \mathcal{M}(R_0)$ {předpodmínění}
 - 3: $\rho_0 \leftarrow \langle Z_0, R_0 \rangle$
 - 4: $P_0 \leftarrow Z_0$ {počáteční směrový vektor}
 - 5: **for** $k = 1, 2, \dots$ **do**
 - 6: $W_k \leftarrow AP_{k-1} + P_{k-1}A^T$
 - 7: $\alpha_k \leftarrow \rho_{k-1} / \langle W_k, P_{k-1} \rangle$
 - 8: $\beta_k \leftarrow \rho_{k-1}$
 - 9: $X_k \leftarrow X_{k-1} + P_{k-1}\alpha_k$ {aproximace řešení}
 - 10: $R_k \leftarrow R_{k-1} - W_k\alpha_k$ {spočtené reziduum}
 - $Z_k \leftarrow \mathcal{M}(R_k)$ {předpodmínění}
 - 11: $\rho_k \leftarrow \langle Z_k, R_k \rangle$
 - 12: $\beta_k \leftarrow \rho_k / \beta_k$
 - 13: $P_k \leftarrow Z_k + P_{k-1}\beta_k$ {směrový vektor}
 - 14: **end for**
 - 15: **return** X_k
-

V grafu na obrázku 4.16 je vidět, že hodnost matic X_k pro takto předpodmíněné MCG nenarůstá. V grafu na obrázku 4.17 jsou zobrazena předpodmíněná rezidua Z_k , na obrázku 4.18 jsou grafy reziduí R_k , směrových vektorů P_k a vektorů W_k .

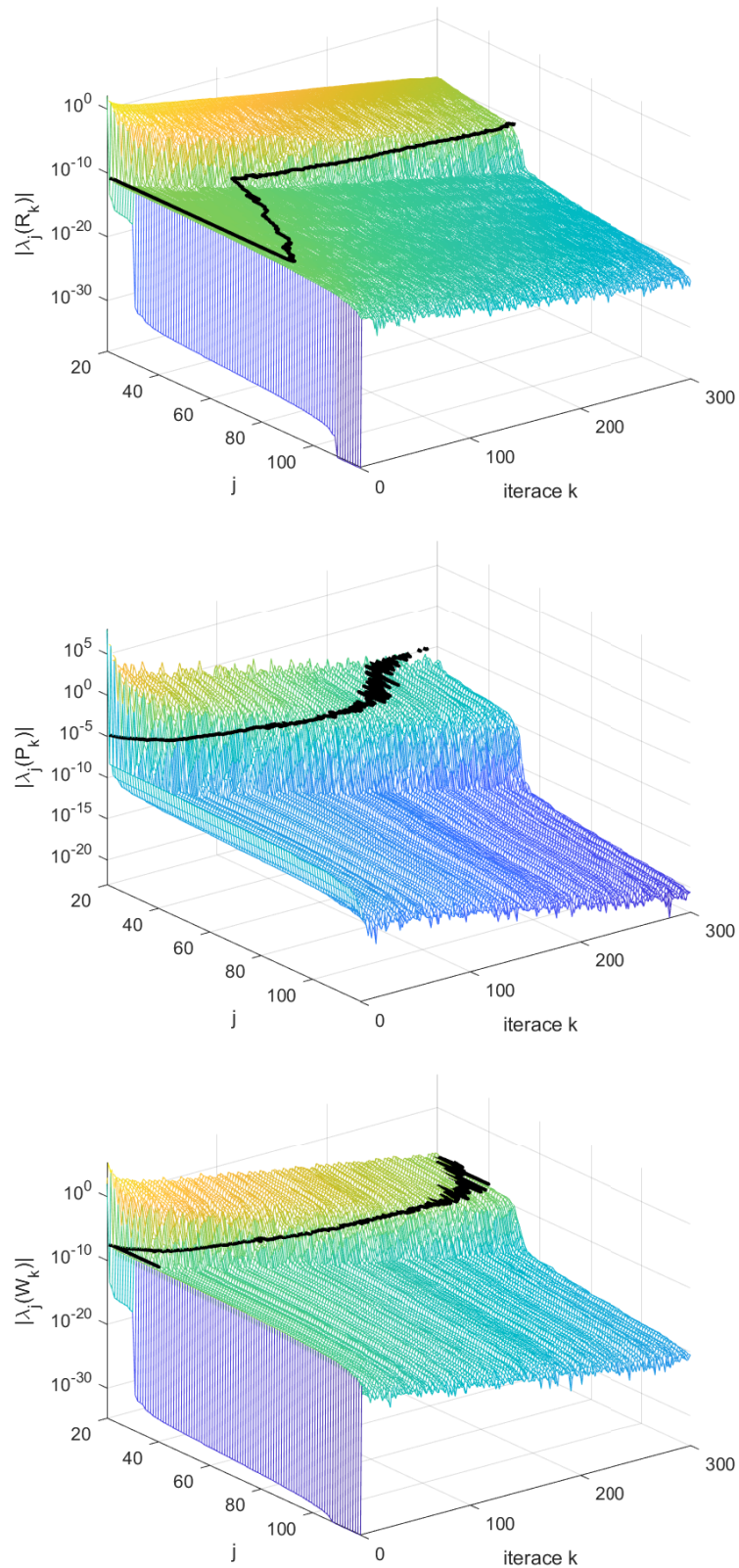


Obrázek 4.16: V grafu jsou zobrazeny absolutní hodnoty vlastních čísel matic X_k pro předpodmíněné MCG.



Obrázek 4.17: V grafu jsou zobrazena předpodmíněná rezidua Z_k pro předpodmíněné MCG.

Nevýhodou každého předpodmiňovače je nějaká práce navíc, v našem případě dva součiny matic řádu n v každé iteraci. Pokud to ale umožní vyřešení úlohy (vejít se do paměti počítače), tak se použití předpodmiňovače vyplatí. Tento konkrétní předpodmiňovač ve tvaru inverzní matice je velmi naivní, pro praktické (velké) úlohy je získání inverze a jakákoliv práce s ní neproveditelná. Na testování různých jiných předpodmiňovačů již není v práci prostor.



Obrázek 4.18: Grafy zobrazují průběhy vektorů R_k , P_k a W_k pro předpodmíněné MCG; srovnajme s obrázkem 4.5. Je vidět, že k výpočtu je potřeba více iterací, které jsou ale levnější (matice mají nižší hodnotu).

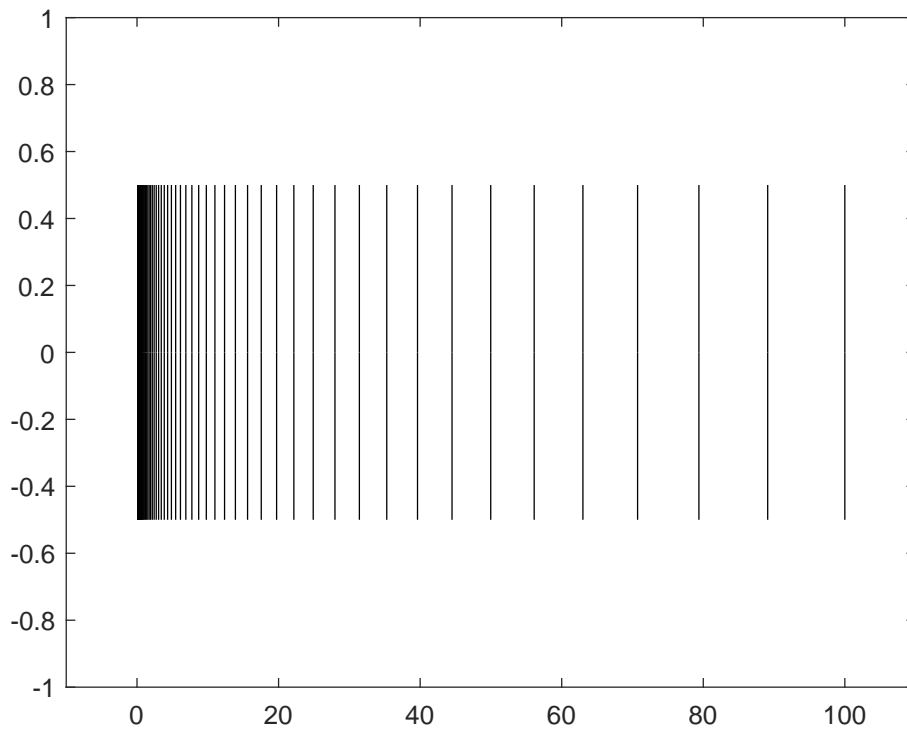
4.4 Jiné zobrazení \mathcal{L}

Nyní můžeme zkoumat, zda se budou výsledky lišit pokud zvolíme jinou matici A a změníme tak zobrazení \mathcal{L} . Je-li zvolená matice A diagonální, řešení je v principu jednoduché a lze ho snadno spočítat.

Matici A volíme Strakoš($n, \lambda_1, \lambda_n, \rho$), prvky na diagonále jsou vzestupně řazená vlastní čísla. Konkrétně zvolíme $\lambda_1 = 0.1$, $\lambda_n = 100$ a $\rho = 0.9$. Ostatní vlastní čísla matice Strakoš(100, 0.1, 100, 0.9) konstruujeme pomocí

$$\lambda_i = \lambda_1 + \left(\frac{i-1}{n-1}\right)(\lambda_n - \lambda_1)\rho^{n-i}, \quad i = 1, \dots, n.$$

Rozložení takto spočítaných vlastních čísel je přehledně znázorněné na obrázku 4.19.

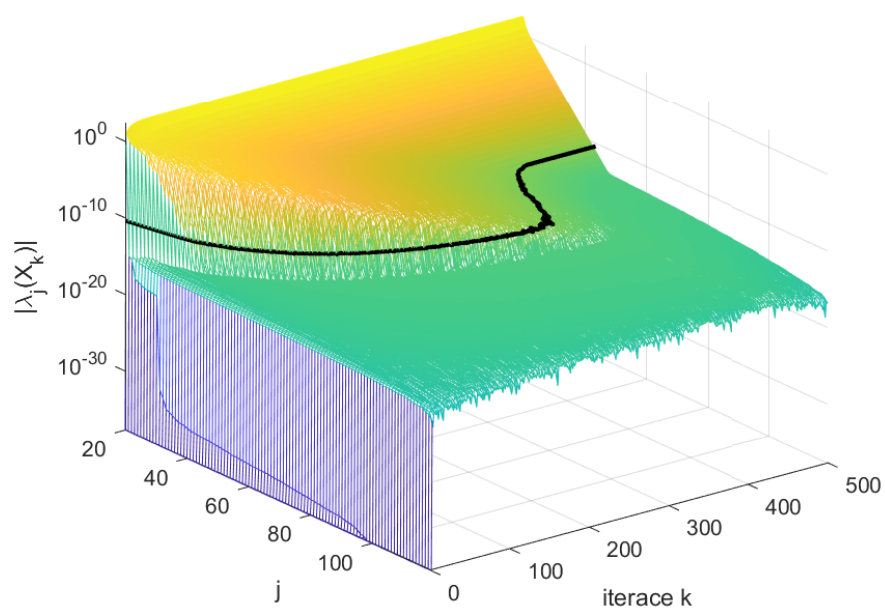


Obrázek 4.19: V grafu je zobrazeno rozložení vlastních čísel matice Strakoš(100) o parametrech $\lambda_1 = 0.1$, $\lambda_n = 100$ a $\rho = 0.9$.

Diagonalita matice A umožňuje vzhled do MCG, rozložení vlastních čísel navíc způsobuje ztrátu ortogonality, proto se používá k analýze metody, viz [13].

Na obrázku 4.20 je zobrazen průběh vlastních čísel matic X_k . Bylo potřeba provést více iterací než v minulých pokusech. Vidíme, že i při použití této matice A došlo k nárůstu hodnoty přibližně na polovinu, růst byl ale pozvolnější, distribuce vlastních čísel má evidentně významný vliv.

K nárůstu hodnoty opět došlo, ale diagonalita matice nám jak jsme již řekli umožňuje vzhled do metody MCG. Zde již není prostor pro podrobný rozbor, je možné případně navázat v další práci.



Obrázek 4.20: Graf znázorňuje průběh vlastních čísel matic X_k pro matici $A = \text{Strakoš}(100)$.

Závěr

V práci jsme se věnovali řešení Ljapunovské rovnice velkého řádu pomocí metody sdružených gradientů. Po teoretickém úvodu bylo provedeno několik numerickým experimentů. Jejich výsledky vedou k několika zjištěním. Ověřili jsme, že hodnost matic v průběhu iteračního procesu narůstá, v některých případech až do plné hodnoty. Pro zvolené matice A a F jsme ukázali, že zaokrouhlovací chyby konvergenci, její rychlost (měřenou počtem iterací) ani růst hodnoty dramaticky neovlivňují. Zabývali jsme se vlivem volby pravé strany na růst hodnoty. Nejzajímavější výsledek jsme získali při použití předpodmínění, které mělo velmi kladný vliv na konvergenci — růst hodnoty matic v průběhu předpodmíněného výpočtu (pro provedený příklad) byl významně omezen. V praxi ale tento konkrétní předpodmiňovač (v podstatě ve tvaru $A^{-1} \otimes A^{-1}$) použít *nemůžeme*, protože inverzní matice je obecně *hustá*, pro velké úlohy které chceme řešit je práce s ní nemožná. V posledním experimentu jsme doposud používanou matici A (diskretizace Laplaceova operátoru) nahradili jednodušší (diagonální) maticí Strakoš; ta by mohla být vhodná k hlubšímu porozumění procesu růstu hodnoty v průběhu výpočtu, takové analýze se bude potřeba v budoucnu dále věnovat.

Literatura

- [1] Mario Bebendorf: *Hierarchical Matrices*, Edice Lecture notes in computational science and engineering (LCNSE) 63, Springer-Verlag, Berlin, Heidelberg, 2008 (ISBN 978-3-540-77147-0).
<http://www.springer.com/la/book/9783540771463>
DOI: 10.1007/978-3-540-77147-0
- [2] E. J. Duintjer Tebbens, I. Hnětynková, M. Plešinger, Z. Strakoš, P. Tichý: *Analýza metod pro maticové výpočty. Základní metody*, druhé vydání, Matfyzpress, Praha, 2023 (ISBN 978-80-7378-481-2).
<https://matfyzpress.cz/cz/e-shop/vsechny-tituly/analyza-metod-pro-maticove-vypocty-zakladni-metody-9788073784812>
- [3] Miroslav Fiedler: *Speciální matice a jejich použití v numerické matematice*, 1. vydání, Státní nakladatelství technické literatury, Praha, 1981.
- [4] Luc Giraud, Julien Langou, Miroslav Rozložník: *On the round-off error analysis of the Gram–Schmidt algorithm with reorthogonalization*, CERFACS Technical Report 33, 2002.
https://www.cerfacs.fr/algor/reports/2002/TR_PA_02_33.pdf
- [5] Gene H. Golub, Charles F. Van Loan: *Matrix Computations*, 4th edition, Johns Hopkins University Press, Baltimor, 2013 (ISBN 978-1-42140-794-4).
<https://www.press.jhu.edu/books/title/10678/matrix-computations>
- [6] Anne Greenbaum, Zdeněk Strakoš: *Predicting the behaviour of finite precision Lanczos and conjugate gradient computations*, SIAM Journal on Matrix Analysis and Applications 1992, 13(1): str. 121–137.
<https://epubs.siam.org/doi/10.1137/0613011>
DOI: 10.1137/0613011
- [7] Wolfgang Hackbusch: *Hierarchische Matrizen: Algorithmen und Analysis*, Springer-Verlag, Berlin, Heidelberg, 2009 (ISBN 978-3-642-00222-9).
<http://www.springer.com/la/book/9783642002212>
DOI: 10.1007/978-3-642-00222-9

- [8] Magnus R. Hestenes, Eduard Stiefel: *Methods of conjugate gradients for solving*, Journal of research of the National Bureau of Standards 1952, 49(6): str. 409–436.
https://nvlpubs.nist.gov/nistpubs/jres/049/jresv49n6p409_A1b.pdf
DOI: 10.6028/jres.049.044
- [9] Barbora Košková: *Hierarchické matice: Moderní přístup k práci s velkými hustými maticemi*, Bakalářská práce, TU v Liberci, 2018.
https://kmd.fp.tul.cz/externi-data/soubory/zaverecne-prace/2018-UM2-BP_Koskova.pdf
- [10] Barbora Košková: *Soustavy lineárních rovnic s hierarchickými maticemi řešené pomocí rozkladů*, Diplomová práce, TU v Liberci, 2020.
https://kmd.fp.tul.cz/externi-data/soubory/zaverecne-prace/2020-UM2-DP_Koskova.pdf
- [11] D. Kressner, M. Plešinger, C. Tobler: *A preconditioned low-rank CG method for parameter-dependent Lyapunov matrix equations*, Numerical Linear Algebra with Applications 2014, 21(5): str. 666–684.
<https://onlinelibrary.wiley.com/doi/abs/10.1002/nla.1919>
DOI: 10.1002/nla.1919
- [12] Jörg Liesen, Zdeněk Strakoš: *Krylov Subspace Methods*, Numerical Mathematics and Scientific Computations, Oxford University Press, Oxford, 2013 (ISBN 9780199655410).
<https://global.oup.com/academic/product/krylov-subspace-methods-9780199655410>
- [13] Gérard Meurant: *The Lanczos and Conjugate Gradient Algorithms*, SIAM, Philadelphia PA, 2003 (ISBN 978-0-89871-534-7).
<https://epubs.siam.org/doi/book/10.1137/1.9780898718140>
DOI: 10.1137/1.9780898718140
- [14] Christopher C. Paige: *Error analysis of the Lanczos algorithm for tridiagonalizing a symmetric matrix*, Journal of the Institute of Mathematics and its Applications 1976, 18(3): str. 341–349.
<https://academic.oup.com/imamat/article-abstract/18/3/341/657901>
DOI: 10.1093/imamat/18.3.341
- [15] Thilo Penzl: *Eigenvalue decay bounds for solutions of Lyapunov equations: the symmetric case*, Systems & Control Letters 2000, 40(2): str. 139–144.
<https://www.sciencedirect.com/science/article/pii/S0167691100000104>
DOI: 10.1016/S0167-6911(00)00010-4

- [16] Yousef Saad: *Iterative Methods for Sparse Linear Systems*, 2nd edition, SIAM, Philadelphia PA, 2003 (ISBN 978-0-89871-534-7).
<https://epubs.siam.org/doi/book/10.1137/1.9780898718003>
DOI: 10.1137/1.9780898718003
- [17] Valeria Simoncini, Yue Hao: *Analysis of the truncated conjugate gradient method for linear matrix equations*, SIAM Journal on Matrix Analysis and Applications 2023, 44(1): str. 359–381.
DOI: 10.1137/22M147880X
- [18] Zdeněk Strakoš, Petr Tichý: *On error estimation in the conjugate gradient method and why it works in finite precision computations*, Electronic Transactions on Numerical Analysis 2002, 13: str. 56–80.
<https://etna.math.kent.edu/vol.13.2002/pp56-80.dir/pp56-80.pdf>
- [19] LAPACK USERS' GUIDE, 3rd edition, SIAM, Philadelphia PA, 1987, 1999 (ISBN 978-0-89871-447-0).
<https://netlib.org/lapack/lug>
- [20] MATRIX MARKET, Databáze 500 řádkých matic. Mathematical and Computational Sciences Division, Information Technology Laboratory (ITL), National Institute of Standards and Technology (NIST).
<http://math.nist.gov/MatrixMarket>